

Sicher Mailen – Wie geht das?

Autor: Hermann Hueck

Version 2.0

Letzte Änderung: 25.07.2014

Vorwort zur Version 1.0

Email-Sicherheit zu implementieren ist eine unbequeme Angelegenheit. Die meisten Nutzerinnen und Nutzer sind froh, wenn alles funktioniert und sie schicken ihre Mails tagtäglich recht bedenkenlos durchs Netz.

Email ist unsicherer als eine Postkarte. Das sollten wir uns klarmachen.

Wie würden wir uns doch aufregen, wenn ein Brief auf dem Weg vom Absender zum Empfänger geöffnet würde. Bei Postkarten gehen wir zwar davon aus, dass die Postangestellten ihren Inhalt lesen könnten, wenn sie wollten. Diese haben jedoch meist kein Interesse daran und tun es deshalb in der Regel nicht. Aber was würden wir sagen, wenn wir erfahren würden, dass die Post unsere Postkarten systematisch auswertet?

In der DDR wurde der Brief- und Paketverkehr (vor allem der mit dem Westen, aber nicht nur der) von der Stasi (Ministerium für Staatssicherheit) systematisch abgefangen und vor der Weiterleitung kontrolliert. Sicher, die DDR war ja auch ein „totalitärer Überwachungsstaat“. Wie wir seit den Enthüllungen von Eduard Snowden wissen, wird unser Mailverkehr, der täglich durch das Internet rauscht, von den Geheimdiensten ausgewertet - allen voran von der amerikanischen NSA (National Security Agency) und dem britischen GCHQ (Government Communication Headquarters). Dies bedeutet einen Eingriff in die Privatsphäre, der den der Stasi weit in den Schatten stellt.

Wie haben wir (meine Generation jedenfalls) in den Siebzigern und Achtzigern gegen die deutschen Volkszählungen protestiert ... Demonstrationen und Zensusverweigerung! Heute geben wir – nicht nur mit jeder Email, die wir versenden oder empfangen – ein Vielfaches an Daten über uns freiwillig preis.

Das Thema Datensicherheit umfasst wesentlich mehr als den sicheren Mailverkehr. In diesem Dokument geht es mir aber um die Emails, die ich täglich mit meinen Kommunikationspartnerinnen und -partnern (mit euch also) austausche. Damit der Mailverkehr zwischen uns sicherer wird, müssen wir uns beide um die Email-Sicherheit kümmern.

Dieses Dokument soll zeigen, was zur Realisierung eines sicheren Mailverkehrs zu beachten ist. Es ist an meine Kommunikationspartner und alle Interessierten gerichtet.

Ich bin mir darüber im Klaren, dass die wenigsten sich durch dieses mehr als 70 Seiten starke Dokument durchkämpfen wollen. Schon dieser Aufwand dürfte vielen zu hoch sein, als dass sie ihre Freizeit dafür opfern. In der selben Zeit könnte man doch auch im Garten arbeiten oder einen guten Krimi lesen.

Auch gut. Es ist wie immer eine Frage der Prioritätensetzung.

„Warum sollte ich denn meinen Provider prüfen oder sogar wechseln? Ich bin doch schon seit Jahren damit zufrieden. Meine Email (bei WEB.DE, GMX, Freenet, Google oder Yahoo und anderen) – sie funktioniert und es kostet nichts.“ mögen sich manche sagen. „Und warum den ganzen Aufwand mit der Verschlüsselung betreiben?“

Ganz kostenlos sind diese Dienste in der Regel jedoch nicht. Wir bezahlen nicht mit Geld, sondern mit unseren Daten.

Eine Umstellung zu mehr Email-Sicherheit wird es nur geben, wenn der Schutz der Privatsphäre ein wichtiges Anliegen wird.

Für diejenigen, die damit beginnen wollen, auch wenn es etwas mühsam ist, eine Empfehlung für das weitere Vorgehen ...

Kapitel 2 enthält die grundlegenden (und auch die einfacheren) Schritte zur Implementierung von

sicherer Email-Nutzung. Wer das Maximum an Email-Sicherheit erreichen will, kann sich zusätzlich mit der Email-Verschlüsselung in Kapitel 3 beschäftigen und diese umsetzen.

Kritik und Verbesserungsvorschläge sind willkommen! Gerne erhalte ich ernst gemeintes Feedback zu diesem Dokument. Ist irgend etwas sachlich nicht korrekt oder ist ein Aspekt unverständlich dargestellt? Kritik und Anregungen bitte per Mail an hermann.hueck@secure.mailbox.org (oder an hermann.hueck@mailbox.org).

Und warum stehen hier zwei Email-Adressen? Eine genügt doch ...

Kurze Antwort: Die erste Adresse ist sicherer, sie funktioniert aber nur in den allermeisten Fällen. Bei der ersten Adresse scheitert die Übertragung, wenn die sichere Übertragung nicht gewährleistet ist. Probier' einfach die erste Mail-Adresse aus. Erhältst du beim Versand eine Fehlermeldung, dann nimm die zweite Adresse. Die zweite Adresse funktioniert immer (im Zweifel auch mit unsicherer Mail-Übertragung). Warum das so ist? ... nach der Lektüre des 2. Kapitels sollte es klar geworden sein. Und teile mir bitte (an die zweite unsichere Mail-Adresse) mit, wenn's mit der ersten, der sicheren nicht geklappt hat. Es interessiert mich.

Und nun viel Spaß beim Lesen und Ausprobieren!

Hermann Hueck

Vorwort zur Version 2.0

Die erste Fassung des Dokuments war ein schneller Wurf, der noch einige Ecken und Kanten aufwies. Mir war klar: es ist noch nicht alles „rund“. Ich wollte nicht damit warten, bis das letzte Komma richtig gesetzt und auch die Inhalte komplett und ausgefeilt dargestellt waren. Deshalb hatte ich diese erste Fassung dennoch schon verbreitet.

Nun liegt mit Version 2.0 die erste gründliche Überarbeitung vor. Auch diese ist sicherlich noch nicht perfekt. Ich habe das Dokument an vielen Stellen formal und inhaltlich überarbeitet und ergänzt, sodass es sich mir und hoffentlich auch der Leserin und dem Leser deutlich runder präsentiert.

An vielen Stellen gibt es inhaltliche Präzisierungen und Ergänzungen. Einige Kapitel sind neu, andere wurden umgestellt. So ist das Dokument im Vergleich zur Version 1.0 auch deutlich umfangreicher geworden.

So habe ich das Einführungskapitel 1 mit dem Unterkapitel 1.5 um die Erläuterung einiger IT-Grundbegriffe (Client und Server, Protokoll und Verschlüsselung) erweitert. Dies soll das Verständnis der nachfolgenden Kapitel insbesondere für IT-Laien, denen diese Begriffe möglicherweise nicht geläufig sind, erleichtern. Experten mögen das Kapitel überspringen.

Das Dokument ist so geschrieben, dass es den normalen Internet-Nutzern, die keine Experten sind, einen möglichst einfachen Zugang zu den Themen, die die Email-Sicherheit betreffen, verschafft und ihnen aufzeigt, wie sichere Email eingerichtet werden kann. Wo möglich, habe ich auf die eher technischen Details verzichtet oder sie vereinfacht, um (vor allem in den Kapiteln 1 bis 5) den Textfluss nicht zu stören und technisch zu überfrachten.

An einigen Stellen wollte ich auf technische Details doch nicht verzichten. War die technische Erläuterung kurz, dann habe ich einen Absatz in einer kleineren Schriftart eingefügt. Bei längeren Erläuterungen habe ich diese in das Kapitel 5 ausgelagert. (So werden die Unzulänglichkeiten der Transport-Verschlüsselung und die Protokolle, mit denen man diese Schwachstellen beheben kann, im Kapitel 2.1.1 gestreift und im Kapitel 5.1 noch einmal vertieft.)

Auch das Glossar ist deutlich umfangreicher geworden und enthält jetzt sogar einige Begriffe wie „Provider“ und „Implementierung“. Diese Begriffe sind für mich selbstverständlich, den technisch weniger vorbelasteten Leserinnen und Lesern möglicherweise nicht. Der Leserin bzw. dem Leser empfehle ich, das Glossar auszudrucken und bei der Lektüre daneben zu legen.

Ich selbst nutze (im Juli 2014) seit ca. 4 Monaten den verschlüsselten und signierten Mail-Verkehr auf dem Mac und auf dem Android-Geräten (Smartphone und Tablet). Meine Erfahrungen sind nun in Version 2.0 als kleine Berichtigungen, Präzisierungen und als teils größere Erweiterungen in das Dokument eingeflossen. Die neu eingeführte Versionshistorie (siehe oben) zeigt, welche Kapitel wesentlich ergänzt bzw. neu hinzugekommen sind.

Neben der inhaltlichen liegt mit dieser Fassung auch eine formale Überarbeitung vor. Dies betrifft alle Aspekte: Satzzeichen, Rechtschreibung, Satzbau, Ausdruck, Stil und die Platzierung der Abbildungen.

Große Unterstützung habe ich dabei von meiner Frau erfahren, die dem Fehlerteufel kräftig zu Leibe gerückt ist und Stilmängel aufgedeckt hat. Sie hat das Dokument aus dem Blickwinkel der IT-Laiin gelesen und mich damit zu inhaltlichen Präzisierungen, zu verständlicheren Formulierungen und zu einigen Erweiterungen des Glossars angeregt. An dieser Stelle ein ausdrückliches „Dankeschön!“ an sie.

Um das Qualitätsniveau weiter anzuheben, möchte ich meine schon in der letzten Fassung

geäußerte Bitte um Feedback wiederholen und bekräftigen.

Das Dokument ist fast nicht gegendert. Ich verwende nahezu durchgängig die männliche Form, auch wenn ich beide Geschlechter ansprechen will. Ich schreibe „der Benutzer“, „der Absender“, „der Empfänger“, „der Laie“ und „der Experte“. Ich habe erwogen, überall die weibliche Form mit großem „I“ (die BenutzerIn) oder jedes Mal beide Geschlechtsformen auszuführen (die Benutzerin und der Benutzer) und mich schließlich doch zu Gunsten eines flüssigeren Textes für die männliche Form entschieden. Ich hoffe, meine emanzipierten Leserinnen fühlen sich nicht zurückgesetzt und haben dennoch den uneingeschränkten Spaß bei der Lektüre des Textes.

Diesen Spaß wünsche ich natürlich auch meinen männlichen Lesern.

Hermann Hueck

Kapitelübersicht

Was enthalten die Kapitel des Dokuments?

- **Kapitel 1** enthält eine **Einführung** ins Thema Email-Sicherheit. Hier werden auch einige Grundbegriffe der IT (Client und Server, Protokoll, Verschlüsselung) erläutert, die den IT-Laien helfen sollen, die nachfolgenden Kapitel besser zu verstehen.
- **Kapitel 2** behandelt die **Auswahl des passenden Email-Providers** und weitere **Hinweise und Erläuterungen zur sicheren Email-Konfiguration und -Verwendung**. Es enthält alles, was bei der unverschlüsselten Mail-Kommunikation zu beachten ist. Dieses Kapitel ist weniger kompliziert und hoffentlich für jeden – auch den IT-Laien – verständlich und anwendbar.
- **Kapitel 3** beschäftigt sich mit **Mail-Verschlüsselung und Mail-Signierung**. Das Thema ist recht komplex. Ich versuche, es auch den IT-Laien so verständlich wie möglich nahezubringen. Es bleibt aber eine harte Nuss. Wer es liest und kein IT-Profi ist, sollte sich darauf einstellen.
- **Kapitel 4** enthält einige **Grundsätze** zur Passwortsicherheit und zur Rechnersicherheit, die – wenn man sie beachtet – auch der Email-Sicherheit zu Gute kommen.
- In **Kapitel 5** werden einige Themen technisch vertieft. Die technischen Details habe ich aus den Kapiteln 1 bis 5 weitgehend herausgehalten, damit diese leichter verständlich und dennoch durchgängig lesbar bleiben. Technisch Interessierte können die betreffenden Themen in diesem Kapitel vertiefen. In erster Linie geht es dabei um die Verfahren zur Absicherung verschlüsselter Übertragungskanäle.
- **Kapitel 6** enthält das **Glossar**, in dem Fachbegriffe und Abkürzungen erläutert werden.

Versionshistorie

Version	Datum	Beschreibung
1.0	01.05.2014	Initiale Fassung
2.0	25.07.2014	<ul style="list-style-type: none">• Redaktionelle und inhaltliche Überarbeitung nach erstem Lektorat• Einführung der Versionshistorie• Vorwort zur Version 2.0• Neues Kapitel 1.5: Wichtige Grundbegriffe• Neues Kapitel 1.5.3.2: Transport-Verschlüsselung vs. Daten-Verschlüsselung• Neues Kapitel 2.1.1: Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?• Neues Kapitel 2.1.2: Die Verschlüsselungsqualität der Mail-Provider prüfen• Neues Kapitel 3.5.2.3: Text-Mails statt HTML-Mails• Neues Kapitel 3.5.2.4: Die verschlüsselt versendeten Mails lesbar machen• Neues Kapitel 3.5.2.6: Empfängerregeln• Ergänzung in Kap. 3.6: PGP-Unterstützung für den Chrome-Browser• Neues Kapitel 3.11: Das verschlüsselte Postfach von mailbox.org• Neues Kapitel 5: Einige technische Erläuterungen• Glossar in Kap. 6 erweitert

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Verlust der Privatsphäre im Internet.....	1
1.2	Sichere Email – warum?.....	3
1.3	Die Daten und die Metadaten der Email.....	4
1.4	Sichere Email – Zuständigkeiten.....	4
1.5	Wichtige Grundbegriffe.....	5
1.5.1	Client und Server.....	5
1.5.1.1	Spezialisierte Clients und Server.....	5
1.5.1.2	Client und Server sind Kommunikationsrollen.....	5
1.5.1.3	Kommunikationsphasen.....	6
1.5.2	Protokoll.....	6
1.5.3	Verschlüsselung.....	8
1.5.3.1	Verschlüsselung einer Datei – ein Beispiel.....	8
1.5.3.2	Transport-Verschlüsselung vs. Daten-Verschlüsselung.....	9
1.5.3.3	Transport-Verschlüsselung.....	9
1.5.3.4	Daten-Verschlüsselung.....	10
1.6	Wer mehr wissen will	11
1.7	Links zu diesem Kapitel.....	12
2	(Möglichst) Sicherer Versand unverschlüsselter Mails.....	13
2.1	Die Mail auf dem Transportweg.....	13
2.1.1	Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?.....	16
2.1.2	Die Verschlüsselungsqualität der Mail-Provider prüfen.....	17
2.2	Der richtige Email-Provider.....	19
2.2.1	Auswahlkriterien.....	19
2.2.2	Meine Auswahl.....	20
2.3	Ein paar Bemerkungen zu Gmail.....	21
2.4	Konfiguration des Mail-Clients.....	22
2.4.1	Thunderbird-Konfiguration.....	22
2.4.2	Thunderbird-Updates.....	24
2.5	Worauf man sonst noch achten muss.....	24
2.5.1	Rechnersicherheit.....	24
2.5.2	Email-Tracking verhindern.....	24
2.6	Die passende Email-App auf dem Smartphone.....	26
2.7	Warnung vor der Nutzung von Webmail.....	27
2.8	Sicherer Mail-Alias bei mailbox.org.....	27
2.9	Links zu diesem Kapitel.....	28
3	Mails verschlüsseln und signieren.....	29
3.1	Asymmetrische Verschlüsselung.....	29
3.2	Welches Verschlüsselungsverfahren – S/MIME oder PGP?.....	30
3.3	Private Key und Public Key – Wie funktioniert's?.....	32
3.4	Verwaltung des Schlüsselbunds.....	34
3.4.1	Tools zur Schlüsselverwaltung.....	35
3.4.2	Erzeugung eines neuen Schlüsselpaars.....	37
3.4.2.1	Schlüsselerzeugung mit Enigmail / Thunderbird.....	37
3.4.2.2	Limitationen von Enigmail / Thunderbird.....	38
3.4.2.3	Schlüsselerzeugung mit Gpg4win.....	39
3.4.2.4	Schlüsselerzeugung mit GPG Keychain Access.....	39
3.4.2.5	Schlüsselerzeugung auf der Kommandozeile.....	39
3.4.3	Das Widerrufszeugnis.....	39
3.4.4	Weitere Benutzer-IDs (Email-Adressen) hinzufügen.....	40

3.4.5 Die wichtigsten PGP-Schlüsseleigenschaften.....	40
3.4.6 Schlüssel exportieren und importieren.....	42
3.4.6.1 Schlüssel sicher aufbewahren.....	42
3.4.7 Konfiguration der Key-Server (Enigmail).....	43
3.4.8 Öffentliche Schlüssel mit Key-Server synchronisieren.....	43
3.4.9 Schlüssel beglaubigen.....	45
3.4.9.1 Verifikation von Schlüssel-Identität und Benutzer-Identität.....	46
3.4.9.2 c't-Kryptokampagne.....	47
3.5 Signierter und verschlüsselter Mailverkehr.....	48
3.5.1 OpenPGP-Einstellungen für jedes Mail-Konto.....	48
3.5.2 Versand und Empfang signierter und verschlüsselter Emails.....	50
3.5.2.1 Versand.....	50
3.5.2.2 Empfang.....	50
3.5.2.3 Text-Mails statt HTML-Mails.....	51
3.5.2.4 Die verschlüsselt versendeten Mails lesbar machen.....	51
3.5.2.5 Mit signierten Mails beginnen	52
3.5.2.6 Empfängerregeln.....	52
3.6 Fallstricke beim Einsatz von GnuPG.....	53
3.7 Webmail? Vergiss es!.....	53
3.7.1 Die Webmail-Alternative – Thunderbird portable.....	54
3.8 Verschlüsselte Mails auf dem Zweitrechner.....	54
3.9 Verschlüsselte Mails auf iPhone oder iPad.....	56
3.10 Verschlüsselte Mails auf Android-Smartphone oder -Tablet.....	56
3.10.1 Die passenden Android-Apps.....	56
3.10.2 Installation der Apps.....	57
3.10.3 Übertragung der Schlüssel.....	57
3.10.4 Schlüssel-Import.....	58
3.10.5 Konfiguration der Schlüsselservers in APG.....	59
3.10.6 PGP-Mail-Empfang und -Versand.....	59
3.10.6.1 Einstellungen.....	59
3.10.6.2 Versand.....	60
3.10.7 Empfang.....	61
3.11 Das verschlüsselte Postfach von mailbox.org.....	61
3.12 Links zu diesem Kapitel.....	62
4 Passwortsicherheit und Rechnersicherheit.....	64
4.1 Sichere Passwörter.....	64
4.2 Sicherheit von Rechner, Tablet und Smartphone.....	64
4.2.1 Windows sicher betreiben.....	65
4.2.2 Mac OS X sicher betreiben.....	65
4.2.3 Linux sicher betreiben.....	66
4.2.4 iOS sicher betreiben.....	66
4.2.5 Android sicher betreiben.....	66
4.3 Links zu diesem Kapitel.....	67
5 Einige technische Erläuterungen.....	68
5.1 Verschlüsselte Übertragungskanaäle – Wie sicher sind sie wirklich?.....	68
5.1.1 SSL (Secure Socket Layer) und TLS (Transport Layer Security).....	69
5.1.2 PFS (Perfect Forward Secrecy).....	69
5.1.3 HSTS (HTTP Strict Transport Security).....	70
5.1.4 DANE (DNS-based Authentication of Named Entities).....	70
5.2 Links zu diesem Kapitel.....	72
6 Glossar.....	73

Abbildungsverzeichnis

Abbildung 1: Die Mail auf ihrem Transportweg.....	15
Abbildung 2: starttls.info: Abfrage von secure.mailbox.org (Übersicht).....	18
Abbildung 3: starttls.info: Abfrage von secure.mailbox.org (Detail-Ansicht für ersten Server).....	18
Abbildung 4: Thunderbird-Konfiguration für den Postausgang-Server (SMTP).....	23
Abbildung 5: Thunderbird-Konfiguration für den Posteingang-Server (IMAP).....	23
Abbildung 6: Thunderbird: Werbemail von WEB.DE vor dem Laden der externen Inhalte.....	25
Abbildung 7: Thunderbird: Werbemail von WEB.DE nach dem Laden der externen Inhalte.....	26
Abbildung 8: Asymmetrisch verschlüsselte Kommunikation.....	30
Abbildung 9: Thunderbird: Der Enigmail-Schlüsselbund mit einigen Schlüsseln.....	35
Abbildung 10: Thunderbird: Verwaltung der Add-ons öffnen.....	35
Abbildung 11: Thunderbird: Das neuen OpenPGP-Menü nach der Installation von Enigmail.....	36
Abbildung 12: Thunderbird: Konten-spezifische OpenPGP-Konfiguration.....	37
Abbildung 13: Thunderbird: Schlüsselbund öffnen.....	37
Abbildung 14: Enigmail: Neues Schlüsselpaar erzeugen.....	37
Abbildung 15: Enigmail: Dialog zum Erzeugen eines neuen Schlüsselpaars.....	38
Abbildung 16: Enigmail: Der Schlüsselbund mit dem neu erzeugten Schlüssel (markiert).....	38
Abbildung 17: Enigmail: Liste der Benutzer-IDs eines Schlüssels.....	40
Abbildung 18: Enigmail: Schlüsselattribute.....	41
Abbildung 19: Enigmail: Den markierten Schlüssel exportieren.....	42
Abbildung 20: Enigmail: Konfiguration der Schlüssel-Server.....	43
Abbildung 21: Enigmail: Optionen für die Synchronisation mit einem Schlüssel-Server.....	44
Abbildung 22: Enigmail: Den markierten Schlüssel unterschreiben/beglaubigen.....	45
Abbildung 23: Enigmail: Definieren des Besitzervertrauens.....	45
Abbildung 24: Enigmail: PGP-Einstellungen für ein Mail-Konto.....	49
Abbildung 25: Enigmail: Weitere PGP-Optionen.....	50
Abbildung 26: PGP-Versand-Optionen.....	50
Abbildung 27: Neue Empfängerregel erstellen.....	52
Abbildung 28: Enigmail: Schlüssel-Export.....	54
Abbildung 29: Enigmail: Auch die geheimen Schlüssel exportieren?.....	55
Abbildung 30: Android: K-@ Mail: Nur Zeichensalat – Die verschlüsselte Mail ist nicht lesbar....	57
Abbildung 31: Android-Filetransfer auf dem Mac, Smartphone über USB angeschlossen: Die Schlüssel werden ins Download-Verzeichnis kopiert.....	57
Abbildung 32: Android: Die importierten Schlüssel im APG-Schlüsselbund.....	58
Abbildung 33: Android: APG-Schlüsseldetails: Ein Schlüssel mit drei Benutzer-IDs.....	58
Abbildung 34: Android: K-@ Mail: Die Mail kann jetzt gelesen werden.....	60
Abbildung 35: Android: K-@ Mail: Kryptographie-Optionen.....	60

1 Einleitung

1.1 Verlust der Privatsphäre im Internet

Tagtäglich nutzen wir das Internet: wir surfen; wir suchen (meist bei Google) nach Begriffen oder Produkten; wir chatten; wir nutzen soziale Dienste (Facebook, Twitter, Google Plus); wir skypen; wir nutzen WhatsApp statt SMS auf dem Smartphone; wir erledigen unsere Bankgeschäfte online; und wir versenden und empfangen Emails.

Viele Internetdienste nutzen wir kostenlos. Doch sie sind nicht kostenlos, wir bezahlen mit unseren Daten. Bei der Kommunikation über das Internet hinterlassen wir - wissentlich oder unwissentlich - eine Menge Spuren ... Datenspuren. Basierend auf diesen Datenspuren können sehr detailreiche Profile unserer Persönlichkeit erstellt werden, über die wir keine Kontrolle mehr haben. Unsere Privatsphäre ist nicht mehr geschützt.

Wer hat Interesse an unseren Daten? Da gibt es drei unterschiedliche Gruppen:

- Die **Anbieter (Provider) von Internetdiensten** bieten ihre Dienste häufig kostenlos an. Sie verdienen nicht an der Leistung, die sie für uns erbringen. Sie verdienen an der Werbung, die sie uns zukommen lassen. Beahlt werden sie von denjenigen, deren Werbung sie auf ihrer Internet-Plattform platzieren. Wir Nutzer bekommen die Werbung zu sehen, wenn wir die Plattform der Provider nutzen. Die Plattformen vieler Mail-Provider (web.de, gmx.net, freenet.de und viele andere) präsentieren sich deshalb häufig wie richtige Internet-Litfass-Säulen. Die Seiten von Google's Mail-Dienst sind dagegen eine wahre Erholung. Google müllt uns nicht mit Werbung zu. Google sammelt dafür unsere Daten und platziert die personalisierte Werbung geschickt und dezent in der Ergebnisseiten unserer Suche. Dabei werden auch die Inhalte unserer Mails verarbeitet, falls diese unverschlüsselt sind. (Das steht bei Google auch in den Allgemeinen Geschäftsbedingungen, denen man zustimmt, wenn man einen Google-Mail-Account eröffnet.) Abgesehen davon sind die Provider gesetzlich verpflichtet, die über ihre Nutzer gespeicherten Informationen an staatliche Stellen weiterzugeben und über diese Informationspreisgabe Stillschweigen zu bewahren.
- Die **Geheimdienste**: Die Geheimdienste aller Staaten sammeln Informationen und lassen sich naturgemäß nicht gerne auf die Finger schauen. Sie entziehen sich möglichst der parlamentarischen und der gerichtlichen Kontrolle. Wie wir seit den Enthüllungen Edward Snowden's im Sommer 2013 wissen, tun sich die NSA und der britische Geheimdienst GCHQ dabei besonders hervor. Sie ernten in großem Stil alle Daten, die im Internet unverschlüsselt oder nur schwach verschlüsselt unterwegs sind. Sie speichern sie in riesigen Datenzentren, um sie bei Bedarf auswerten zu können. Emails wurden bis vor Kurzem meist unverschlüsselt durch das Netz transportiert und waren für die Geheimdienste deshalb eine leichte Beute. (Die Verschlüsselung der Mails bei der Übertragung ist im Laufe der vergangenen zwei Jahre deutlich besser geworden und macht weiter Fortschritte.)
- **Hacker, Internet-Kriminelle**: Diese Gruppe ist meist daran interessiert, uns finanziell zu schädigen. Sie wollen Zugriff auf unser Online-Bankkonto oder auf unseren Mail-Account. Oder sie wollen auf unsere Kosten bei Amazon shoppen gehen. Oder sie wollen in unsere Rechner einbrechen, um weitere sensible Informationen zu ergattern oder um diese zu kontrollieren und weiteren Unfug damit zu treiben (z.B. diesen als Spamschleuder zu verwenden, wofür sie dann bezahlt werden). Hacker versuchen ebenfalls unverschlüsselte Mails mitzulesen, um Passwörter, Kontonummern oder Kreditkarteninformationen

abzugreifen. Haben wir den Mail-Account nur mit einem schwachen Passwort geschützt, ist dieser leicht zu knacken. Die Kriminellen loggen sich über Webmail in den Account ein und verschicken Mails in unserem Namen (typischerweise Spam). Oder sie greifen unseren Mail-Provider direkt an und erlangen so auch die Kontrolle über unseren Mail-Account und unsere Mails. Sie können also alles lesen, was nicht verschlüsselt ist.

Wie sehr wir die Kontrolle über unsere Privatsphäre verloren haben, ist nicht erst seit der NSA-Affäre klar, die der Whistleblower Edward Snowden im Sommer 2013 durch seine Enthüllungen ins Rollen gebracht hat. Allerdings hat die NSA-Affäre die Themen Datenschutz und Privatsphäre doch so stark ins öffentliche Bewusstsein gebracht, dass sie für immer mehr Menschen ein wichtiges Anliegen werden, auch für Menschen, die keine IT-Spezialisten sind.

Am 05. Juni 2014 jährt sich die ersten Enthüllungen von Edward Snowden zum ersten Mal. An dieser Stelle möchte ich einige Links zu Kommentaren prominenter Persönlichkeiten zur Situation des Datenschutzes ein Jahr danach nennen:

- „Im NSA-Skandal ist ein langer Atem gefragt!“, ein Kommentar von Peter Schar, Bundesbeauftragter für Datenschutz und Informationsfreiheit von 2003 bis 2013:
<http://www.heise.de/newsticker/meldung/Kommentar-Im-NSA-Skandal-ist-ein-langer-Atem-gefragt-2214717.html>
- „Es ist Zeit, die Netze zurückzuerobern“, ein Kommentar von Erich Moechel, Journalist mit den Schwerpunktthemen Datenschutz, Datensicherheit, Verschlüsselung und militärische IT:
<http://www.heise.de/newsticker/meldung/Kommentar-zum-NSA-Skandal-Es-ist-Zeit-die-Netze-zurueckzuerobern-2216016.html>
- „Die technologische Souveränität zurückgewinnen“, ein Kommentar von Dorothee Bär, Bundestagsabgeordnete der CSU und Parlamentarische Staatssekretärin beim Bundesministerium für Verkehr und digitale Infrastruktur:
<http://www.heise.de/newsticker/meldung/Kommentar-zum-NSA-Skandal-Die-technologische-Souveraenitaet-zurueckgewinnen-2216143.html>
- „Ein Jahr NSA-Skandal und noch viel zu tun“, ein Kommentar von Christoph Wegener, IT-Leiter der Fakultät für Elektrotechnik und Informationstechnik an der Ruhr-Universität Bochum und freiberuflich in den Bereichen Informationssicherheit und Datenschutz tätig:
<http://www.heise.de/newsticker/meldung/Analyse-Ein-Jahr-NSA-Skandal-und-noch-viel-zu-tun-2215730.html>

Auf die Kompromittierung der Privatsphäre müssen Antworten gefunden werden. Einerseits sind dies politische Antworten (z.B. Kontrolle der Geheimdienste). Diese sollen hier nicht das Thema sein. Andererseits sind dies technologische Antworten. Diese müssen die Anbieter der diversen Internetdienste liefern. Aber auch wir Internetnutzer können einiges tun. Darum geht es hier.

Was kann man also tun, ohne gleich auf die Nutzung des Internets zu verzichten?

Ein Grundsatz ist die **Datensparsamkeit**. Wir sollten davon ausgehen, dass Daten, die wir einmal im Internet hinterlassen haben, nicht mehr gelöscht werden können. Wir sollten uns also überlegen, welche Informationen über uns wir bei Facebook und in anderen sozialen Netzwerken posten.

Ein weiterer Grundsatz ist die **Verwendung sicherer Passwörter** für Dienste, bei denen wir uns anmelden müssen. (siehe auch Kap. 4.1)

Datensparsamkeit und sichere Passwörter sind in diesem Dokument aber nicht das Hauptthema. Hier geht es mir in erster Linie um sichere Email-Kommunikation.

1.2 Sichere Email – warum?

Zu jedem der verschiedenen Kommunikationsdienste (Surfen, Chatten, Skypen, Mailen, SMS versenden, etc.) gibt es Möglichkeiten, sie sicherer zu machen und so ein Stück der Privatsphäre wieder unter die eigene Kontrolle zu bringen. In diesem Dokument geht es mir um die möglichst sichere Verwendung des Dienstes Email.

Eines gilt allerdings für alle Internetdienste: Will man die Internetnutzung sicherer machen, um die Preisgabe privater Daten zu minimieren, muss man zusätzlichen Aufwand betreiben. **Sicherheit ist unbequem.** Dies beginnt schon mit der Passwortverwaltung. Ein einfaches, leicht zu merkendes Passwort für alle Dienste ist viel einfacher zu nutzen als viele unterschiedliche und komplizierte Passwörter. Ein einfaches Passwort ist aber auch leichter zu knacken. Und wenn sich die Passwörter nicht unterscheiden, kann man mit dem geknackten Passwort gleich in alle Accounts einbrechen, die dieses Passwort verwenden.

Mit der Email verhält es sich genau so. Auch sicherer Mailverkehr ist zunächst einmal unbequem für den Benutzer. Das beginnt bereits damit, dass man sich plötzlich mit Dingen beschäftigen soll, die einen gar nicht wirklich interessieren. Eigentlich will man ja nur schnell mal 'ne Mail schreiben.

Es gibt ein paar einfache Dinge, die man leicht beachten kann und die kein großes Expertenwissen erfordern. Dies ist z.B. die Wahl des richtigen Email-Providers. Diese und ein paar weitere Hinweise zur Email-Sicherheit füllen das Kapitel 2 dieses Dokuments. Um die Email-Kommunikation komplett abzusichern, müssen die Mails verschlüsselt werden. Dies ist allerdings nicht trivial und ist Thema im Kapitel 3.

Warum schreibe ich dieses Dokument? Aus eigenem Interesse.

Eine sichere Suche im Internet durchzuführen oder sicheres Online-Banking zu betreiben, dies ist meine Sache und die des Anbieters des Dienstes (also des Anbieters der Suchmaschine oder der Bank, die das Online-Banking im Internet bereitstellt). Den Suchmaschinen-Provider kann ich mir aussuchen. Es muss nicht immer Google sein. Auch nicht *Bing (Microsoft)* oder *Yahoo*. Es geht auch mit *DuckDuckGo*, *Startpage*, *Metager* oder *Unbubble*. Diese alternativen Anbieter behaupten zumindest von sich, meine Suchanfragen nicht auszuwerten, um daraus ein Persönlichkeitsprofil zu konstruieren. Weil Google mich kennt, kann Google bessere, genau auf mich zugeschnittene Suchergebnisse liefern. Weil Google mich kennt, kann mir Google die genau auf mich zugeschnittene Werbung aber auch gleich mitliefern. Und wird dafür von den Werbetreibenden gut bezahlt. Das ist ja schließlich das Google-Geschäftsmodell.

Um eine sichere Internetsuche muss ich mich alleine kümmern. Emails sind Nachrichten zwischen zwei Kommunikationspartnern, Nachrichten zwischen dir und mir. Deshalb sind wir beide für die Email-Sicherheit zuständig. Ich versuche also, dich – meinen Kommunikationspartner – für sichere Email-Kommunikation zu gewinnen, zu unser beider Vorteil.

„Encryption works.“ Richtig implementierte Verschlüsselung funktioniert und ist laut Edward Snowden das Einzige, worauf wir uns heute im Internet noch verlassen können.

Edward Snowden konnte die NSA-Dokumente unbemerkt entwenden, weil er sie verschlüsselt hatte. Die Kommunikation mit den Journalisten Glenn Greenwald erfolgte über verschlüsselte Mails, die auch die NSA nicht mitlesen konnte.

Was heißt das konkret? Wenn ich einen sicheren Email-Provider habe und du einen unsicheren,

kann ich dir keine vertrauliche Mail schicken. Und du mir natürlich auch nicht. Und verschlüsselte Email-Kommunikation funktioniert auch nur, wenn beide Kommunikationspartner einen Schlüssel haben. Bei der Email müssen wir uns also gemeinsam um die Sicherheit kümmern.

Deshalb habe ich dieses Dokument für meine Kommunikationspartner geschrieben. Ich habe es so geschrieben, dass es auch für den IT-Laien möglichst verständlich ist und dabei einige Vereinfachungen in Kauf genommen, die für die praktische Umsetzung nicht entscheidend sind. Dennoch ist vor allem das Kapitel 3, für den Laien, der sich das erste Mal mit dem Thema Mail-Verschlüsselung beschäftigt, durchaus eine Herausforderung.

1.3 Die Daten und die Metadaten der Email

Die **Email-Daten**, das ist der Inhalt der Mail, der Nachrichtentext.

Bei einem klassischen Brief ist es das, was ich in den Umschlag stecke und was auch die Post (mein Brief-Provider) nicht lesen kann.

Die **Email-Metadaten** sind Absender-Adresse, Empfänger-Adresse und einige weitere Informationen (Betreff, CC, Zeit, Art des Mail-Inhalts (HTML oder reiner Text), verwendeter Zeichensatz, etc.).

Bei einem klassischen Brief ist es das, was sich auf dem Umschlag befindet, also Absender- und Empfänger-Adresse. Aber auch die Briefmarke und den Poststempel könnte man zu den Metadaten zählen. Ebenso wie die Metadaten eines Briefs kann man die Email-Metadaten vor den Mail-Providern nicht verstecken. Sonst können diese die Mail nicht zustellen.

Die Email-Metadaten verraten also, wer wann mit wem und zu welchem Thema kommuniziert. Auch sie sind ein „wertvolles Gut“, an dem nicht nur die Geheimdienste interessiert sind. Wertet man die Metadaten richtig aus, kann man sehr viel über die betreffenden Kommunikationspartner erfahren und daraus ein sehr persönliches Profil erstellen. Deshalb sollte niemand außer den Mail-Providern Zugriff auf die Metadaten der Mails erhalten.

Die Email-Daten sollten im Idealfall auch die Email-Provider (mein Provider und dein Provider) nicht lesen können. Emails sind viel mehr mit Postkarten zu vergleichen. Jeder Postbeamte, der sie in die Finger bekommt, könnte meine Postkarte an dich lesen. Wenn der Postbeamte nicht vertrauenswürdig ist, könnte er auch post-fremden Personen oder Instanzen die Daten und auch die Metadaten der Postkarte zur Verfügung stellen.

Für die Neugierigen unter den Lesern: Wenn wir Mails versenden und empfangen, interessieren uns die Metadaten (außer Absender, Empfänger und Betreff) meist nicht besonders. Die Metadaten der Mail sind technisch gesprochen die sog. Mail-Header. Sie werden vor dem Mail-Anwender meist verborgen. Wer einmal die Metadaten einer empfangenen Mail sehen möchte, muss die Header nur sichtbar machen. In *Thunderbird* wählt man den Menüpunkt *Ansicht* → *Nachrichtenquelltext*. Dann sieht man die gesamte Nachricht „unfrisiert“, die Mail-Header (Metadaten) und den Mail-Body (Daten, bzw. der eigentliche Mail-Inhalt). Header und Body sind durch eine Leerzeile getrennt.

1.4 Sichere Email – Zuständigkeiten

Für eine sichere Mail-Übertragung (einschließlich dem Schutz der Metadaten vor ungebetenen Zaungästen) gibt es vier Verantwortliche:

- Absender
- Provider des Absenders
- Provider des Empfängers

- Empfänger

Für die Mail-Verschlüsselung, die die Mail-Inhalte auch vor den Mail-Providern verbirgt, gibt es zwei Verantwortliche:

- Absender
- Empfänger

Um die Email-Metadaten vor allen Neugierigen außer unseren Providern (vor meinem Provider und vor deinem Provider) zu verbergen, benötigen wir beide vor allem einen kompetenten, zuverlässigen und in erster Linie **vertrauenswürdigen Provider**. Dieses Thema ist nicht so kompliziert und darum (und um einige weitere Hinweise zur Email-Sicherheit) geht es im Kapitel 2 dieses Dokuments.

Um die Email-Daten zu verbergen, müssen wir die Mails verschlüsseln. **Email-Verschlüsselung** (insbesondere die Schlüssel-Verwaltung) ist - vor allem für den IT-Laien – ziemlich kompliziert und ist das Thema im Kapitel 3 dieses Dokuments. Die Verantwortung dafür liegt ausschließlich bei den beiden Kommunikationspartnern.

1.5 Wichtige Grundbegriffe

Einige Begriffe, die ich in den folgenden Ausführungen verwende, sind dem IT-Experten geläufig. Dieses Dokument richtet sich jedoch auch an den IT-Laien. Deshalb will ich die Begriffe *Client und Server*, *Protokoll* sowie *Verschlüsselung* erläutern. Ich erlaube mir dabei einige Vereinfachungen, um die Konzepte besser herauszuarbeiten. Der Experte möge mir das verzeihen. Er kann dieses Kapitel auch überspringen oder selektiv lesen.

1.5.1 Client und Server

In Computer-Netzwerken (in lokalen Netzwerken ebenso wie im Internet) kommunizieren Programme miteinander. Bei der Kommunikation der Programme gibt es meist zwei Rollen, die des Client und die des Servers. Ein Server bietet einen Dienst (Service) an und der Client kann den Dienst in Anspruch nehmen.

1.5.1.1 Spezialisierte Clients und Server

Es gibt verschiedene Arten von Diensten: Web-Dienst, Mail-Dienst und viele weitere. In der Regel ist jeder Server auf einen bestimmten Dienst spezialisiert. Ein Web-Server liefert Web-Seiten aus, ein Mail-Server versendet und empfängt Mails. Beide sind Server, ihre Aufgaben sind jedoch völlig unterschiedlich.

Ebenso sind die Clients auf die Nutzung bestimmter Dienste spezialisiert. Ein Web-Client ist beispielsweise der Browser (*Chrome, Firefox, Safari, Internet Explorer*, etc.); er fordert Web-Seiten beim Web-Server an, empfängt sie und präsentiert sie dem Benutzer in einem Browser-Fenster. Ein Mail-Client (*Thunderbird, Outlook*, etc.) ist darauf spezialisiert, Mails vom Mail-Server des Providers zu empfangen und dem Benutzer darzustellen. Er ermöglicht ihm außerdem, Mails zu verfassen und an den Mail-Server des Providers zu versenden, damit dieser sie an den Empfänger weiterleitet. Für jeden spezialisierten Dienst gibt es ein eigenes Protokoll (siehe Kap. 1.5.2).

1.5.1.2 Client und Server sind Kommunikationsrollen

Meist gibt es für die Server-Rolle und die Client-Rolle ein eigenständiges Programm. Beim Web-Dienst implementiert der Web-Server die Server-Rolle und der Web-Browser implementiert die

Client-Rolle.

Beim Mail-Dienst (insbesondere beim Mail-Versand) ist es allerdings nicht mehr ganz so klar. Das Mail-Programm (z.B. *Thunderbird*) ist in der Rolle des Client und versendet die Mail an den Mail-Server des Mail-Providers des Absenders der Mail. Der Mail-Server des Absenders der Mail schickt die Mail weiter an den Mail-Server des Providers des Empfängers der Mail. In diesem Fall kommunizieren zwei Server miteinander. Gibt es hier überhaupt einen Client und einen Server? Eindeutig ja. Der Mail-Server des Absender-Providers ist in der Rolle des Client und der Mail-Server des Empfänger-Providers ist in der Rolle des Servers. Denn Letzterer bietet den Dienst der Mail-Weiterleitung an und Ersterer nimmt diesen in Anspruch.

1.5.1.3 Kommunikationsphasen

Ein Server bietet in der Regel einen Dienst an und wartet darauf, dass ein Client ihn in Anspruch nimmt. Gibt es keinen aktiven Client, ist der Server untätig. Nutzen viele Clients des Servers Dienste, muss dieser die Clients auch gleichzeitig bedienen können.

Die Kommunikation zwischen einem Client und dem Server findet in drei Phasen statt: Verbindungsaufbau, Datenübertragung und Verbindungsabbau.

Verbindungsaufbau: Möchte ein Client den Dienst des Servers in Anspruch nehmen, baut er zunächst eine Verbindung zu ihm auf. Dazu muss er die Adresse des Servers kennen. Dies ist analog zum Verbindungsaufbau bei einem Telefonat. Der Anrufende (Client) wählt die Telefonnummer (die Adresse im Telefonnetz) des Angerufenen (Servers). Der Angerufene nimmt seinerseits den Hörer ab; der Server nimmt die Verbindungsanfrage des Client an. Jetzt erst steht die Leitung (die Kommunikationsverbindung) und die Kommunikation (Datenübertragung) kann beginnen.

Datenübertragung: In der Datenübertragungsphase „reden“ Client und Server miteinander; d.h. sie tauschen definierte Nachrichten aus. In der Regel tun sie das im Wechsel. Zuerst schickt der Client eine Nachricht mit einer Anfrage; er sendet einen sog. Request an den Server. Der Server bearbeitet die Anfrage und sendet daraufhin eine Nachricht mit einer Antwort, der sog. Response an den Client zurück. Dann folgt wieder der Client mit dem nächsten Request, den der Server wiederum mit einer weiteren Response beantwortet. Der Client agiert, der Server reagiert.

Verbindungsabbau: Hat der Client alle Anfragen gesendet und vom Server alle Antworten erhalten, verabschiedet er sich vom Server und legt auf. Technisch ausgedrückt: der Client baut die Verbindung ab, in dem er dem Server eine Verabschiedungsnachricht sendet und dann die Verbindung beendet. Der Server legt nach dem Empfang des Abschiedsgrußes ebenfalls auf; d.h. auch er beendet die Verbindung. In der Regel initiiert der Client den Verbindungsabbau. Es kann aber auch sein, dass der Server den Verbindungsabbau anstößt, z.B. wenn er auf Grund eines Fehlers die Anfragen des Client nicht mehr beantworten kann.

1.5.2 Protokoll

Wie wir im vorigen Kapitel gesehen haben, gibt es bei der Kommunikation in Netzwerken viele verschiedene Dienste (Services) und zu jedem Dienst spezifische Clients und Server. Damit die Clients und Server eines Dienstes sich verständigen können, sind für jeden Dienst eine Reihe verschiedener Arten von Nachrichten (Request-Nachrichten und Response-Nachrichten) definiert. Würde ein Client einem Server eine Nachricht schicken, die dieser nicht versteht, kann der Server auch keine Antwort senden. Er würde im besten Fall mit einer Fehler-Nachricht antworten. Selbst dies ist möglicherweise nutzlos, da der Client vielleicht nicht einmal die Fehler-Nachricht versteht. Es geht bei der Kommunikation zwischen Client und Server also nicht ohne Vereinbarungen, an die

sich beide Seiten halten müssen. Eine solche Nachrichten-Vereinbarung für bestimmte Anwendungen/Dienste nennt man ein Protokoll.

Ein Protokoll ist die Definition der zwischen zwei Kommunikationsrollen (Client und Server) zulässigen Nachrichten (und Nachrichten-Formate) für ein einen spezifischen Dienst. Nur so können sich beide Seiten verständigen und sinnvoll kooperieren.

Fast jedes Protokoll ist standardisiert und in einem sog. RFC (Request for Comment) technisch beschrieben. Die Entwickler der Server und der Clients müssen das jeweilige Protokoll kennen, um diese (die Server und Clients) korrekt zu implementieren. Für die normalen Anwender spielt die genaue Kenntnis der Protokolle keine Rolle.

Für die vielen unterschiedlichen Dienste gibt nun jeweils ein Protokoll, das festlegt, wie sich die Clients und Server des betreffenden Dienstes verständigen. Um dies ein wenig konkreter zu machen, habe ich beispielhaft einige gängige Protokolle aufgeführt, die der Internetnutzer (vielleicht ohne es zu wissen) täglich nutzt.

- **HTTP (Hypertext Transfer Protocol)** ist das Protokoll zur Übertragung von verlinkten Webseiten (Hypertext). Der HTTP-Client (auch Web-Client) ist der Web-Browser (*Chrome, Firefox, etc.*), den wir täglich zum Surfen benutzen. Der HTTP-Server (auch Web-Server) stellt die Hypertext-Seiten im Web zur Verfügung. Er wird in der URL-Zeile des Browsers angezeigt.
- **FTP (File Transfer Protocol)** ist das Protokoll zur Dateiübertragung. Der Benutzer bedient den FTP-Client an seinem Rechner. Mit diesem Protokoll lassen sich Dateien vom eigenen, lokalen Rechner auf einen fernen FTP-Server oder umgekehrt vom FTP-Server auf den lokalen Rechner übertragen und vieles mehr. Z.B. kann man auch Dateien und Ordner auf dem fernen FTP-Server anlegen, löschen und umbenennen oder den Inhalt eines fernen Ordners anfordern. Der FTP-Server liefert (wenn er durch eine entsprechende protokoll-spezifische Nachricht den Auftrag dazu erhalten hat) die Liste der Dateien im betreffenden Ordner. Der FTP-Client empfängt diese Liste und zeigt sie dem Benutzer an. *FileZilla* ist ein Beispiel für ein solches FTP-Client-Programm. Aber auch der Web-Browser beherrscht das FTP-Protokoll. Gibt man in der URL-Zeile des Browsers eine mit **ftp://** statt **http://** beginnende URL ein, schaltet der Browser automatisch auf das FTP-Protokoll um. Der Browser wird normalerweise als HTTP-Client verwendet, er kann bei Bedarf aber auch als FTP-Client agieren.
- **SMTP (Simple Mail Transfer Protocol)** ist das Protokoll zum Versenden von Emails. Mit diesem Protokoll überträgt der SMTP-Client oder Mail-Client (*Thunderbird, Outlook, etc.*) die Mail an den SMTP-Server oder Mail-Server des Providers. SMTP kommt auch zum Einsatz, wenn der SMTP-Server des Absender-Providers eine Mail an den SMTP-Server des Empfänger-Providers weiterleitet. Ersterer ist dabei (wie oben schon gezeigt) in der Rolle des Client, Letzterer in der Rolle des Servers.
- **POP (Post Office Protocol)** ist ein Protokoll zum Abruf von Mails. Der Mail-Client ist also nicht nur SMTP-Client. Da er auch das Protokoll POP unterstützt, ist er gleichzeitig ein POP-Client. Er ruft die Mails, die auf dem Mail-Server des Providers eingegangen sind, ab und überträgt sie auf den lokalen Rechner des Nutzers. Der Mail-Server des Providers kann also auch als POP-Server agieren, er beherrscht das POP-Protokoll. POP wird heute auf Grund seines eingeschränkten Funktionsumfangs nur noch relativ selten verwendet.
- **IMAP (Internet Mail Access Protocol)** ist heute als Nachfolger von POP weit verbreitet. Dieses Protokoll ist wesentlich leistungsfähiger. Es ermöglicht nicht nur den Abruf von

eingegangenen Mails. Mit diesem Protokoll kann man die gesamte Mail-Ordnerstruktur auf dem Mail-Server verwalten. Man kann auf dem Mail-Server Mail-Ordner anlegen, umbenennen und löschen; man kann Mails von einem Ordner in einen anderen verschieben; auch die Mails kann man umbenennen und löschen. Wenn man nicht weiß, in welchem Unterordner sich eine Mail befindet, kann man mit IMAP auch Mails auf dem Mail-Server suchen lassen. Der Mail-Server agiert dabei als IMAP-Server und führt all diese Operationen auf Geheiß des Clients aus. Der Mail-Client (*Thunderbird*, *Outlook*, etc.) unterstützt (zusätzlich zu SMTP und POP) auch IMAP. Als IMAP-Client sendet er dem IMAP-Server IMAP-Nachrichten, die diesen veranlassen, die gewünschten Operationen im Auftrag des Client auszuführen.

Wie wir gesehen haben, muss ein Mail-Server ebenso wie ein Mail-Client drei Protokolle – SMTP, IMAP und POP – unterstützen.

Es gibt viele weitere Protokolle, von denen der IT-Laie meist noch nichts gehört hat. So ist z.B. das SNMP (Simple Network Management Protocol) ein Protokoll zur Verwaltung von Computer-Netzwerken.

1.5.3 Verschlüsselung

„Verschlüsselung“ heißt das Zauberwort, wenn es darum geht, Daten vor neugierigen Blicken zu schützen.

Was ist unter Verschlüsselung zu verstehen? Sehen wir uns dazu ein kleines Beispiel an:

1.5.3.1 Verschlüsselung einer Datei – ein Beispiel

Dies ist der Inhalt der Datei *gedicht.txt*:

```
Es war einmal ein Wiesel.
Das saß auf einem Kiesel.
Es sagte sich im Stillen:
Ich tu's um Reimes willen.
```

Mit folgendem Kommando kann ich die Datei *gedicht.txt* verschlüsseln.

```
openssl des3 -in gedicht.txt -out gedicht-chiffriert.bin
```

Das Kommando verlangt von mir die Eingabe eines Verschlüsselungspassworts, das zum Entschlüsseln benötigt wieder wird. Beim Verschlüsseln (oder Chiffrieren) wird das Ergebnis der Verschlüsselung (das Chifftrat) in die Datei *gedicht-chiffriert.bin* geschrieben. Das Chifftrat ist ein für niemanden verständlicher Kauderwelsch und sieht (wenn man den Dateinhalt ausgibt) in diesem Fall so aus:

```
Salted__
?=2?r?*?s???.??q?]??#?8?Ad_T!1Nt[]??mo?????B?^*
??X???l6??M?}???q?
```

Die Datei *gedicht.txt* enthält den unverschlüsselten Text, der verschlüsselte Text ist in *gedicht-chiffriert.bin* enthalten. Lösche ich *gedicht.txt*, so ist der unverschlüsselte Text nicht mehr verfügbar. Der chiffrierte Text in *gedicht-chiffriert.bin* ist für niemanden nutzbar, es sei denn er kennt das Passwort.

Will ich den unverschlüsselten Text wieder anzeigen, muss der Inhalt der Datei *gedicht-chiffriert.bin* wieder entschlüsselt werden. Dies geht mit folgendem Kommando:

```
openssl des3 -d -in gedicht-chiffriert.bin
```

Dieses Kommando fragt mich wieder nach dem Passwort, das ich beim Verschlüsseln eingegeben habe. Bei Eingabe des richtigen Passwortes wird mir der unverschlüsselte (dechiffrierte) Text wieder ausgegeben:

```
Es war einmal ein Wiesel.  
Das saß auf einem Kiesel.  
Es sagte sich im Stillen:  
Ich tu's um Reimes willen.
```

Dies ist ein einfaches Beispiel, das das Prinzip verdeutlichen soll. Statt eines Passwortes wird in der Regel ein digitaler Schlüssel verwendet. Doch der digitale Schlüssel ist im Grunde nichts anderes als ein sehr, sehr langes Passwort, das natürlich nicht für die manuelle Eingabe gedacht ist.

Mehrere digitale Schlüssel werden in einem sog. Schlüsselbund gespeichert, der durch ein Benutzerpasswort gesichert ist. Soll ein Programm einen der Schlüssel des Schlüsselbundes zum Chiffrieren oder zum Dechiffrieren benutzen, muss der Benutzer durch die Eingabe des Passwortes den Schlüsselbund entsperren, um dem Programm Zugriff auf den Schlüssel zu gewähren.

1.5.3.2 Transport-Verschlüsselung vs. Daten-Verschlüsselung

Sind Daten im Internet unterwegs, sind sie grundsätzlich unverschlüsselt und für jeden lesbar (und evtl. auch manipulierbar). Um dies zu verhindern, muss man sie verschlüsseln. Dabei sind zwei Verschlüsselungsarten grundsätzlich zu unterscheiden: die Transport-Verschlüsselung und die Daten-Verschlüsselung.

Bei der Transport-Verschlüsselung ist der Übertragungskanal verschlüsselt, durch den die Daten transportiert werden. Die Daten selbst können unverschlüsselt oder verschlüsselt sein.

Bei der Daten-Verschlüsselung sind (der Name sagt es) die Daten verschlüsselt. Der Übertragungskanal kann unverschlüsselt oder verschlüsselt sein.

Beide Verschlüsselungsarten können (rein technisch gesehen) unabhängig voneinander oder kombiniert eingesetzt werden. Es ist also durchaus möglich, verschlüsselte Daten (z.B. Mails) über einen (oder mehrere) verschlüsselten Kanal zu übertragen.

Das Beispiel im vorigen Kapitel 1.5.3.1 war übrigens ein Beispiel für die Daten-Verschlüsselung. Die Daten der Datei *gedicht.txt* wurden dabei nicht transportiert, also nicht von einem auf einen anderen Rechner übertragen.

1.5.3.3 Transport-Verschlüsselung

Bei der Transport-Verschlüsselung – das sagt schon der Begriff – sind nicht die Daten selbst verschlüsselt, sondern der Übertragungsweg, auf dem die Daten transportiert werden. Man kann sich vorstellen, die Daten fließen durch einen undurchsichtigen Tunnel. Beim Eintritt in den Tunnel werden sie automatisch chiffriert und beim Austritt automatisch dechiffriert. Solange die Daten im Tunnel unterwegs sind, sind sie nicht einsehbar. Bevor sie in den Tunnel hineinfließen und nachdem sie wieder herausfließen, sind die Daten aber sehr wohl einsehbar, da die Daten selbst nicht verschlüsselt sind.

Transport-Verschlüsselung kommt heute häufig zum Einsatz, zum Beispiel im Web, bei der Kommunikation zwischen Web-Browser und Web-Server oder auch beim Emailversand und -empfang. Man spricht auch von einem **verschlüsselten Übertragungskanal**.

Der verschlüsselte Übertragungskanal wird übrigens in der Phase des Verbindungsaufbaus (siehe Kap. 1.5.1.3) hergestellt. Danach (in der Datenübertragungsphase) ist die gesamte Kommunikation in beide Richtungen (vom Client zum Server und umgekehrt) verschlüsselt. Beim Verbindungsabbau wird der verschlüsselte Kanal wieder zerstört.

Die Kommunikation zwischen Web-Browser und Web-Server wird durch das HTTP-Protokoll (siehe Kap. 1.5.2) definiert (**H**ypertext **T**ransfer **P**rotokoll). **HTTPS (HTTP Secure)** ist die transport-verschlüsselte Variante des HTTP-Protokolls. Jeder Internetnutzer kennt das. Sobald wir uns mit dem Browser an einem Dienst (z.B. Bank-Account, Email-Account, Account bei der Deutschen Bahn, Account bei einem Online-Shop wie Amazon oder Zalando, etc.) mit Benutzernamen und Passwort anmelden, schaltet der Browser in der Regel automatisch um auf HTTPS. Wir erkennen dies an der URL-Zeile des Browsers: die URL beginnt mit **https://** statt mit **http://**. Zusätzlich zeigt der Browser durch das Symbol eines Schusses an, dass er die Daten mit dem Web-Server über einen verschlüsselten Kanal austauscht. Heute läuft auch jede Google-Suche über einen verschlüsselten Übertragungskanal.

In diesem Dokument geht es um die Mail-Übertragung. Anders als beim Surfen im Web wird hier nicht ein Übertragungskanal verwendet sondern drei Kanäle, da eine Mail vom Absender zum Empfänger über drei Teilstrecken transportiert wird (siehe Kap. 2.1). Die erste der Teilstrecken ist der Transport vom Absender zum Provider des Absenders. Die zweite geht vom Provider des Absenders zum Provider des Empfängers. Die dritte Teilstrecke schließlich ist die vom Provider des Empfängers zum Empfänger. Jeder dieser drei Übertragungskanäle kann grundsätzlich unverschlüsselt oder verschlüsselt sein. Außerdem ist die Mail (wenn ihre Daten nicht verschlüsselt wurden) auf jeder Zwischenstation – also beim Provider des Absenders und beim Provider des Empfängers – lesbar.

Der Transport der Mail (SMTP) ist also durchaus kritischer zu sehen als die Web-Kommunikation mit dem Browser (HTTP). Z.B. kann die Web-Kommunikation zwischen mir und meiner Bank viel leichter abgesichert werden, da es sich nur um eine einzige Transportstrecke handelt, die durch einen verschlüsselten Übertragungskanal gesichert wird. Bei der Mail-Übertragung zwischen mir (dem Absender) und meinem Kommunikationspartner (dem Empfänger) ist der Transportweg unterbrochen. Selbst wenn alle drei Teilstrecken verschlüsselt sind, eine Ende-zu-Ende-Verschlüsselung für die gesamte Übertragungsstrecke vom Absender bis zum Empfänger kann mit der Transport-Verschlüsselung allein prinzipbedingt nicht sichergestellt werden.

Transport-Verschlüsselung ist dennoch auch für die Mailübertragung wichtig. Sie ist außerdem für den Benutzer auch nicht so schwierig anzuwenden. Beim Surfen im Web schaltet der Browser in der Regel automatisch auf verschlüsselte Übertragung um, spätestens wenn man sich bei einem Dienst anmeldet. Bei der Mail-Übertragung ist heute (vor wenigen Jahren war das noch anders) die verschlüsselte Übertragung zwischen dem Mail-Programm auf meinem Rechner und dem Mail-Server des Providers in der Regel die Standardeinstellung. Unverschlüsselte Übertragung funktioniert meist gar nicht mehr. Auch ohne viel davon zu verstehen, macht der Benutzer bei den Einstellungen des Mail-Programms meist nichts verkehrt.

Allerdings braucht man einen vertrauenswürdigen Provider, denn dieser kann immer auf die Mails zugreifen, solange die Mail-Daten nicht verschlüsselt sind. Dies ist Thema in Kapitel 2.

1.5.3.4 Daten-Verschlüsselung

Verschlüsselung ist einfach, denn das erledigen Programme. Die Schlüssel-Verwaltung ist kompliziert, denn das ist Aufgabe des Benutzers. Dieser muss wissen, was er tut und benötigt dazu ein gewisses Know-how. Schon bei dem einfachen Beispiel aus Kapitel 1.5.3.1 musste sich der

Benutzer das Passwort (das war der Schlüssel) merken. Würde er es vergessen, könnte er das verschlüsselte Gedicht nicht mehr entschlüsseln.

Transport-Verschlüsselung ist deshalb relativ einfach, da hier die Benutzer die Verschlüsselung implizit verwenden, ohne technisch viel davon verstehen zu müssen. Die Benutzer benötigen keine eigenen Schlüssel und sie müssen sie auch nicht verwalten.

Bei der Daten-Verschlüsselung wird – im Kontext der Mail-Übertragung – der Inhalt einer Email verschlüsselt. Die verwendeten Teilstrecken der Übertragung können unverschlüsselt oder verschlüsselt sein. Es werden auch nur die Daten der Email (der Inhalt der Mail) verschlüsselt, die Metadaten (Absender-Adresse, Empfänger-Adresse, Betreff, etc.) bleiben immer unverschlüsselt (siehe Kap. 1.3). Mit der Verschlüsselung der Email-Daten lässt sich die erstrebenswerte **Ende-zu-Ende-Verschlüsselung** realisieren. Der Inhalt der Mails bleibt garantiert privat.

Ohne Verschlüsselung des Inhaltes entspricht das Niveau der Vertraulichkeit einer Email dem einer Postkarte. Ist der Inhalt einer Mail verschlüsselt, ist der Grad der Vertraulichkeit höher als der eines versiegelten Briefes. Das Siegel eines Briefes kann man brechen und den Brief trotzdem lesen. Eine korrekt verschlüsselte Mail ist nicht zu knacken. Sie kann nur vom Absender und vom Empfänger gelesen werden. Diese beiden sitzen an den Enden des gesamten, mehrteiligen Kommunikationsweges. Deshalb spricht man von **Ende-zu-Ende-Verschlüsselung**.

Bei der Verschlüsselung von Mails muss jeder Kommunikationspartner ein Schlüsselpaar erzeugen. Er muss den eigenen öffentlichen Schlüssel exportieren, die öffentlichen Schlüssel seiner Kommunikationspartner in seinen Schlüsselbund importieren und beglaubigen. Der Benutzer benötigt ein Grundverständnis der asymmetrischen Verschlüsselung und davon, wie der Schlüsselaustausch zwischen den Kommunikationspartnern funktioniert. Dies stellt natürlich eine ziemliche hohe Einstiegshürde für den technisch nicht versierten Benutzer dar. Im Kapitel 3 erläutere ich dieses Thema detailliert und versuche die Einstieg in den Versand und Empfang verschlüsselter Mails zu erleichtern.

1.6 Wer mehr wissen will ...

In diesem Dokument möchte ich euch, meinen Kommunikations-Partnern, die wichtigsten Informationen für eine sichere Email-Kommunikation bieten. Dabei versuche ich, mich auf das Wichtigste zu beschränken und dies so darzustellen, dass es auch für den IT-Laien möglichst verständlich ist. (Es ist trotzdem eine ganze Menge Text. ;-))

Auch habe ich manche Sachverhalte etwas vereinfacht, damit sie leichter verdaulich sind. Der IT-Profi, der sich mit der Materie auskennt, möge mir das verzeihen.

Natürlich enthält dieses Dokument nicht alles, was es zum Thema zu sagen gibt. Der Heise-Verlag hat (sicherlich auch aus aktuellem Anlass) ein c't-Sonderheft mit dem Titel „Sichere E-Mail – NSA aussperren – Privates schützen“ herausgegeben. Ich habe das Heft gelesen und kann es jedem empfehlen, der sich tiefer mit der Materie auseinandersetzen will.

Das Heft kann in Papierform oder als PDF bestellt werden unter folgender Web-Adresse:

<http://shop.heise.de/katalog/ct-wissen-sichere-e-mail>

Über diese Web-Adresse kann man auch das Inhaltsverzeichnis des Heftes einsehen und sich einen ersten Überblick verschaffen.

An einigen Stellen des vorliegenden Dokuments werde ich mich auf dieses c't-Sonderheft beziehen.

1.7 Links zu diesem Kapitel

- „Im NSA-Skandal ist ein langer Atem gefragt!“, ein Kommentar von Peter Schar, Bundesbeauftragter für Datenschutz und Informationsfreiheit von 2003 bis 2013:
<http://www.heise.de/newsticker/meldung/Kommentar-Im-NSA-Skandal-ist-ein-langer-Atem-gefragt-2214717.html>
- „Es ist Zeit, die Netze zurückzuerobern“, ein Kommentar von Erich Moechel, Journalist mit den Schwerpunktthemen Datenschutz, Datensicherheit, Verschlüsselung und militärische IT:
<http://www.heise.de/newsticker/meldung/Kommentar-zum-NSA-Skandal-Es-ist-Zeit-die-Netze-zurueckzuerobern-2216016.html>
- „Die technologische Souveränität zurückgewinnen“, ein Kommentar von Dorothee Bär, Bundestagsabgeordnete der CSU und Parlamentarische Staatssekretärin beim Bundesministerium für Verkehr und digitale Infrastruktur:
<http://www.heise.de/newsticker/meldung/Kommentar-zum-NSA-Skandal-Die-technologische-Souveraenitaet-zurueckgewinnen-2216143.html>
- „Ein Jahr NSA-Skandal und noch viel zu tun“, ein Kommentar von Christoph Wegener, IT-Leiter der Fakultät für Elektrotechnik und Informationstechnik an der Ruhr-Universität Bochum und freiberuflich in den Bereichen Informationssicherheit und Datenschutz tätig:
<http://www.heise.de/newsticker/meldung/Analyse-Ein-Jahr-NSA-Skandal-und-noch-viel-zu-tun-2215730.html>
- c't-Sonderheft mit dem Titel „Sichere E-Mail – NSA aussperren – Privates schützen“:
<http://shop.heise.de/katalog/ct-wissen-sichere-e-mail>

2 (Möglichst) Sicherer Versand unverschlüsselter Mails

Um Emails wasserdicht sicher zu versenden (dies wäre sicherer als ein versiegelter Brief), müssen wir sie verschlüsseln. (Siehe Kap. 3) Verschlüsselung von Mails (insbesondere die Schlüsselverwaltung) ist kompliziert und die Metadaten der Mails bleiben dabei trotzdem einsehbar, da die Mail-Provider ohne die Metadaten (Absender- und Empfängeradresse, Betreff, etc.) die Mail nicht zustellen können.

Eine sehr hohe Sicherheit – und damit die Minimierung des Risikos der Kompromittierung der Mails – kann man aber auch erreichen, wenn man unverschlüsselte Mails über sichere Transportwege versendet. Im Idealfall sind die Teilstrecken auf dem Transportweg und die Lagerung bei den Mail-Providern so sicher, dass nur Absender, Empfänger und die beiden Provider darauf zugreifen können (siehe Kap. 2.1). Für diese Sicherheitsstufe ist die Wahl des richtigen Mail-Providers von zentraler Bedeutung (siehe Kap. 2.2).

Da die Mail in diesem Fall unverschlüsselt ist, ist sie nach wie vor nicht mit einem Brief, sondern mit einer Postkarte zu vergleichen, die aber nur von vertrauenswürdigen Kurieren transportiert wird.

Ich (Absender) schreibe die Postkarte und lasse mir dabei von keinem über die Schulter schauen. Dann trage ich sie zum Briefkasten. Mit dem Einwurf in den Kasten übergebe ich die Karte an meinen Provider. Wenn ich die Postkarte nicht verliere und sie niemandem zeige, sind die Daten und die Metadaten sicher. Die Deutsche Post (mein Postkarten-Provider) ist vertrauenswürdig (gehen wir mal davon aus), die Postangestellten und auch die Sortiermaschinen der Deutschen Post lesen zwecks Zustellung nur die Adresse der Postkarte, aber nicht deren Inhalt (selbst wenn sie es könnten). Und sie geben die Adressdaten auch nicht weiter, nicht an Kriminelle und nicht an Geheimdienste oder andere Staatsorgane.

Nehmen wir weiter an, du (Empfänger) lebst in Frankreich und dein Post-Provider ist die französische Post. Die Deutsche Post übergibt die Postkarte an die französische Post. Bei der Übergabe bekommt kein anderer die Postkarte in die Finger. Nehmen wir an, die französische Post ist genau so vertrauenswürdig. Sie liest nicht den Inhalt, fertigt keine Kopie an, zeigt die Karte keiner anderen Person oder Instanz. Auch die französische Post sieht sich nur die Adresse an, um die Karte in deinem Posteingang abzulegen, d.h. in deinen Briefkasten zu werfen. Du holst die Karte aus dem Kasten, liest sie und hältst sie auch keiner anderen Person unter die Nase.

Wenn unsere beiden Postdienst-Provider ihre Neugier in Zaum halten konnten und wir selbst die Karte niemandem vorlesen oder sie herumliegen lassen, kennen nur du und ich ihren Inhalt.

Das ganze Verfahren steht und fällt mit der **Vertrauenswürdigkeit beider Provider**. Die Provider müssen professionell arbeiten, sodass die Postkarte nicht verloren geht und nicht in die falschen Hände gerät. Die Provider sollten kein Interesse am Inhalt der Karte haben. Ich bezahle den Transport der Karte mit dem Porto. Also müssen sich die beiden Provider nicht dadurch finanzieren, dass sie die Adressdaten oder den Inhalt meiner Karte lesen und auswerten oder an andere mögliche Interessierte verkaufen. Dies gilt vergleichbar für die Mail-Provider.

2.1 Die Mail auf dem Transportweg

Der gesamte Transportweg einer Mail kann in verschiedene Teilstrecken aufgeteilt werden. Auch wird die Mail auf dem Transportweg bei den Providern zwischengespeichert. Um den gesamten Weg sicher zu machen, muss die Mail auf den drei Teilstrecken durch verschlüsselte Kanäle übertragen und bei den Providern sicher zwischengespeichert werden.

Im Folgenden ist häufig von *verschlüsselten Übertragungskanälen* die Rede. Der IT-Laie kann sich darunter meist nichts vorstellen. Ich will das kurz mit einem Bild erläutern: Ein *verschlüsselter Übertragungskanal* kann mit einer undurchsichtigen Röhre (oder mit einer nicht abhörbaren Leitung) verglichen werden. Alle Informationen, die durch die Röhre fließen, sind nur für die beiden Kommunikationspartner sichtbar, die diese Röhre als Kommunikationsverbindung zwischen sich aufbauen. Für Außenstehende ist die übertragene Information nicht einsehbar und nicht manipulierbar.

Ein *unverschlüsselter Kanal* ist mit einer durchsichtigen Röhre (oder mit einer abhörbaren Leitung) zu vergleichen, bei der Außenstehende alle durchfließenden Informationen mitlesen und möglicherweise auch manipulieren können.

Das herkömmliche Internet (einschließlich Email-Verkehr) ist ein Gewirr von unverschlüsselten Kanälen. Erst so langsam beginnen sich die verschlüsselten Kanäle durchzusetzen.

Technisch wird ein verschlüsselter Kanal durch das Protokoll SSL (Secure Socket Layer), bzw. durch das neuere TLS (Transport Layer Security) implementiert und bereitgestellt. Beide Kommunikationspartner müssen diese Protokolle beherrschen, um einen verschlüsselten Transportweg aufzubauen. Auf weitere technische Details verzichte ich hier. Mehr dazu in Kap. 2.1.1 und 5.1.

Bei Emails sichert ein verschlüsselter Kanal immer nur eine Teilstrecke des gesamten Übertragungsweges vom Absender zum Empfänger. Mit verschlüsselten Kanälen kann keine Ende-zu-Ende-Verschlüsselung realisiert werden, bei der der Inhalt der Mail auf dem gesamten Übertragungsweg vom Absender zum Empfänger verschlüsselt ist (siehe Kap. 3).

Unter *Kompromittierung einer Mail* verstehe ich, dass entweder der Inhalt der Mail von anderen Personen oder Instanzen als dem Absender oder dem Empfänger gelesen oder verändert wird oder dass die Metadaten von anderen Personen oder Instanzen als dem Absender oder dem Empfänger oder den beiden Mail-Providern gelesen werden.

Wo kann die Mail auf dem Weg von mir (Absender) zu dir (Empfänger) kompromittiert werden? Welche Angriffsmöglichkeiten gibt es?

Grundsätzlich ist eine Mail vor dem Versand (Schritt 1) und nach dem Empfang (Schritt 7) auf allen drei Teilstrecken (Schritte 2, 4 und 6) und bei der Zwischenspeicherung bei den Providern (Schritte 3 und 5) kompromittierbar. Die einzelnen Schritte des Transportweges in Abbildung 1 dargestellt:

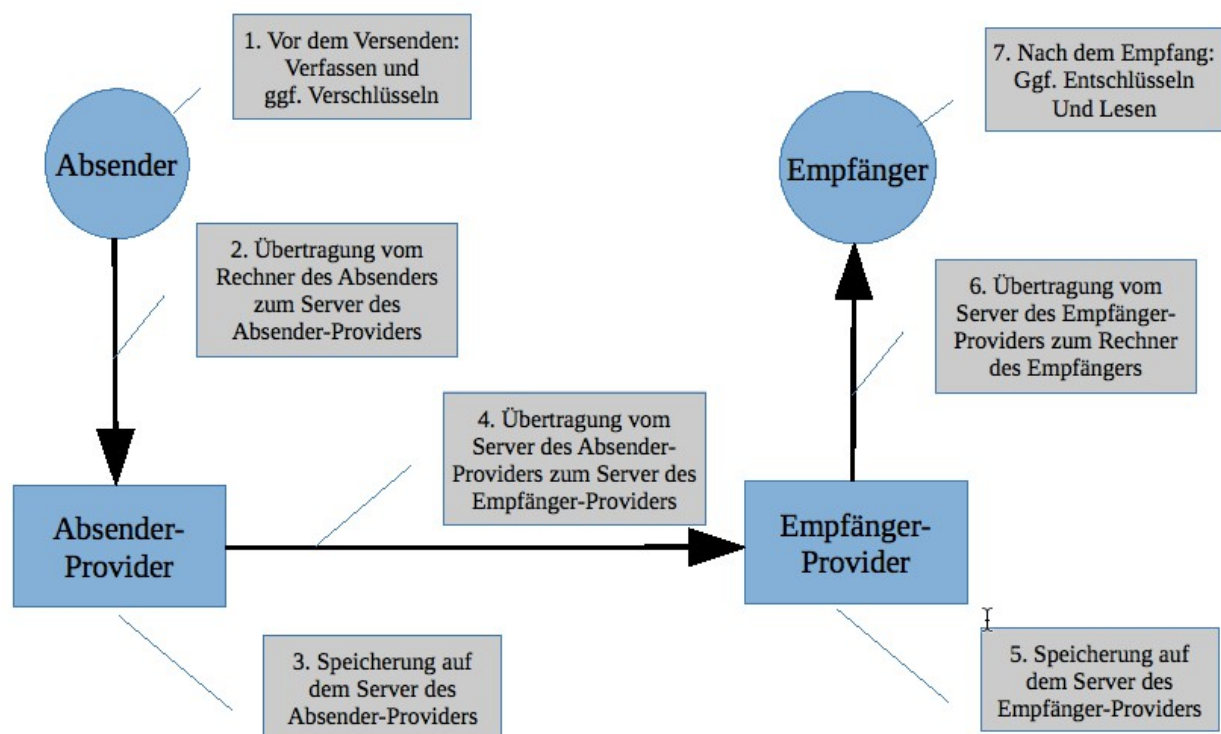


Abbildung 1: Die Mail auf ihrem Transportweg

- 1. Der Rechner des Absenders vor dem Versenden:** Ist mein Rechner mit einem Virus infiziert, könnte dieser die Kopie der Mail auf der Festplatte meines Rechners an einen Cyberkriminellen im Internet senden. Damit dies möglichst nicht geschieht, muss ich meinen Rechner virenfrei halten und dafür sorgen, dass die Software des Rechners immer auf dem aktuellen Stand ist.
 - 1a.** Wird eine Mail verschlüsselt (siehe Kapitel 3), geschieht dies, bevor ich sie an meinen Provider zur Weiterleitung an dich übergebe. Sie kann dann erst nach der Zustellung an dich (vor Schritt 7) und nur von dir entschlüsselt werden.
- 2. Der Übertragungsweg von Absender-Rechner zum Server des Absender-Providers:** Die Mail wird per SMTP (Simple Mail Transfer Protocol) übertragen. Beim Versand der Mail erfolgt eine Anmeldung beim Provider mit meiner Benutzerkennung und meinem Passwort. Danach erst wird die Mail zum Provider übertragen. Sowohl die Login-Daten, als auch die Mail müssen über einen verschlüsselten Übertragungskanal übertragen werden. Mein Provider muss einen verschlüsselten Kanal anbieten und ich muss ihn auch nutzen. D.h. ich muss in meinem Mail-Programm die richtigen Einstellungen vornehmen. Unverschlüsselte Kanäle werden von den Mail-Providern heute meist nicht mehr angeboten. Anfang 2014 haben auch die großen Anbieter in Deutschland (WEB.DE, GMX, 1&1, Freenet, Telekom) auf die ausschließliche Verwendung verschlüsselter Kanäle umgestellt.
- 3. Die Speicherung der Mail beim Provider des Absenders:** In der Regel wird eine Kopie der Mail auf dem Server des Providers im Ordner „Gesendet“ gespeichert. Von dort kann ich die gesendete Mail wieder abrufen, falls ich sie nochmal lesen will. Meine Mails sicher zu speichern, ist der Job meines Providers. Mein Provider sollte meine Mails nicht scannen und

auswerten. Google, jedoch nicht nur Google, tut genau das. Mein Provider muss seine Server auch sicher verwalten, damit weder Geheimdienste noch Cyberkriminelle dort eindringen können und Zugriff auf meine Mails erhalten.

4. **Der Übertragungsweg von Absender-Provider zum Empfänger-Provider:** Die Mail-Übertragung zwischen den Providern muss über einen verschlüsselten Übertragungskanal erfolgen. Das ist Sache der Provider. Es funktioniert nur, wenn beide Provider den Aufbau eines verschlüsselten Übertragungskanals unterstützen. Wir Benutzer erhalten von den Providern meist keine Informationen dazu. Natürlich ist Schritt 4 gar nicht in Betracht zu ziehen, wenn Absender und Empfänger ihren Account bei dem selben Provider haben. Haben beide ihr Mail-Konto beispielsweise bei Google, ist keine Übertragung der Mail zwischen den Providern erforderlich.
5. **Die Speicherung der Mail beim Empfänger-Provider:** Der Empfänger-Provider speichert die Mail im Posteingang deines Mail-Accounts auf seinem Server ab. Von dort kannst du sie dann abrufen. Deine Mails sicher zu speichern, ist der Job deines Providers. Dein Provider sollte deine Mails nicht scannen und auswerten. Google, jedoch nicht nur Google, tut genau das. Dein Provider muss seine Server auch sicher verwalten, damit weder Geheimdienste noch Cyberkriminelle dort eindringen können und Zugriff auf deine Mails erhalten.
6. **Der Übertragungsweg von Empfänger-Provider zum Empfänger-Rechner:** Beim Abrufen der Mail aus dem Posteingang wird die Mail per POP3 (Post Office Protocol, Version 3) oder per IMAP (Internet Message Access Protocol) übertragen. Dabei erfolgt eine Anmeldung beim Provider mit deiner Benutzerkennung und deinem Passwort. Danach erst wird die Mail von deinem Provider zu dir übertragen. Sowohl die Login-Daten, als auch die Mail müssen über einen verschlüsselten Kanal übertragen werden. Dein Provider muss einen verschlüsselten Kanal anbieten und du musst ihn auch nutzen. D.h. du musst in deinem Mail-Programm die richtigen Einstellungen vornehmen. Unverschlüsselte Kanäle werden von den Mail-Providern heute meist nicht mehr angeboten. Anfang 2014 haben auch die großen Anbieter in Deutschland (web.de, GMX, 1&1, Freenet, Telekom) auf die ausschließliche Verwendung verschlüsselter Kanäle umgestellt.
 - 6a. Wurde eine Mail vom Absender verschlüsselt (Schritt 1a), so wird sie jetzt auf dem Rechner des Empfängers entschlüsselt, damit sie von diesem gelesen werden kann. Nur der Empfänger besitzt den Schlüssel zur Entschlüsselung der Mail, nur er kann sie entschlüsseln.
7. **Der Rechner des Empfängers nach dem Empfang:** Ist dein Rechner mit einem Virus infiziert, könnte dieser die Kopie der Mail auf der Festplatte deines Rechners an einen Cyberkriminellen im Internet senden. Damit dies möglichst nicht geschieht, musst du deinen Rechner virenfrei halten und dafür sorgen, dass die Software des Rechners immer auf dem aktuellen Stand ist.

2.1.1 Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?

Verschlüsselte Übertragungskanäle bieten nur relative Sicherheit.

Werden die Daten, die auf einem verschlüsselten Transport-Kanal übertragen werden, mitgelesen, so bekommt der Mitlesende nur einen unverständlichen Kauderwelsch zu sehen. Wäre er im Besitz des Schlüssels, mit dem die Übertragung verschlüsselt wurde, könnte er den Datenkauderwelsch

entschlüsseln und die übertragenen Inhalte im Klartext mitlesen.

Genau das versuchen sowohl Geheimdienste wie die NSA und GCHQ als auch Hacker. Sie versuchen die verschlüsselten Kanäle anzugreifen. Dies geht umso leichter, je schlechter die Verschlüsselung eines Transport-Kanals implementiert ist. Je besser die Verschlüsselung des Kanals desto schwieriger ist es, sie zu aufzubrechen. Bei der Mail-Übertragung ist es Sache der Provider, die Kanäle optimal zu verschlüsseln und damit die Hürden für die Kompromittierung des Kanals möglichst hoch zu setzen.

Für „gut gemachte“ Verschlüsselung gibt es einige Qualitäts-Kriterien, an denen man auch die Mail-Provider messen kann:

- Unterstützung der neuesten TLS-Version 1.2 (siehe Kap. 5.1.1)
- Unterstützung von PFS: Mit diesem Verfahren lässt sich die nachträgliche Entschlüsselung von aufgezeichneter, verschlüsselter Kommunikation zu verhindern. (siehe Kap. 5.1.2)
- Unterstützung von HSTS für die Webmail-Schnittstelle: Dieses Verfahren erzwingt die Verwendung von HTTPS im Browser, auch wenn der unbedachte Benutzer im Browser via HTTP auf seine Mails zugreifen will. (siehe Kap. 5.1.3)
- Unterstützung von DANE: Dieses noch recht neue Verfahren erschwert die Korruption von Zertifikaten, auf denen die Transport-Verschlüsselung basiert. (siehe Kap. 5.1.4)

Für die Bewertung der Mail-Provider sind diese Kriterien wichtig. Dazu muss der IT-Laie aber nicht unbedingt verstehen, was sich technisch hinter diesen Akronymen verbirgt. Ich habe diese eher technischen Aspekte im Kapitel Fehler: Referenz nicht gefunden etwas näher erläutert.

2.1.2 Die Verschlüsselungsqualität der Mail-Provider prüfen

Wie gut die Qualität der SSL/TLS-Verschlüsselung wirklich ist, das lässt sich auch prüfen: Dazu gibt man im Browser die URL: <https://starttls.info> ein. Auf der Seite erscheint ein Eingabefeld, in das man eine Email-Adresse oder mit einer Email-domain (dem Suffix der Email-Adresse hinter dem @-Symbol) eingeben kann.

Ich habe das einmal mit meiner Email-Domain *secure.mailbox.org* ausprobiert und das Ergebnis in Abbildung 2 erhalten.



Abbildung 2: starttls.info: Abfrage von secure.mailbox.org (Übersicht)

Für jeden Server ist eine detaillierte Ansicht verfügbar (siehe Abbildung 3).

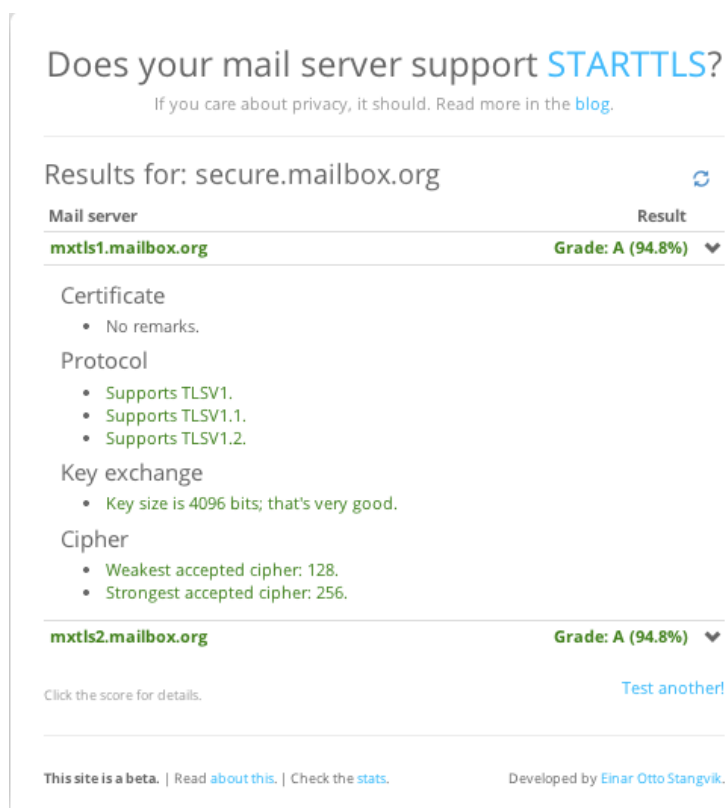


Abbildung 3: starttls.info: Abfrage von secure.mailbox.org (Detail-Ansicht für ersten Server)

Es bietet sich an, die Email-Domains verschiedener Provider zu vergleichen. Ich habe diesen Vergleich für einige bekannte Provider und auch für die Favoriten aus Kapitel 2.2.2 durchgeführt. Die nachstehende Tabelle zeigt das Ergebnis meiner Tests am 18.07.2014.

Email-Domain	Provider	Score
--------------	----------	-------

gmail.com	Google	90,6 %
outlook.com	Microsoft	90,6 %
icloud.com	Apple	79,4 %
yahoo.com	Yahoo	89,2 %
web.de	WEB.DE	90,6 %
gmx.net	GMX	90,6 %
freenet.de	Freenet	Error: Could not connect (timeout)
telekom.de	Telekom	48,8 %
mykolab.com	MyKolab	39,0 %
posteo.de	Posteo	Error: Connection rejected
jpberlin.de	JPBerlin	33,0 %
mailbox.org	mailbox.org	82,0 %
secure.mailbox.org	mailbox.org	94,8 %
mail.de	mail.de	90,6 %

Die Werte können sich natürlich ändern, wenn die Provider die SSL/TLS-Implementierung für die Verschlüsselung ändern oder verbessern. Den Test kann man für den eigenen Mail-Provider durchführen oder mehrere Provider vergleichen, bevor man den Provider zu wechselt. Er kann auch ein Auswahlkriterium im Provider-Vergleich (siehe Kap. 2.2.1) sein.

2.2 Der richtige Email-Provider

Wie wir gesehen haben, ist ein vertrauenswürdiger, professionell arbeitender Email-Provider die Grundvoraussetzung für eine sichere Email-Kommunikation. Hier sind meine Auswahlkriterien für die Provider-Wahl, zu der ich insbesondere auch den Artikel „*Email-Provider im Test*“ aus dem c't-Sonderheft „*Sichere E-Mail*“ auf Seite 20 herangezogen habe. Somit muss man sich nicht nur auf die Versprechungen verlassen, die die Provider auf ihren Websites abgeben. Zum Bezug des Heftes, siehe Kap. 1.6.

2.2.1 Auswahlkriterien

1. Der Mail-Provider sollte zur Transport-Verschlüsselung die neueste TLS-Version 1.2 unterstützen. (Ältere Versionen werden aus Gründen der Abwärtskompatibilität meist zusätzlich unterstützt: SSLv3, TLSv1, TLSv1.1)
2. Der Mail-Provider sollte PFS unterstützen. (siehe Kap. 2.1.1)
3. Der Mail-Provider sollte HSTS auf seiner Web-Site verwenden (siehe Kap. 2.1.1)
4. Der Mail-Provider sollte kein Interesse an den Inhalten meiner Mail haben und nicht mit ihrer Auswertung Geld verdienen.
5. Der Mail-Provider sollte sich besonders für die Email-Sicherheit stark machen und optimaler Weise das Thema Sicherheit nicht erst seit Bekanntwerden des NSA-Skandals auf seine Fahnen geschrieben haben.

6. Der Mail-Provider sollte idealerweise ein deutscher Anbieter sein oder mindestens in Europa angesiedelt sein. Damit untersteht er auch europäischem bzw. deutschem Recht. So können die Behörden eines ausländischen Staates (typischerweise die US-Behörden) den Provider nicht zur Herausgabe von Informationen über den Kunden zwingen. Das deutsche Datenschutzrecht ist eines der besten weltweit. Noch sicherer ist es, wenn auch die Server des Anbieters in Europa oder besser in Deutschland stehen. So ist auch der physische Zugriff durch ausländische Geheimdienste auf die Server des Providers schwerer, wenn auch nicht unmöglich.
7. Der Mail-Provider sollte DANE unterstützen. (siehe Kap. 2.1.1)
8. Der Mail-Provider sollte einen guten Score auf <https://starttls.info> aufweisen.
9. Bestimmte Sicherheitsfeatures, die andere Anbieter nicht haben.
10. Spamschutz
11. Weitere Services wie die zentrale Verwaltung von Kontakten, Kalender, Aufgabenlisten und Dateien in der Cloud. Diese Dienste müssen ebenfalls gut gesichert sein.
12. Preis der Leistungen
13. Nachhaltigkeit, Nutzung von Ökostrom
14. Professioneller Webaufttritt

Die Kriterien 1 bis 6 halte ich für unabdingbar. Die weiteren Kriterien mag jeder anders gewichten. Sie können dazu dienen, unter den Providern, die die Kriterien 1 bis 6 erfüllen, denjenigen auszuwählen, der den eigenen Wünschen und Vorstellungen am nächsten kommt.

Das 7. Kriterium (DANE) ist wünschenswert. Da die DANE noch relativ neu ist, kann man die DANE-Unterstützung noch nicht von allen Mail-Providern erwarten. Die DANE-Pioniere in Deutschland sind *Posteo* und *mailbox.org*. Sie bieten die DANE-Unterstützung seit Mai 2014 an. Vermutlich wird sie im Laufe der kommenden Monate auch von weiteren Providern implementiert.

2.2.2 Meine Auswahl

Eine Aussage vorweg: Über Email-Provider aus dem nicht deutsch-sprachigen Raum kann ich keine generelle Aussage treffen. Viele vor allem kleinere Anbieter sind mir sicherlich nicht bekannt.

Nach dem Studium o.g. Artikels kamen die folgenden Anbieter in die engere Wahl. Sie erhielten im c't-Provider-Vergleich die besten Bewertungen und erfüllen die Kriterien 1 bis 6. Posteo und mailbox.org bieten DANE-Unterstützung und erfüllen damit auch das 7. Kriterium.

- **MyKolab**, Schweizer Anbieter, Website: <https://mykolab.com>
- **Posteo**, Deutscher Anbieter, Website: <https://posteo.de>
- **JPBerlin**, Deutscher Anbieter, Website: <https://www.jpberlin.de>
- **mailbox.org**, Deutscher Anbieter, Website: <https://mailbox.org>
- **mail.de**, Deutscher Anbieter, Website: <https://mail.de>

MyKolab bietet über Email hinaus die meisten zusätzlichen Leistungen. Er ist aktuell (April 2014) mit einem Preis von 7,59 €/Monat für das kleinste Leistungspaket mit Abstand auch der teuerste Anbieter. Alle anderen Anbieter bieten das jeweils kleinste Leistungspaket für 1,- €/Monat an. Dieses Paket ist für den Privatanwender in der Regel ausreichend.

Posteo, *JPBerlin* und *mailbox.org* bieten über das sichere Mail-Angebot hinaus in etwa die gleichen Leistungen zum selben Preis. Da fällt die Wahl schwer.

Hinter *JPBerlin* und hinter *mailbox.org* steht dasselbe Team, die *Heinlein Support GmbH*. Heinlein Support bietet sichere Mail seit 1992 an. *mailbox.org* ist im c't-Provider-Vergleich nicht enthalten. Da hinter beiden Anbietern dasselbe Team und dasselbe Know-how steckt, gehe ich auch von derselben technischen Qualität des Angebots aus.

mail.de ist ebenfalls ein sehr preisgünstiger Anbieter mit vier Produkten. Das kleinste ist ein kostenloses, jedoch nicht werbefreies Freemail-Angebot. Weitere werbefreie Angebote werden zwischen zwei und fünf Euro/Monat angeboten (Stand Juli 2014).

Die Website von *mailbox.org* ist noch deutlicher auf Sicherheit ausgerichtet als *JPBerlin*. Außerdem bietet *mailbox.org* zwei zusätzliche Sicherheitsfeatures, die bei den anderen Anbietern nicht zu finden sind:

- Ein zweiter Mail-Alias mit garantiert verschlüsseltem Übertragungskanal zwischen den Providern: Zusätzlich zur Haupt-Mail-Adresse mein.name@mailbox.org kann man einen weiteren Mail-Alias mein.name@secure.mailbox.org einrichten. Bei Verwendung dieses Alias stellt der Anbieter sicher, dass die Mail nur über einen verschlüsselten Kommunikationskanal vom oder zum Provider meines Kommunikationspartners übertragen wird (siehe Kap. 2.1, Schritt 4). Kann *mailbox.org* diese verschlüsselte Übertragung nicht sicherstellen, scheitert die Übertragung. (Beschreibung in Kapitel 2.8 und unter <https://mailbox.org/mails-definitiv-sicher-versenden/>)
- Verschlüsseltes Postfach: *mailbox.org* bietet außerdem an, Mails, die der Absender nicht verschlüsselt hat, sofort nach dem Eintreffen zu verschlüsseln und dann verschlüsselt in meinem Posteingang abzulegen. Ist die eingegangene Mail verschlüsselt gespeichert, kann auch das Team von *mailbox.org* oder ein Hacker, der in den Server von *mailbox.org* einbricht, die Mail nicht mehr entschlüsseln und lesen. Da nur ich selbst den passenden Schlüssel habe, kann nur ich selbst sie entschlüsseln und lesen. (Beschreibung in Kapitel 3.11 und unter <https://mailbox.org/ihr-e-mail-postfach/#Datenschutz>)

Meine persönliche Entscheidung ist schließlich für den Anbieter ***mailbox.org*** gefallen.

2.3 Ein paar Bemerkungen zu Gmail

Der Mail-Service von Google zählt sicherlich zu den komfortabelsten und professionellsten Angeboten. Meines Erachtens nimmt Google auch den Datenschutz sehr ernst und arbeitet bei der Mail-Übertragung wenn möglich auch mit starker Verschlüsselung. Allerdings gehört das Auswerten meiner Mails (Daten und Metadaten) zum Geschäftsmodell von Google. Deshalb ist Google nicht mehr mein bevorzugter Mail-Provider. Meinen Mailverkehr werde ich sukzessive von Google auf *mailbox.org* umstellen.

Ein anderes (unrealistisches) Szenario: Ich bleibe bei Google und verschlüssele alle meine Mails. Dann könnte Google die Inhalte meiner Mails nicht mehr lesen und auswerten. Dies ist praktisch nicht durchführbar. Dazu müssten alle meine Kommunikationspartner einen öffentlichen Schlüssel haben, mit dem ich die Mails an sie verschlüsseln könnte. Und alle meine Kommunikationspartner müssten die Mails an mich mit meinem öffentlichen Schlüssel verschlüsseln. Da sich Mail-Verschlüsselung zurzeit noch nicht auf breiter Basis durchgesetzt hat, sind wir von diesem Idealszenario sehr weit entfernt. Mehr zur Verschlüsselung mit öffentlichen und privaten Schlüsseln in Kapitel 3.3.

Google bietet außer dem Mail-Dienst noch viele weitere wichtige Dienste. Um diese zu nutzen, ist ein Gmail-Account die Voraussetzung. Z.B. kann man ein Android-Smartphone oder -Tablet kaum sinnvoll ohne einen Google-Mail-Account betreiben. Nur über die Anmeldung im Google Play Store erhält man die Updates für das Android-Gerät. Für die Anmeldung benötigt man eine Google-Mail-Adresse.

Auch wenn ich meinen Mailverkehr nicht mehr über Google abwickele, werde ich meinen Account trotzdem behalten, um für mich nützliche Google-Dienste weiter nutzen zu können.

Meinen privaten Terminkalender, der auf den Google Servern liegt kann ich ebenfalls zum neuen Provider meines Vertrauens *mailbox.org* umziehen, ebenso das Adressbuch und die Aufgabenliste.

2.4 Konfiguration des Mail-Clients

Um eine sichere Übertragung beim Mailversand und -empfang zu gewährleisten, muss der Provider – wie oben beschrieben – einen verschlüsselten Übertragungskanal anbieten. Im Mail-Programm auf meinem Rechner muss ich diesen aber auch nutzen. Heute (Mai 2014) habe ich meist keine Wahl mehr; die meisten deutschen Provider bieten nur noch die verschlüsselte Übertragung an; unverschlüsselte Übertragung funktioniert nicht mehr. Dasselbe gilt auch für die meisten amerikanischen Provider.

Ich nutze den verschlüsselten Kanal, indem ich verschlüsselte Übertragung (SSL/TLS oder STARTTLS) im Email-Programm (Email-Client) einstelle. Dies gilt grundsätzlich für alle Mail-Clients (*Thunderbird*, *Outlook*, *Pegasus Mail*, *Apple Mail* und viele weitere; es gilt auch für die Mail-App auf dem Smartphone). In diesem Dokument wird dies exemplarisch für den weit verbreiteten Mail-Client *Thunderbird* erläutert.

2.4.1 Thunderbird-Konfiguration

Mozilla *Thunderbird* ist das von Privatanwendern meist genutzte und auch mein bevorzugtes Mail-Programm. *Thunderbird* ist für Windows, Mac OS X und für alle Linux-Varianten verfügbar. Für Smartphones und Tablets unter iOS und Android ist *Thunderbird* nicht verfügbar. Mehr dazu im Kapitel 2.6. Ich beziehe mich hier nur auf *Thunderbird*.

Richtet man in *Thunderbird* einen neuen Mail-Account ein, gibt man die neue Email-Adresse an. *Thunderbird* kann aus der Email-Adresse allermeist auf den Provider schließen und das neue Konto passend zum neuen Provider automatisch richtig und sicher konfigurieren. Man kann dies aber auch nachträglich in den Konto-Einstellungen des betreffenden Mail-Accounts prüfen und ggf. auch ändern.

1. Prüfung für den Mail-Versand mit SMTP (Simple Mail Transfer Protocol): In den Konten-Einstellungen muss beim Postausgangsserver (SMTP) als Verbindungssicherheit entweder SSL/TLS oder besser STARTTLS eingetragen sein. (Abb.2)
2. Prüfung für den Mail-Versand mit SMTP (Simple Mail Transfer Protocol): In den Konten-Einstellungen muss beim Postausgangsserver (SMTP) als Verbindungssicherheit entweder SSL/TLS oder besser STARTTLS eingetragen sein. (Abb. 3)
3. Prüfung für den Mail-Empfang mit POP3 (Post Office Protocol, Version 3): In den Konten-Einstellungen muss beim Posteingangsserver (POP) als Verbindungssicherheit entweder SSL/TLS oder besser STARTTLS eingetragen sein. POP wird heute nur noch selten verwendet. Bei der Verwendung von POP werden die Mails auf den Rechner des Benutzers heruntergeladen und standardmäßig vom Server des Providers gelöscht. Die Mails können

nur noch auf einem Gerät gelesen und bearbeitet werden. (Die Einstellungen sind den IMAP-Einstellungen ähnlich und werden nicht in einer eigenen Abbildung gezeigt.)

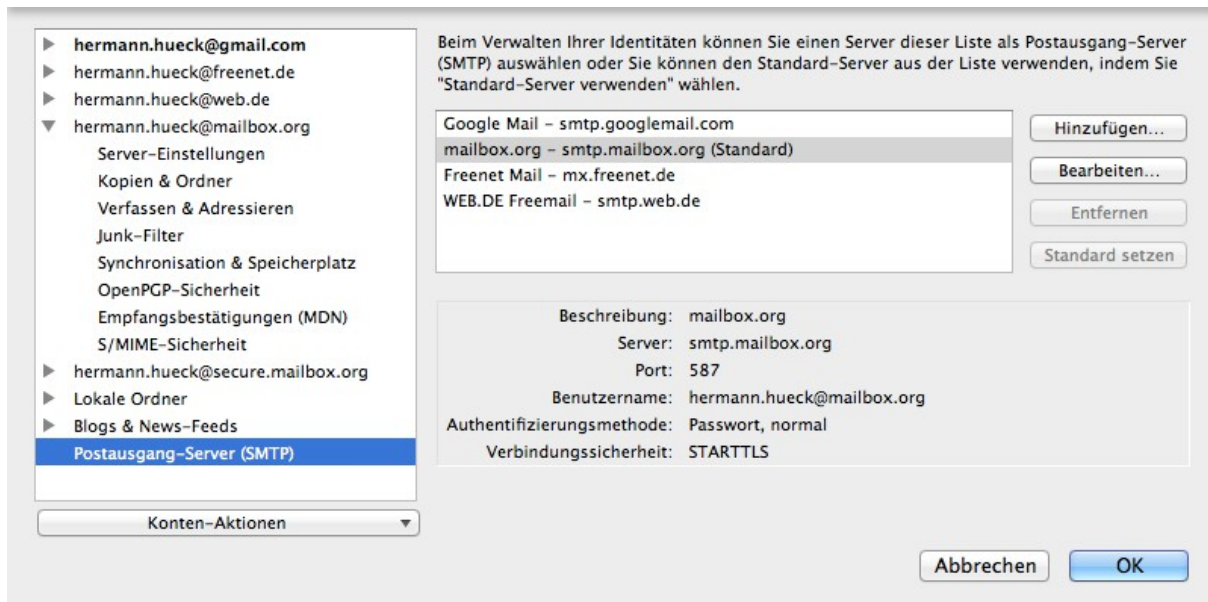


Abbildung 4: Thunderbird-Konfiguration für den Postausgang-Server (SMTP)

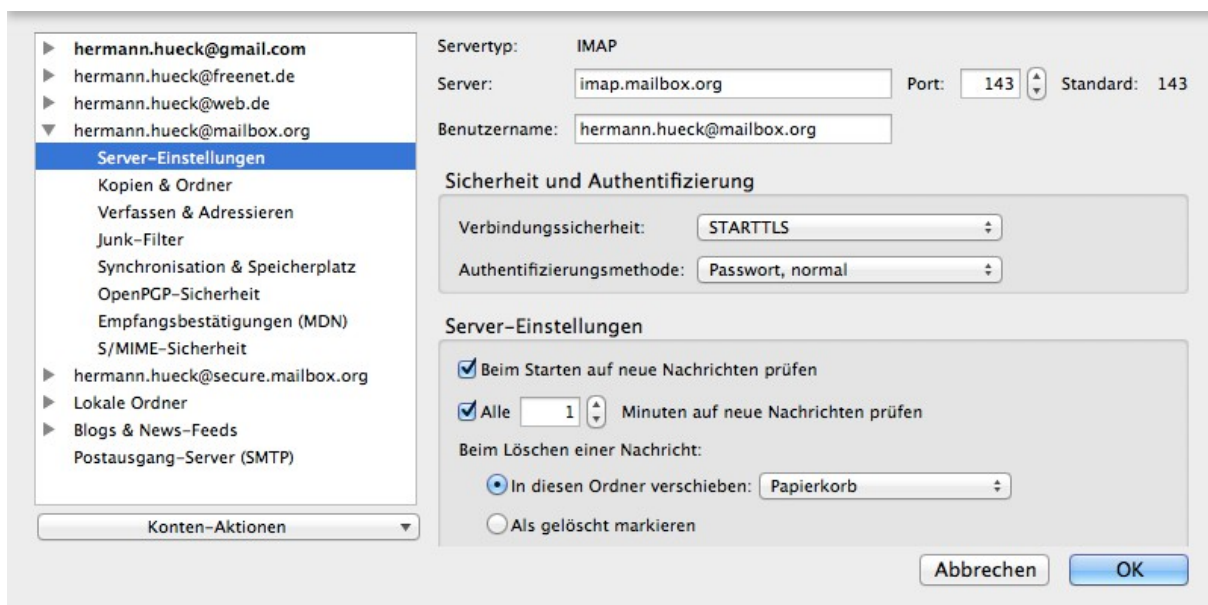


Abbildung 5: Thunderbird-Konfiguration für den Posteingang-Server (IMAP)

Bei allen drei Protokollen ist allermeist die Verbindungssicherheit *STARTTLS* erforderlich. Seltener ist die Einstellung *SSL/TLS* die Richtige.

Auf keinen Fall sollte für die Verbindungssicherheit die Option *keine* gewählt werden. Mit dieser Einstellung würden die Mails und auch das Anmeldepasswort im Klartext über das Netz übertragen und ist für jeden einsehbar. Diese Einstellung funktioniert allerdings in den meisten Fällen nicht

mehr, da die unverschlüsselte Übertragung zwischen Mail-Provider und dem Kunden in der Regel nicht mehr unterstützt wird.

Die Details der *Thunderbird*-Konfiguration will ich hier nicht wiederholen. Sie ist an vielen Stellen im Web zu finden, z.B. unter folgenden URLs:

- <https://support.mozilla.org/de/kb/konto-einrichten>
- [http://www.thunderbird-mail.de/wiki/Postausgang-Server_\(SMTP\)_einrichten](http://www.thunderbird-mail.de/wiki/Postausgang-Server_(SMTP)_einrichten)
- [http://www.thunderbird-mail.de/wiki/E-Mail-Konto_\(IMAP\)_einrichten](http://www.thunderbird-mail.de/wiki/E-Mail-Konto_(IMAP)_einrichten)
- http://www.thunderbird-mail.de/wiki/E-Mail-Konto_einrichten

2.4.2 Thunderbird-Updates

Thunderbird muss – wie alle anderen Programme – aktuell gehalten werden. Manuell erledigt man das unter *Hilfe* → *Auf Updates überprüfen*.

Es lässt sich auch eine Update-Automatik einstellen in *Thunderbird* unter *Einstellungen* → *Erweitert* → *Update*. Dort sollten beide Häkchen gesetzt werden, für das Update von *Thunderbird* und für die Updates der Add-ons.

Siehe auch folgende URL:

- <https://support.mozilla.org/de/kb/thunderbird-update-einstellungen>

2.5 Worauf man sonst noch achten muss

2.5.1 Rechnersicherheit

Gehen wir davon aus, dass wir einen tauglichen Mail-Provider gewählt haben und dass dieser einen guten Job macht. Die oben beschriebenen Maßnahmen stellen die Sicherheit nur zwischen dem Benutzer-Rechner und dem Server des Providers – also bei den Schritten 2 bis 6 des Übertragungswegs (siehe Kapitel 2.1) sicher.

Für die Sicherheit der Schritte 1 und 7 sind wir selbst als Absender und Empfänger zuständig. Wir müssen unseren Rechner aktuell und virenfrei halten. Da fast täglich neue Sicherheitslücken entdeckt und neue Viren entwickelt werden, müssen das Betriebssystem des Rechners, die installierten Programme und ggf. die Viren-Signaturen dauernd aktuell gehalten werden.

Den Rechner aktuell zu halten, dient nicht nur dazu, die Kompromittierung der Mails zu verhindern, sondern es ist eine allgemeine Sicherheitsmaßnahme. (Siehe auch Kap. 4.2)

2.5.2 Email-Tracking verhindern

Mails sind heute meist kein reiner Text, sondern sie sind im HTML-Format verfasst. HTML-Mails erlauben die Strukturierung des Textes, Formatierungen (verschiedene Schriftarten, Schriftgrößen) und auch das Einbetten von Bildern.

Sog. **Tracking-Images** sind kleine, unsichtbare, in eine HTML-Mail eingebettete Bilder. Sie sind in der Regel nur 1x1 Pixel groß und damit für den Mail-Empfänger quasi unsichtbar. Beim Öffnen der Mail werden die Bilder meist aus dem Web nachgeladen. Durch das Nachladen der Bilder beim Öffnen der Mail übermittelt der Mail-Client (oder der Browser bei der Nutzung von Webmail, siehe Kap. 2.7) einige Informationen über den Empfänger an den Server, auf dem die Bilder liegen und damit an den Mail-Absender. Der Mail-Versender kann auf diesem Wege erfahren,

- wann die Mail geöffnet wurde,
- wo die Mail gelesen wurde,
- an welchem Betriebssystem (Windows, OS X, iOS oder Android) die Mail gelesen wurde,
- mit welchem Mail-Client die Mail gelesen wurde (oder bei Webmail mit welchem Browser)
- und bei welchem Internet-Provider der Mail-Empfänger aktuell seinen Internetzugang hat.

Alles in Allem sind dies durchaus sensible Informationen. Interesse daran haben sowohl große Internetfirmen, die Profile über das Verhalten ihrer Benutzer erstellen, als auch Spam-Versender, die so den Erfolg ihres Spam-Versandes prüfen können.

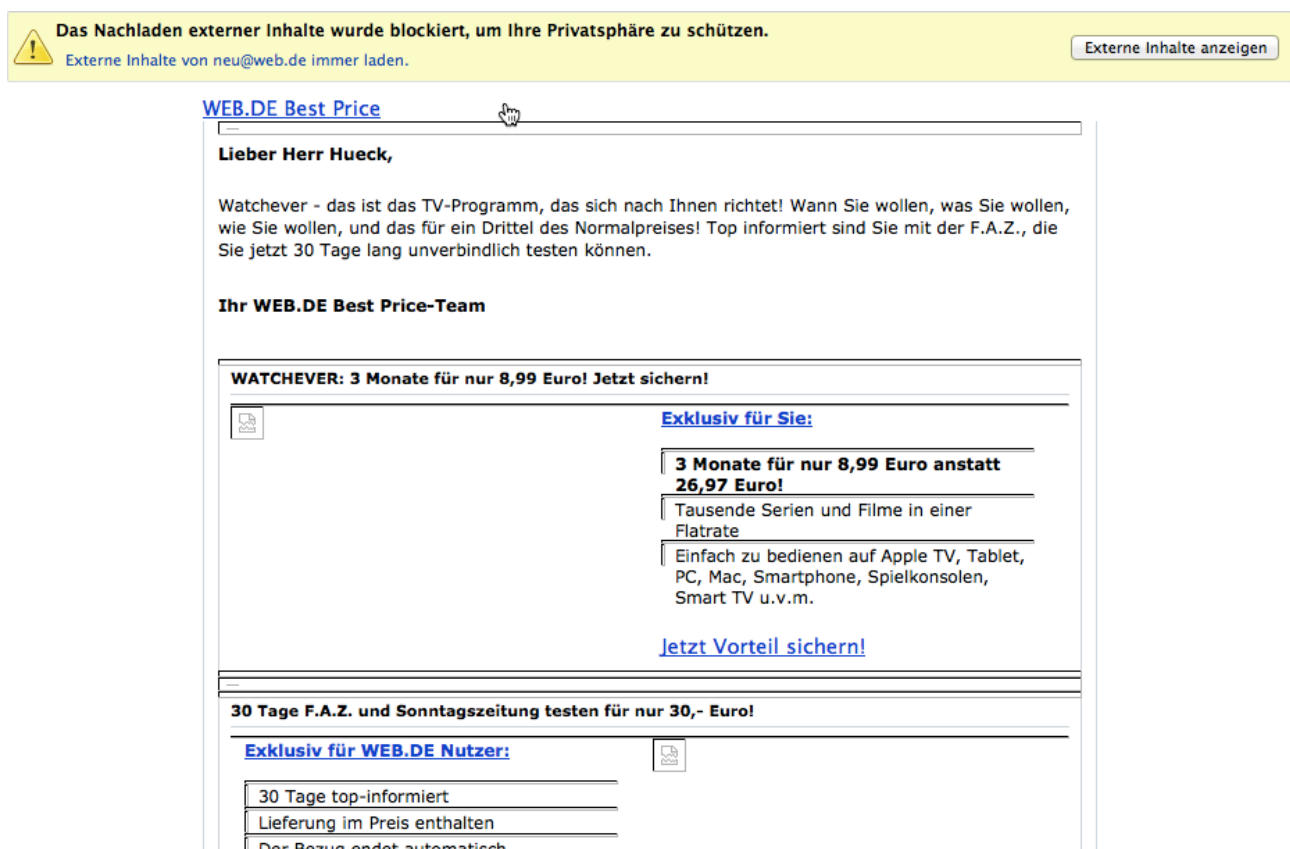


Abbildung 6: Thunderbird: Werbemail von WEB.DE vor dem Laden der externen Inhalte

Diese Gefahr lauert nicht nur in den Bildern, sondern auch in anderen externen Inhalten wie Audio- und Video-Dateien, aber auch CSS und JavaScript).

Um dies zu vermeiden, sollte der Mail-Client **externe Inhalte nicht automatisch aus dem Web nachladen**. Bei *Thunderbird* ist dies die Voreinstellung. Das Nachladen der externen Inhalte muss in *Thunderbird* vom Benutzer bei jeder Mail explizit mit einem Mausklick angefordert werden.

Wie sehr die externen Inhalte das Aussehen einer Mail bestimmen können, lässt sich schön an folgendem Beispiel einer Werbemail von WEB.DE veranschaulichen, die einmal vor dem Laden (Abb. 4) und einmal nach dem Laden (Abb. 5) der externen Inhalte gezeigt wird.

Dieses Beispiel zeigt auch, wie lästig Mails mit externen Inhalten sein können. Außerdem kann es

teuer sein, wenn man solche Mails auf dem Smartphone anzeigt und dabei das beschränkte monatliche Internet-Transfer-Kontingent verbraucht, das einem der Mobilfunkvertrag einräumt. Ob diese Mail auch ungewünschtes User-Tracking durchführt, sieht man ihr äußerlich nicht an. Man sollte jedoch davon ausgehen. Der Provider (hier WEB.DE) möchte doch gar zu gerne wissen, ob ich mir seine Werbung genauer angesehen und dazu die externen Inhalte nachgeladen habe.

Mehr Infos zu diesem Thema sind wieder im c't-Sonderheft „Sichere E-Mail“ im Artikel „Tracking aufspüren und abstellen“ zu finden. (Siehe Kap. 1.6)

WEB.DE Best Price
Exklusiv für WEB.DE Nutzer

Lieber Herr Hueck,

Watchever - das ist das TV-Programm, das sich nach Ihnen richtet! Wann Sie wollen, was Sie wollen, wie Sie wollen, und das für ein Drittel des Normalpreises! Top informiert sind Sie mit der F.A.Z., die Sie jetzt 30 Tage lang unverbindlich testen können.

Ihr WEB.DE Best Price-Team

WATCHEVER: 3 Monate für nur 8,99 Euro! Jetzt sichern!

Exklusiv für Sie:

- 3 Monate für nur 8,99 Euro anstatt 26,97 Euro!
- Tausende Serien und Filme in einer Flatrate
- Einfach zu bedienen auf Apple TV, Tablet, PC, Mac, Smartphone, Spielkonsolen, Smart TV u.v.m.

Nur bis 31.05.

Jetzt Vorteil sichern!

30 Tage F.A.Z. und Sonntagszeitung testen für nur 30,- Euro!

Exklusiv für WEB.DE Nutzer:

- 30 Tage top-informiert
- Lieferung im Preis enthalten
- Der Bezug endet automatisch

1 Geschenk für Sie

Abbildung 7: Thunderbird: Werbemail von WEB.DE nach dem Laden der externen Inhalte

2.6 Die passende Email-App auf dem Smartphone

Wer seine Mails definitiv nur auf einem Gerät – typischerweise dem PC – bearbeitet, der kann den Mail-Abruf im Mail-Client (*Thunderbird*, etc.) mit POP (Post Office Protocol) einrichten. Die Mails werden dabei lokal auf dem betreffenden Gerät gespeichert und vom Server gelöscht. Das ist heute eher unüblich geworden.

Bei IMAP (Internet Mail Access Protocol) bleiben die Mails zentral auf dem Server des Providers gespeichert. Der Zugriff ist von beliebig vielen Geräten mit einem Mail-Client sowie über die Webmail-Schnittstelle mit dem Browser (z.B. im Internet-Café) möglich.

Auf dem Rechner ist *Thunderbird* häufig der Mail-Client der Wahl. Er ist für die gängigen PC-Betriebssysteme (Windows, Mac OS X, Linux) verfügbar. Auf Tablets und Smartphones stehen unter Android und iOS diverse Mail-Apps zur Verfügung. Die Auswahl ist groß. Das IMAP-Protokoll wird von fast allen unterstützt. (Siehe c't-Sonderheft „Sichere Email“, Seite 50)

Will man seine Mails später auch noch mit PGP verschlüsseln, wird die Auswahl viel kleiner. Mit PGP-Unterstützung kommt unter Android gegenwärtig (2014) nur *K-9 Mail*, *K-@ Mail* oder *Kaiten* in Frage. Unter iOS steht nur *iPGMail* zur Verfügung. (Siehe Kap. 3.10.1)

Die Mail-App ist analog einzurichten wie *Thunderbird* (siehe Kap. 2.4.1). Als Zugriffsprotokolle sind SMTP für den Versand und IMAP für den Abruf der Mails zu konfigurieren. Wie bei *Thunderbird* ist für die Verbindungssicherheit meist STARTTLS, seltener SSL/TLS einzutragen.

2.7 Warnung vor der Nutzung von Webmail

Fast immer bieten die Provider auch den Webmail-Zugriff auf den Account an. Der Zugriff erfolgt mit einem beliebigen Browser auf jedem beliebigen Rechner mit Internetzugang.

Diese Option ist natürlich komfortabel, vor allem, wenn man auf seine Mails zugreifen möchte und nicht am eigenen Rechner sitzt und auch keinen Hosentaschenrechner (sprich: Smartphone) mit sich führt.

Die Web-Schnittstelle stellt allerdings auch eine zusätzliche Angriffsfläche auf den Mail-Account dar.

- Der fremde Rechner (z.B. im Internet-Café) steht in der Regel nicht unter der eigenen Kontrolle. Die Gefahr, dass er mit Malware verseucht ist, ist deutlich größer als am PC zu Hause. Dies erhöht die Gefahr des Passwortdiebstahls und der Kompromittierung des Mail-Accounts.
- Der Web-Server des Mail-Providers könnte gehackt und die Web-Site des Providers mit Malware verseucht worden sein. Dann ist der Webmail-Zugriff gefährlich, gleichgültig ob man den PC zu Hause oder den fremden Rechner im Internet-Café nutzt. Mit einem Mail-Client wie *Thunderbird* oder *Outlook* ist man dieser Gefahr deutlich weniger ausgesetzt.

Eine gute Grundregel: Webmail so wenig wie möglich nutzen. (Für Smartphone-Besitzer ist das auch kaum mehr erforderlich, da sie mit dem Smartphone fast an jedem Ort auf ihre Mails zugreifen können.)

Entscheidet man sich später auch für die Verschlüsselung und Signierung der Mails mit PGP, kommt Webmail prinzipbedingt nicht mehr in Frage. Denn der Browser hat keinen Zugriff auf die Schlüssel, auch dann nicht, wenn die Schlüssel auf dem System liegen, auf dem der Browser läuft. Man kann mit Webmail also keine verschlüsselten Mails versenden und auch keine verschlüsselten Mails lesen. (Siehe Kap. 3.7)

2.8 Sicherer Mail-Alias bei mailbox.org

Beim Provider *mailbox.org* kann jeder registrierte Benutzer (zusätzlich zur Standard-Mailadresse max.mayer@mailbox.org) einen weiteren Email-Alias

max.mayer@secure.mailbox.org

einrichten. Bei Verwendung dieser sicheren Mail-Adresse garantiert *mailbox.org*, dass die Übertragung der Mail vom oder zum Provider des Kommunikationspartners (Kap. 2.1, Schritt 4) verschlüsselt erfolgt. Das gilt sowohl für die Versandrichtung als auch für die Empfangsrichtung. Ist die verschlüsselte Übertragung nicht möglich, wird die Mail nicht übertragen und der Absender erhält eine Antwort mit einer Fehlermeldung. Die Mail kann also nicht von der NSA oder von anderen Internet-Bösewichtern gelesen werden.

Schickst du eine Mail an mich, ist die sichere Mail-Adresse hermann.hueck@secure.mailbox.org zu

bevorzugen. Sie funktioniert nach meiner bisherigen Erfahrung in den meisten Fällen. Es gibt nur sehr wenige Ausnahmen. Sollte die Mail-Übertragung mit der sicheren Adresse wider Erwarten scheitern, kannst du alternativ meine Standard-Mail-Adresse hermann.hueck@mailbox.org verwenden. Sie funktioniert immer.

Bei der Verwendung der Standard-Mail-Adresse versuchen Absender-Provider und Empfänger-Provider ebenfalls, einen verschlüsselten Kanal aufzubauen. Sollte die verschlüsselte Übertragung scheitern, da dein Provider dies nicht unterstützt, wird die Mail dennoch übertragen – allerdings unverschlüsselt.

Siehe auch: <https://mailbox.org/mails-definitiv-sicher-versenden/>

2.9 Links zu diesem Kapitel

- Qualität der SSL/TLS-Verschlüsselung von Email-Providern prüfen:
<https://starttls.info>
- *MyKolab*, Schweizer Email-Provider:
<https://mykolab.com>
- *Posteo*, Deutscher Email-Provider:
<https://posteo.de>
- *JPBerlin*, Deutscher Email-Provider:
<https://www.jpberlin.de>
- *mailbox.org*, Deutscher Email-Provider:
<https://mailbox.org>
- *mail.de*, Deutscher Anbieter, Website:
<https://mail.de>
- Verschlüsselte Mail-Übertragung von Provider zu Provider bei *mailbox.org*:
<https://mailbox.org/mails-definitiv-sicher-versenden/>
- Verschlüsseltes Postfach bei *mailbox.org*:
<https://mailbox.org/ihr-e-mail-postfach/#Datenschutz>
- *Thunderbird*-Konfiguration:
<https://support.mozilla.org/de/kb/konto-einrichten>
[http://www.thunderbird-mail.de/wiki/Postausgang-Server_\(SMTP\)_einrichten](http://www.thunderbird-mail.de/wiki/Postausgang-Server_(SMTP)_einrichten)
[http://www.thunderbird-mail.de/wiki/E-Mail-Konto_\(IMAP\)_einrichten](http://www.thunderbird-mail.de/wiki/E-Mail-Konto_(IMAP)_einrichten)
http://www.thunderbird-mail.de/wiki/E-Mail-Konto_einrichten
- *Thunderbird*-Updates:
<https://support.mozilla.org/de/kb/thunderbird-update-einstellungen>

3 Mails verschlüsseln und signieren

Ohne die Mail zu verschlüsseln, brauchen Absender und Empfänger vertrauenswürdige Provider, die Mails sicher (über verschlüsselte Übertragungskanäle) transportieren, die die Mail-Inhalte selbst nicht auslesen und sie auch an keine anderen Personen oder Instanzen weitergeben.

Wenn wir die Mail-Inhalte verschlüsseln, können unsere Provider (und auch sonst niemand) die Mails nicht mehr lesen. (Z.B. kann Google die Inhalte nicht mehr auswerten.) Sie bekommen nur noch unleserlichen Datensalat zu sehen.

Auch bei verschlüsselten Mails erfahren die Provider immer noch die Metadaten der Mails (Absender-Adresse, Empfänger-Adresse, Betreff, CC, Zeit, etc. – anders ausgedrückt: wer kommuniziert wann mit wem und worum geht's). Ohne die Metadaten könnten sie die Mails nicht zustellen. Auch bei verschlüsselten Mails brauchen wir vertrauenswürdige Provider, die ihren Job verstehen. Wir wollen nicht, dass die Metadaten in die falschen Hände geraten, denn auch aus den Metadaten lassen sich sehr persönliche Profile der Kommunizierenden erstellen.

Das Verschlüsseln der Mail-Inhalte liegt nur in unserer eigenen Verantwortung, in der des Absenders und des Empfängers. Auch kann ein Kommunikationspartner alleine die Verschlüsselung nicht realisieren. Es müssen immer beide Kommunikationspartner zusammenspielen.

Man spricht hier von der sog. **Ende-zu-Ende-Verschlüsselung: Auf dem gesamten Transportweg vom Absender bis zum Empfänger ist die Nachricht chiffriert.** So kann nur der Empfänger den Inhalt der Nachricht dechiffrieren und lesen. (Übrigens bieten die neuen Dienste *de-Mail* und *ePost*, die sich zurzeit als besonders sichere Email-Alternativen auf dem deutschen Markt zu präsentieren versuchen, keine Ende-zu-Ende-Verschlüsselung.)

Das vorliegende Kapitel versucht Email-Verschlüsselung mit **PGP (Pretty Good Privacy)** so einfach wie möglich zu erläutern, sodass auch der IT-Lai sie verstehen und letztlich auch umsetzen kann. Ich will dich als Mail-Partner gewinnen, mit dem ich verschlüsselte Mails austauschen kann.

Ich habe allerdings nicht den Anspruch, alle technischen Details umfassend zu erläutern. Zum Einen möchte ich in diesem Kapitel die wichtigsten Konzepte der Verschlüsselung mit PGP erläutern. Zum Anderen möchte ich den Interessierten die praktischen Anleitungen mitgeben, damit sie erst mal loslegen können. Ich liefere auch viele Web-Links mit, die es den Lesern ermöglichen, weitergehende Informationen an den betreffenden Web-Adressen abzurufen.

Die Email-Verschlüsselung (insbesondere die Schlüsselverwaltung) bleibt ein komplexes Thema. Doch wenn wir damit unsere Mails wieder unter unsere Kontrolle bringen und die Nachrichten vor der NSA und GCHQ, vor Google und Co. und vor neugierigen Internet-Kriminellen verbergen können, dann könnte es die Mühe wert sein.

Verschlüsselung zu verstehen und einzurichten, ist eine harte Nuss, aber es lohnt sich, sie zu knacken. Ich versuche in diesem Kapitel, euch dafür den Nussknacker zu liefern.

3.1 Asymmetrische Verschlüsselung

Einige wichtige Grundbegriffe erleichtern das Verständnis der nachfolgenden Kapitel.

Bei der **symmetrischen Verschlüsselung** wird derselbe digitale Schlüssel zur Verschlüsselung und zur Entschlüsselung verwendet. Werden für Verschlüsselung und Entschlüsselung zwei unterschiedliche (aber zusammengehörende) digitale Schlüssel verwendet, so spricht man von **asymmetrischer Verschlüsselung**. Gleichgültig, welches Verschlüsselungsverfahren man

verwendet, kommt bei der Email-Verschlüsselung immer die asymmetrische Verschlüsselung zur Anwendung.

Zur asymmetrischen Verschlüsselung von Nachrichten benötigen beide Kommunikationspartner ein Schlüsselpaar. Jedes Schlüsselpaar besteht aus einem privaten Schlüssel (**Private Key**), der niemals an andere weitergegeben werden darf und aus einem öffentlichen Schlüssel (**Public Key**), den man möglichst an alle Kommunikationspartner weiterreicht.

Will ich dir eine verschlüsselte Mail schicken, muss ich den Mail-Inhalt mit deinem öffentlichen Schlüssel verschlüsseln und du entschlüsselst die Mail mit deinem dazugehörigen privaten Schlüssel. Nur du kannst sie entschlüsseln, solange du der Einzige bist, der im Besitz dieses privaten Schlüssels ist. Umgekehrt verschlüsselst du die Mail an mich mit meinem öffentlichen Schlüssel und ich entschlüssele sie mit meinem privaten Schlüssel.

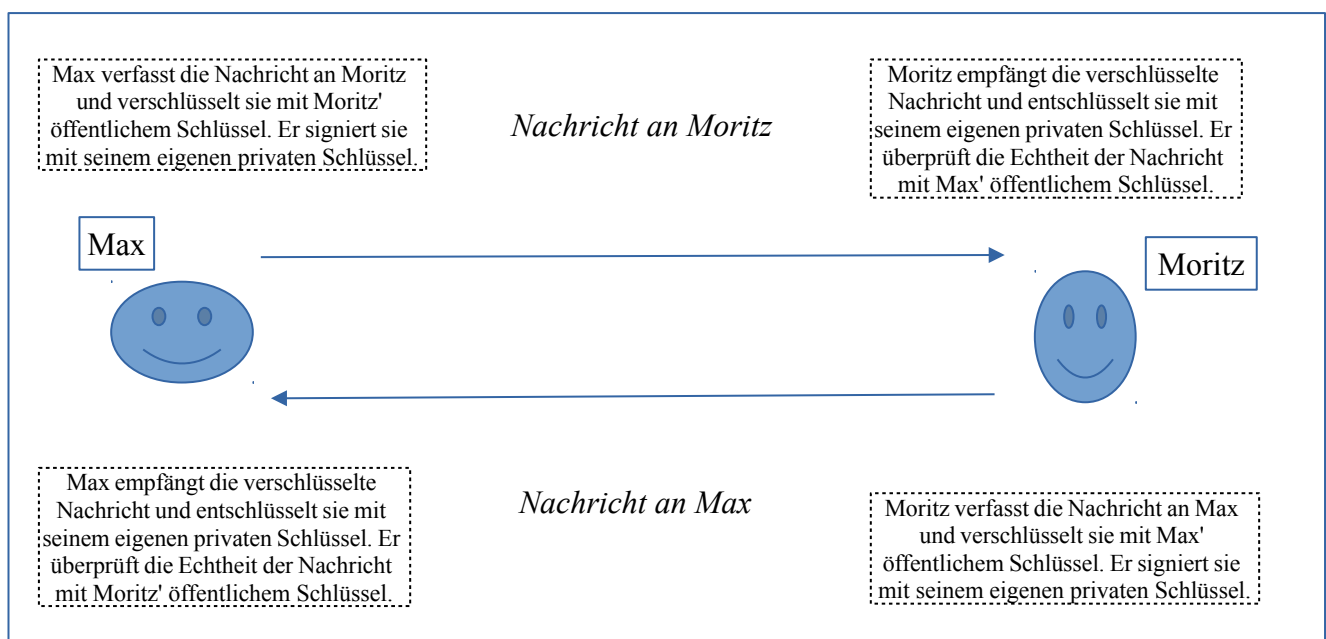


Abbildung 8: Asymmetrisch verschlüsselte Kommunikation

Können wir verschlüsseln, dann können wir das digitale Schlüsselpaar auch für das Signieren der Mails verwenden.

Wenn ich die Mail an dich mit meinem privaten Schlüssel signiere, kannst du die Echtheit der Signatur mit meinem öffentlichen Schlüssel überprüfen. Umgekehrt geht's natürlich genau so.

Damit Verschlüsselung und Signierung richtig funktionieren, muss man sicherstellen, dass der öffentliche Schlüssel des Empfängers, den der Absender zum Verschlüsseln verwendet, auch wirklich diesem Empfänger gehört. Eine eindeutige **Zuordnung des Schlüssels zu einer Benutzer-ID** (bestehend aus Name und Email-Adresse) muss garantiert sein. Dazu muss man den öffentlichen Schlüssel auf irgend eine Art beglaubigen (siehe Kap. 3.4.9).

3.2 Welches Verschlüsselungsverfahren – S/MIME oder PGP?

Zwei digitale asymmetrische Verschlüsselungsverfahren stehen zur Verfügung:

- **S/MIME** (Secure / Multipurpose Internet Mail Extensions); siehe auch

<http://de.wikipedia.org/wiki/S/MIME>

- **PGP** (Pretty Good Privacy):

<http://de.wikipedia.org/wiki/OpenPGP>

Die gängige Implementierung ist OpenPGP; siehe auch:

http://de.wikipedia.org/wiki/Pretty_Good_Privacy

Beide Verfahren sind asymmetrische Verschlüsselungsverfahren (siehe Kap. 3.3). Bei beiden Verfahren erzeugt der Benutzer ein Schlüsselpaar aus privatem und öffentlichem Schlüssel. Das Schlüsselpaar kann zum Verschlüsseln/Entschlüsseln von Dateien und Mails, als auch zum Signieren von Mails verwendet werden. Der private Schlüssel muss beim Benutzer verbleiben. Damit kein Missbrauch möglich ist, muss er sicher verwahrt werden und darf nicht in fremde Hände geraten. Der öffentliche Schlüssel wird möglichst vielen anderen Benutzern zugänglich gemacht.

Damit ein öffentlicher Schlüssel sinnvoll verwendet werden kann, muss sichergestellt sein, wem der betreffende Schlüssel gehört. Dazu muss er beglaubigt werden. Ein beglaubigter Schlüssel gilt dann als vertrauenswürdig. Durch die Beglaubigung kann man sicher sein, dass ein bestimmter Schlüssel einem bestimmten Benutzer gehört. Eine mit dem Schlüssel unterschriebene Mail kann so dem Benutzer mit Sicherheit zugeordnet werden (vorausgesetzt, man vertraut der Instanz, die die Beglaubigung vorgenommen hat).

Die Konzepte von S/MIME und PGP unterscheiden sich wesentlich durch das Beglaubigungsverfahren. Bei S/MIME basiert die Beglaubigung auf einer begrenzten Anzahl von Zertifizierungsstellen, denen man dann vertrauen muss. Bei PGP können die Benutzer ihre Schlüssel gegenseitig beglaubigen.

Bei **S/MIME** gibt es weltweit etwas mehr als 200 Zertifizierungsstellen. Diese nennt man CAs (Certificate Authorities). Bei solch einer CA kann man seinen öffentlichen Schlüssel beglaubigen/zertifizieren lassen. Nach Abschluss der Zertifizierung bekommt man den beglaubigten öffentlichen Schlüssel zurück, um ihn dann zu verwenden. Man kann sich das so vorstellen, als würde man zum Notar gehen und sich eine notarielle Beglaubigung für seinen Schlüssel holen. Diesem beglaubigten öffentlichen Schlüssel vertrauen die Benutzer, da die Browser, die Betriebssysteme und auch Mail-Clients wie *Thunderbird* schon bei der Installation eine Liste von CAs mitbringen. Sie können die Echtheit eines öffentlichen Schlüssels feststellen, indem sie prüfen, ob er von einer CA aus der CA-Liste beglaubigt wurde.

Das ganze S/MIME-System steht und fällt allerdings mit der Glaubwürdigkeit der CAs.

Das Zertifizierungssystem der CAs kommt nicht nur beim Verschlüsseln von Mails mit S/MIME zum Einsatz, sondern auch beim verschlüsselten Surfen im Internet mit dem HTTPS-Protokoll. Zwei Szenarien sollen die Schwachstellen, dieses System deutlich machen.

Erstes Szenario: Man stelle sich vor, die NSA zwingt Verisign (eine der bekanntesten CAs) ein Zertifikat für *google.com* auszustellen. Damit könnte die NSA z.B. einen falschen Web-Server *falscher-google.com* mit einem beglaubigten Zertifikat von *google.com* betreiben. Nehmen wir an, ich besuche den Server (<https://falscher-google.com>) mit meinem Browser mit HTTPS (Bei HTTPS wird die Übertragung zwischen Web-Server und Browser verschlüsselt.). Mein Browser prüft das Zertifikat, das der Server *falscher-google.com* schickt und vertraut diesem, da es von Verisign beglaubigt wurde und Verisign sich in der Liste vertrauenswürdiger CAs meines Browsers befindet.

Zweites Szenario: Hacker brechen bei einer CA ein und können ihre eigenen Zertifikate im Namen der gehackten CA ausstellen. Sie damit ebenfalls einen falschen Google-Server mit vermeintlich echten Zertifikaten betreiben. Dies ist vor ca. 2 Jahren bei der holländischen CA

Diginotar geschehen. (Diginotar hatte die Sicherheitsvorkehrungen auf seinen Servern sträflich vernachlässigt.) Als der Fall bekannt wurde, mussten die Hersteller der Browser schnell Updates mit aktualisierten CA-Listen liefern. Diginotar war danach nicht mehr in der Liste der vertrauenswürdigen CAs enthalten. Somit wurden die von Diginotar beglaubigten Zertifikate ungültig. Danach warnt mich der aktualisierte Browser, wenn ich eine Web-Site mit einem von Diginotar beglaubigten Zertifikat per HTTPS besuche. Ein Warnungshinweis im Browser teilt mir mit, dass das Zertifikat der Web-Site, die ich besuchen will, nicht überprüft werden kann. Nun kann ich die Warnung beachten und auf den Besuch der fraglichen Web-Site verzichten oder die Warnung ignorieren und die Site dennoch aufrufen.

So wie der Browser mit falschen Zertifikaten „hinters Licht geführt“ werden kann, genau so wäre auch eine auf S/MIME basierende Mail-Verschlüsselung korrumpierbar.

Abgesehen von dieser kurzen Übersicht gehe ich in diesem Dokument nicht weiter auf S/MIME ein.

Bei **PGP** gibt es keine zentralen Beglaubigungsstellen. Benutzer, die sich kennen, können ihre Schlüssel wechselseitig signieren und damit beglaubigen.

Benutzer, die sich kennen, können wechselseitig ihre öffentlichen Schlüssel signieren. Kennen sie sich nicht, müssen sie sich vor der Schlüsselsignierung gegenseitig ausweisen, z.B. mit dem Personalausweis oder einem anderen Ausweis-Dokument (Führerschein, Versicherungsausweis, etc.). Siehe Kap. 3.4.9)

3.3 Private Key und Public Key – Wie funktioniert's?

Empfehlung: Auf der Website von *mailbox.org* wird die asymmetrische Verschlüsselung und Signierung mit PGP in einem kurzen Stift-Film gut illustriert: <https://mailbox.org/stiftfilm-wie-funktioniert-e-mail-verschluesselung-mit-pgp/>. (Witzig und lehrreich! Am besten vor und nach dem Lesen dieses Kapitels ansehen! Und danach dieses Kapitel vielleicht noch ein zweites Mal lesen!)

Asymmetrische Verschlüsselung basiert auf einem digitalen Schlüsselpaar bestehend aus dem **Private Key** (dem privaten und geheimen Schlüssel, der niemals in den Besitz einer anderen Person gelangen darf) und dem **Public Key** (dem öffentlichen Schlüssel, den man an möglichst viele andere Personen verbreitet). Jeder Benutzer, der verschlüsselte und/oder signierte Nachrichten versenden und empfangen will, benötigt ein solches Schlüsselpaar. (Wie man ein Schlüsselpaar erzeugt, dazu später mehr, in Kap. 3.4.2.)

Was mit dem öffentlichen Schlüssel verschlüsselt wird (eine Datei oder eine Nachricht), wird mit dem privaten Schlüssel entschlüsselt. Und vice versa: was mit dem privaten Schlüssel verschlüsselt wird, kann nur mit dem öffentlichen Schlüssel entschlüsselt werden. Man kann sich das etwa vorstellen, als gäbe es ein Schloss, das mit dem einen Schlüssel verschlossen wird und nur mit dem passenden Partnerschlüssel wieder geöffnet werden kann.

Jeder Benutzer veröffentlicht seinen öffentlichen Schlüssel; d.h. er gibt ihn allen anderen Benutzern oder macht ihn sehr leicht zugänglich. Seinen privaten Schlüssel sichert er so gut es geht, am besten in einem **Schlüsselbund**, so dass nur er selbst ihn benutzen kann.

Verschlüsselung einer Nachricht: Willst du (oder irgend ein anderer Benutzer) mir eine Nachricht schicken, dann verschlüsselst du die Nachricht mit meinem öffentlichen Schlüssel. Nur ich kann sie lesen, da nur ich den privaten und geheimen Schlüssel dazu habe. Jeder andere, der die Nachricht unterwegs abfängt, kann zwar sehen, dass sie von dir stammt und dass sie an mich gerichtet ist und er kann auch ihren Betreff und andere Metadaten lesen. Die Nachricht selbst kann er aber nicht

lesen, da er den privaten Schlüssel nicht hat. Schicke ich dir eine Nachricht, verschlüssele ich sie mit deinem öffentlichen Schlüssel. Tatsächlich ist das Verfahren etwas komplizierter, aber diese etwas vereinfachte Vorstellung genügt, um damit zu arbeiten.

Signatur einer Nachricht: Mit der Signatur einer Nachricht kann ihre Echtheit oder Unverfälschtheit garantiert werden. D.h. es kann sichergestellt werden, dass die empfangene Nachricht exakt der gesendeten entspricht und dass nichts, nicht einmal ein Komma hinzugefügt, gelöscht oder verändert wurde. Außerdem ist garantiert, dass der Besitzer des öffentlichen Schlüssels der Absender ist. (Dabei wird vorausgesetzt, dass der Besitzer seinen privaten Schlüssel nicht verloren hat und dass er nicht entwendet wurde.)

Kommt der private (und geheime) Schlüssel (durch Verlust oder Diebstahl) in die falschen Hände, könnte der Schlüsseldieb sich als der Schlüsselbesitzer ausgeben und im Namen des Besitzers Nachrichten verschlüsseln und/oder signieren.

Eine Nachricht wird mit dem privaten und geheimen Schlüssel vom Absender signiert. Mit dem öffentlichen Schlüssel des Absenders kann der Empfänger die Unverfälschtheit der Nachricht überprüfen.

Wer's technisch genau wissen will: Aus dem Nachrichtentext wird ein Hash-Wert (eine Art eindeutige Quersumme) gebildet und der Hash-Wert wird mit dem privaten Schlüssel des Absenders verschlüsselt und an die Nachricht angehängt. (Das bedeutet Signieren.) Der Empfänger kann den Hash-Wert mit dem öffentlichen Schlüssel des Absenders entschlüsseln und er kann aus der empfangenen Nachricht ebenfalls den Hash-Wert errechnen. Sind beide Hash-Werte - der entschlüsselte und der errechnete - identisch, dann ist sichergestellt, dass sie während des Transportes nicht verfälscht wurde und dass der bekannte Besitzer des öffentlichen Schlüssels der Absender ist.

Verschlüsselung und Signatur können kombiniert oder völlig unabhängig voneinander verwendet werden. D.h. eine verschlüsselte Nachricht muss nicht signiert sein und eine signierte Nachricht muss nicht verschlüsselt sein.

Die **Beglaubigung der Schlüssel** ist erforderlich, damit sie sinnvoll zum Signieren und Verschlüsseln von Nachrichten verwendet werden können. Nur den beglaubigten Schlüsseln kann man wirklich vertrauen. Anders ausgedrückt: Die wechselseitige Beglaubigung von Schlüsseln erzeugt eine **Vertrauensbeziehung** zwischen zwei Partnern.

Zur Beglaubigung eines fremden, öffentlichen Schlüssels verwendet man wiederum den eigenen Schlüssel. Einen fremden Schlüssel zu beglaubigen bedeutet technisch gesehen nichts anderes als ihn mit dem eigenen Schlüssel zu unterschreiben. Mehr dazu in Kap. 3.4.9

Beglaubigen Max und Moritz ihre öffentlichen PGP-Schlüssel, vertrauen sie sich gegenseitig und können den jeweils anderen immer an seinem Schlüssel „erkennen“. Schickt Max an Moritz eine mit dem beglaubigten Schlüssel signierte Mail, so kann Moritz sicher sein, dass sie von Max ist und nicht von jemand anders, der sich als Max ausgibt. Schickt Max an Moritz eine verschlüsselte Mail, ist sie mit dem beglaubigten, öffentlichen Schlüssel von Moritz verschlüsselt. Max kann sicher sein, dass nur Moritz sie lesen kann, solange nur Moritz im Besitz des privaten und geheimen Schlüssel ist. Durch die wechselseitige Beglaubigung ihrer öffentlichen Schlüssel drücken die beiden ihr **direktes Vertrauen** aus.

Moritz hat auch den öffentlichen Schlüssel von Witwe Bolte beglaubigt. Max hat Moritz schon sein Vertrauen ausgedrückt, indem er dessen Schlüssel beglaubigt hat. Er vertraut nun auch, dass Moritz die Beglaubigung von Witwe Bolte korrekt durchgeführt hat. (Entweder kennt er Witwe Bolte persönlich oder er hat sich ihren Ausweis zeigen lassen.) Also kann er jetzt Witwe Bolte vertrauen, da er Moritz (Witwe Boltes Beglaubigender) vertraut. In diesem Fall spricht man von **transitivem Vertrauen**.

Durch viele direkte und transitive Vertrauensverhältnisse entsteht ein Netz des Vertrauens, ein sogenanntes **WoT** oder **Web of Trust**. Man benötigt also keine besonderen Schlüssel-Zertifizierungsstellen, die CAs oder Certificate Authorities, wie das bei S/MIME der Fall ist.

Hört man das alles zum ersten Mal, mag es recht kompliziert klingen. Hat man die Konzepte der asymmetrischen Verschlüsselung und das Prinzip wechselseitiger Schlüsselbeglaubigung einmal richtig verstanden, ist es gar nicht mehr so schwierig.

Dies sind die Grundlagen zum Verständnis. In den nächsten Kapiteln sehen wir uns an, welche Tools man auf dem eigenen System installieren muss und wie man diese Tools nutzt, um ein Schlüsselpaar zu erzeugen, öffentliche Schlüssel zu beglaubigen und zu veröffentlichen und um schließlich signierte und verschlüsselte Mails zu versenden.

Genauer und technisch detaillierter ist dies alles beschrieben im c't-Sonderheft „Sichere Email“ (siehe Kap. 1.6) auf Seite 82 im Artikel „Verschlüsseln und Signieren mit PGP“. Meine Ausführungen reduziere ich auf das Wichtigste, auf das was man braucht, um möglichst schnell zum Ziel zu kommen und verschlüsselte und signierte Mails versenden und empfangen zu können.

Eine kurze Einführung in PGP für Laien (inklusive *Thunderbird*-Konfiguration) ist auch bei *WEB.DE* zu finden: <https://hilfe.web.de/sicherheit/pgp.html>.

3.4 Verwaltung des Schlüsselbunds

Einen Schlüsselbund benötigt man, um mehrere Schlüssel zu bündeln und gemeinsam zu verwalten. **Ein Schlüsselbund enthält typischerweise einen eigenen (privaten und öffentlichen) Schlüssel und die öffentlichen Schlüssel der Kommunikationspartner.** Die Schlüsselverwaltung (oder Schlüsselbund-Verwaltung) kann dem Schlüsselbund ...

- neue Schlüssel hinzufügen oder bestehende löschen
- Schlüsseleigenschaften ändern (z.B. das Verfallsdatum oder das Besitzervertrauen)
- Schlüssel in Dateien exportieren und aus Dateien importieren
- fremde Schlüssel mit dem eigenen signieren / beglaubigen
- Schlüssel widerrufen, um sie ungültig zu machen
- Widerrufszertifikate (für den späteren Widerruf) erstellen
- Schlüssel auf Key-Server hochladen oder von diesen herunterladen

Die Begriffe „Schlüsselbund“, „Schlüsselverwaltung“ und „Schlüsselbund-Verwaltung“ werde ich in den folgenden synonym verwenden und bezeichne damit ein Software-Tool, das die Schlüssel verwaltet, d.h. sie darstellt und bearbeitet, löscht oder neue hinzufügt.

Am besten, man probiert's parallel zum Lesen der folgenden Kapitel gleich aus.

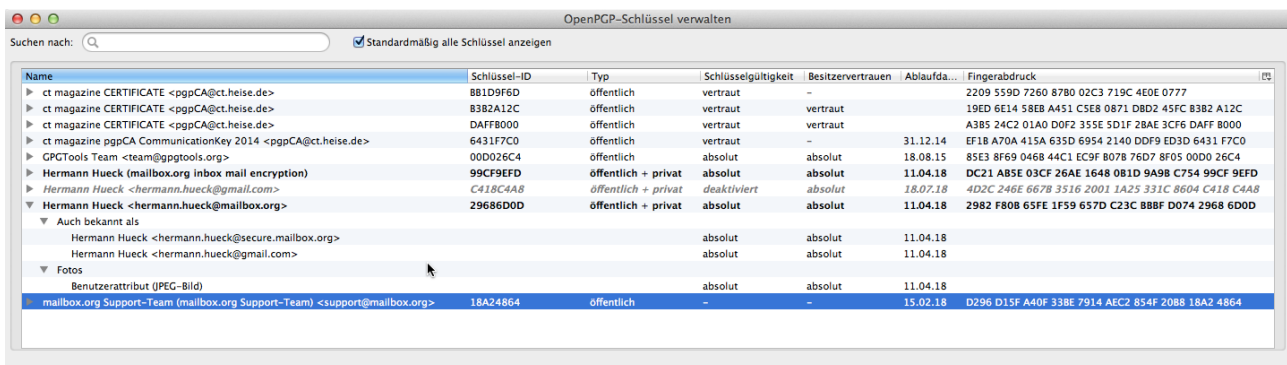


Abbildung 9: Thunderbird: Der Enigmail-Schlüsselbund mit einigen Schlüsseln

3.4.1 Tools zur Schlüsselverwaltung

Wir benötigen folgende Tools auf unserem Rechner:

- Ein Schlüsselverwaltungstool:
 - **Gpg4win** unter Windows; Download unter <http://www.gpg4win.de/> oder unter <http://www.heise.de/download/gpg4win-e2d914fd06f82399ae36052e8362b95c-1398246471-2619466.html>
 - **GPG Keychain Access** von GPGTools unter Mac OS X. (Der Hersteller propagiert heute die *GPG Suite* und bietet dieses Paket nicht mehr auf seiner Website an.) Download von *GPG Keychain Access* bei Heise unter <http://www.heise.de/download/gpg-keychain-access-1178953.html>

Vom Hersteller *GPGTools* gibt es auch die **GPG Suite**. Diese enthält alle Funktionen von *GPG Keychain Access* und weitere Funktionen (z.B. die Unterstützung für Apple Mail). Die *GPG Suite* kann auf dem Mac ebenso gut verwendet werden. Download unter <https://gpgtools.org/> oder bei Heise unter <http://www.heise.de/download/gpg-suite-a8929973af027b9846bd8eeb330da2d4-1398337947-2678714.html>.

Ich beziehe mich in diesem Dokument nur auf *GPG Keychain Access*; die *GPG Suite* ist genauso zu verwenden.

- **GnuPG** unter Linux (ist in den diversen Linux-Distributionen meist enthalten) erlaubt die Schlüsselverwaltung auf der Kommandozeile. Für Ubuntu Linux ist auch das graphische Schlüsselverwaltungstool **Seahorse** verfügbar. Dieses arbeitet analog zu *GPG4win* und *GPG Keychain Access* und wird hier nicht weiter beschrieben.
- **Thunderbird** als Mail-Client. (Ich beziehe mich wieder nur auf *Thunderbird*, da das Programm im Privat-Bereich sehr weit verbreitet ist und da ich es selbst nutze.) Andere Mail-Clients können auch verwendet werden, werden in diesem Dokument aber nicht

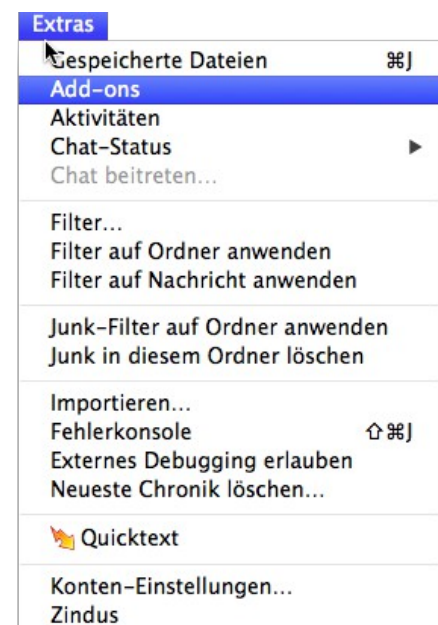


Abbildung 10: Thunderbird: Verwaltung der Add-ons öffnen

erläutert.

- Das *Thunderbird*-Add-On **Enigmail**. *Enigmail* enthält ebenfalls eine Schlüsselverwaltung und stellt außerdem die kryptographischen Funktionen bereit, die für den Versand und Empfang signierter und verschlüsselter Mails mit *Thunderbird* erforderlich sind. *Enigmail* integriert die PGP-Verschlüsselungsfunktionen in *Thunderbird* und macht diese so innerhalb des Mail-Programms komfortabel nutzbar. *Enigmail* kann direkt mit dem *Thunderbird*-Add-On-Manager (unter *Extras* → *Add-ons*) gesucht und installiert werden. Zweckmäßigerweise hat man das native, plattform-spezifische Schlüsselverwaltungstool (*Gpg4win* bzw. *GPG Keychain Access*) schon vorher installiert, sodass *Enigmail* dieses beim ersten Start gleich vorfindet. Eine *Enigmail*-Installationshilfe ist hier zu finden:

- http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP#Enigmail_installieren
- <https://www.verbraucher-sicher-online.de/anleitung/bildfolge-enigmail-installieren-in-mozilla-thunderbird>

Nach der Installation der erforderlichen Tools kann man mit der Schlüsselerzeugung beginnen.

In *Thunderbird* findet sich nach der Installation ein neuer Menüpunkt *OpenPGP*. Außerdem gibt es auch in den Einstellungen jedes Email-Kontos einen Punkt für die kontenspezifische PGP-Konfiguration.

Nun können die meisten Arbeiten der Schlüsselbund-Verwaltung, sowohl in *Gpg4win* / *GPG Keychain Access* als auch in *Enigmail* innerhalb von *Thunderbird* erledigt werden. Die Plattform-spezifischen Schlüsselverwaltungstools bieten bei der Schlüsselerzeugung einige Optionen mehr.

Ich werde mich im Folgenden in erster Linie auf *Enigmail/Thunderbird* beziehen und auf die Detailinformationen verzichten, da sehr gute und detailreiche Anleitungen, Tutorials und FAQs im Web zu finden sind. Folgenden URLs sind zu empfehlen:

- *Thunderbird* und *Enigmail* *OpenPGP*:
http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP
- *Gpg4win*:
<http://gpg4win.de/handbuecher/einsteiger.html>
- *GPG Keychain Access*:
<http://support.gpgtools.org/kb>

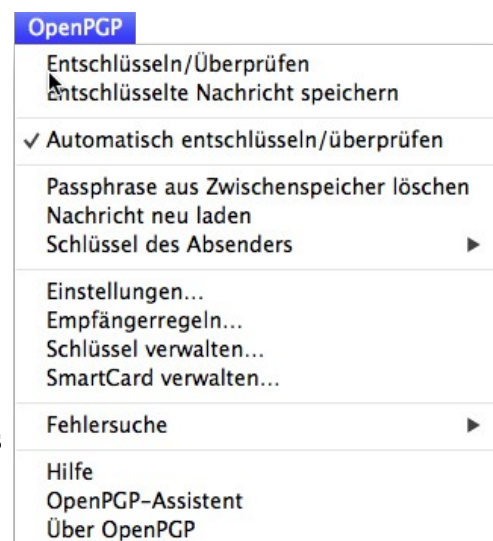


Abbildung 11: *Thunderbird*: Das neue *OpenPGP*-Menü nach der Installation von *Enigmail*

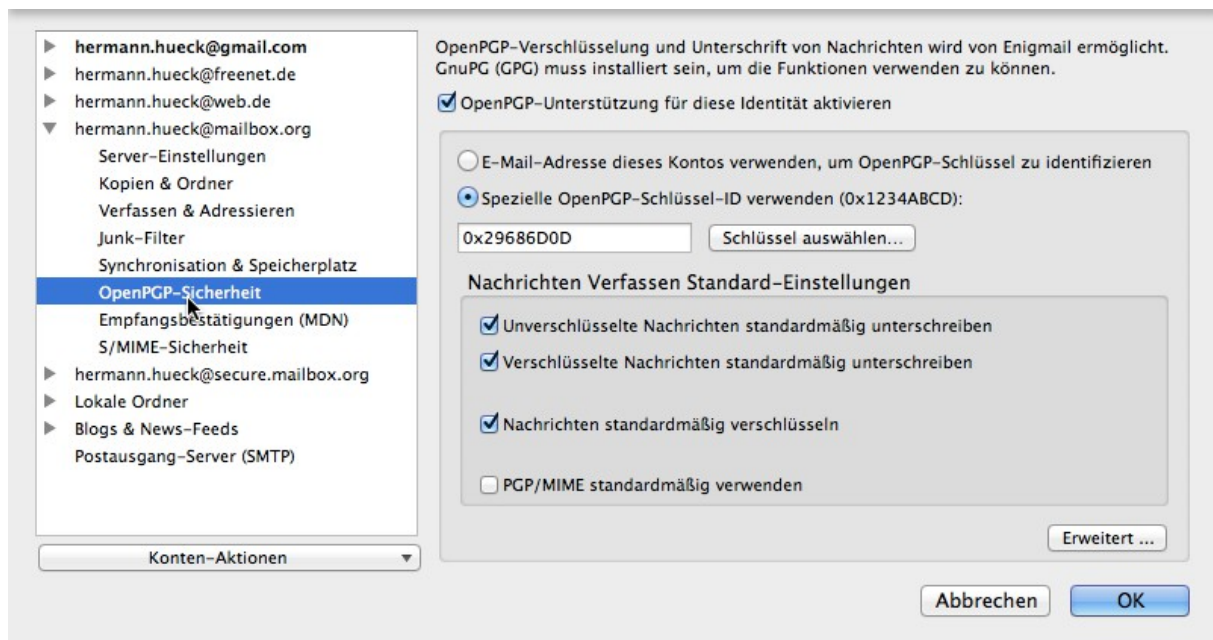


Abbildung 12: Thunderbird: Konten-spezifische OpenPGP-Konfiguration

Auf allen Betriebssystemen kann man immer auch die Kommandozeile verwenden. So lassen sich z.B. mit dem Kommando

```
gpg -list-keys
```

alle im Schlüsselbund enthaltenen Schlüssel anzeigen.

Die Manual-Seite für das gpg-Kommando findet man hier:

<https://www.gnupg.org/documentation/manpage.html>

Die Verwendung der Kommandozeile bietet sehr viele Optionen und sind eher für den IT-Experten gedacht. Der IT-Laie kann die Verschlüsselung auch ohne die Verwendung der Kommandozeile realisieren.

In den folgenden Kapiteln werde ich exemplarisch auch die Verwendung des gpg-Kommandos zeigen.

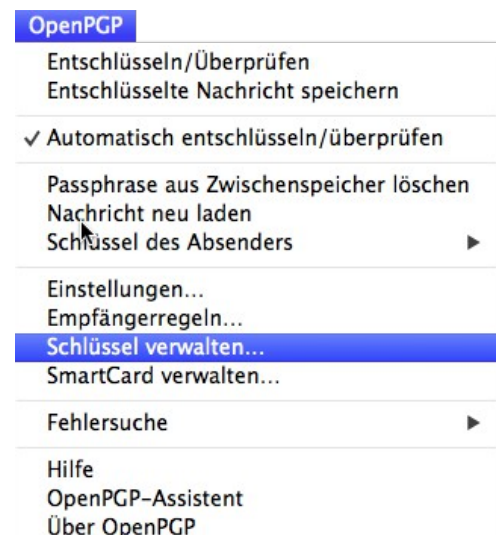


Abbildung 13: Thunderbird: Schlüsselbund öffnen

3.4.2 Erzeugung eines neuen Schlüsselpaars

3.4.2.1 Schlüsselerzeugung mit Enigmail / Thunderbird

Hier startet man in *Thunderbird* über den Menüpunkt *OpenPGP* → *Schlüssel verwalten ...* die *Enigmail*-Schlüsselverwaltung, die sich in einem neuen Fenster öffnet. Beim ersten Öffnen ist der Schlüsselbund leer, es werden keine Schlüssel angezeigt. In der Schlüsselverwaltung wählt man den Menüpunkt *Erzeugen* → *Neues Schlüsselpaar ...*. In einem neuen Fenster definiert man



Abbildung 14: Enigmail: Neues Schlüsselpaar erzeugen

- Die *Benutzer-ID* des neuen Schlüssels. Das ist der

Benutzername und das Email-Konto, für das der Schlüssel gelten soll.

- Die *Passphrase*. Diese sollte nicht zu einfach sein, denn sie schützt den Schlüssel vor Missbrauch. Die Passwortregeln (siehe Kap. 4.1) gelten auch hier. Der Schlüssel wird nach der Erstellung in der Schlüsselliste angezeigt.
- Einen optionalen Kommentar
- Die Gültigkeitsdauer des Schlüssels

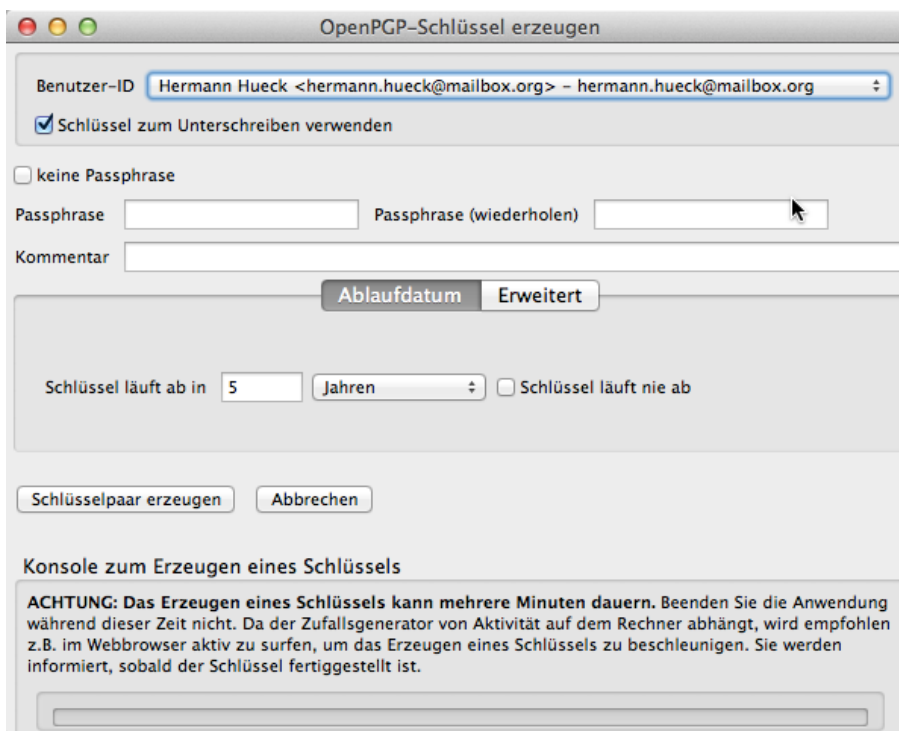


Abbildung 15: Enigmail: Dialog zum Erzeugen eines neuen Schlüsselpaars

gut und detailliert erläutert:

http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Schl%C3%BCsselverwaltung#Ein_Schl.C3.BCsselpaar_erzeugen

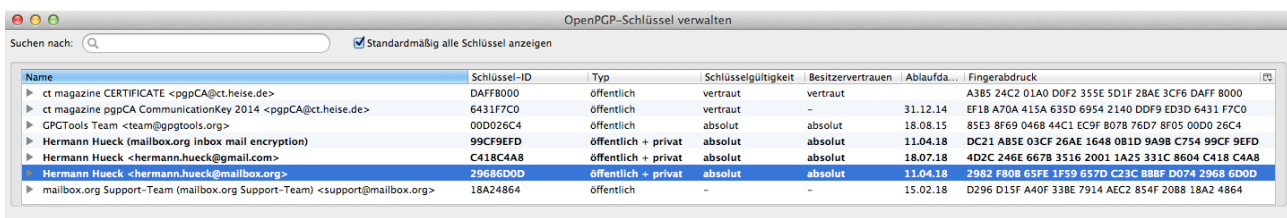


Abbildung 16: Enigmail: Der Schlüsselbund mit dem neu erzeugten Schlüssel (markiert)

Mit einem Klick auf *Schlüssel erzeugen* wird ein Schlüssel für das angegebene Email-Konto erzeugt. (Dieser Schlüssel ist 2048 Bit lang und unterstützt das Verschlüsselungsverfahren RSA.)

Anschließend wird man aufgefordert, ein Widerrufs-zertifikat zu erzeugen. Dies lässt sich auch später noch nachholen. (siehe Kap. 3.4.3)

Der neue Schlüssel wird nun im Schlüsselbund angezeigt und lässt sich weiter bearbeiten.

An dieser Stelle wird die Schlüsselgenerierung mit *Thunderbird/Enigmail* sehr

3.4.2.2 Limitationen von *Enigmail* / *Thunderbird*

Einige wenige Optionen stehen in der *Enigmail*-Schlüsselverwaltung nicht zur Verfügung. Will man diese zusätzlichen Optionen nutzen, muss man die Schlüsselverwaltungen *Gpg4win* (Windows) und *GPG Keychain Access* (Mac OS X) oder die Kommandozeile für die Schlüsselgenerierung nutzen.

- Als Verschlüsselungsverfahren wird bei *Enigmail* immer RSA verwendet. Da dies das meist verwendete Verfahren ist, stellt dies in der Regel keine Einschränkung dar. Wer andere Verschlüsselungsverfahren benötigt, muss eine der genannten Alternativen verwenden.
- Die Schlüssellänge ist auf 2048 Bit festgelegt. Andere Schlüssellängen (1024, 2048, 3072 und 4096 Bit) sind nicht wählbar. Mit Schlüsseln ist es wie mit Passwörtern – je länger, desto besser, denn lange Schlüssel und lange Passwörter sind schwerer zu knacken. Schlüssel mit einer Länge von 1024 Bit gelten heute noch als sicher. Aber da die Rechner immer leistungsfähiger werden, kann diese Aussage in zwei Jahren schon nicht mehr richtig sein. Mit einem 2048 Bit langen Schlüssel ist man für ein paar Jahre wohl auf der sicheren Seite. Es spricht aber nichts dagegen, einen Schlüssel mit einer Länge von 4096 Bit zu erzeugen und statt *Enigmail* *Gpg4win* oder *GPG Keychain Access* oder die Kommandozeile zu verwenden.
- Bei *Enigmail* ist ein Schlüssel immer einer Mail-Adresse zugeordnet. Das muss nicht zwingend so sein, etwa wenn man einen Schlüssel zur Dateiverschlüsselung und nicht zur Verschlüsselung von Emails verwenden will. Auch für diesen Fall ist *Enigmail* nicht das richtige Werkzeug. Mit *Gpg4win* oder *GPG Keychain Access* oder auf der Kommandozeile ist das kein Problem.

3.4.2.3 Schlüsselerzeugung mit *Gpg4win*

Um alle Optionen der Schlüsselerzeugung und -verwaltung unter Windows zu nutzen, ist *Gpg4win* zu verwenden. Siehe http://gpg4win.de/handbuecher/einsteiger_7.html.

3.4.2.4 Schlüsselerzeugung mit *GPG Keychain Access*

Um alle Optionen der Schlüsselerzeugung und -verwaltung unter Mac OS X zu nutzen, ist *GPG Keychain Access* zu verwenden. Siehe <http://support.gpgtools.org/kb>.

3.4.2.5 Schlüsselerzeugung auf der Kommandozeile

Auf allen Betriebssystemen (Windows, Mac OS X und Linux) kann man zur Schlüsselerzeugung auch die Kommandozeile verwenden.

Das Kommando

```
gpg --gen-key
```

fragt vom Benutzer alle benötigten Information (Verschlüsselungsverfahren, Schlüssellänge, Gültigkeitsdauer, Benutzer-ID, Schlüssellänge, etc.) ab und erzeugt dann das Schlüsselpaar.

Alle Schlüssel des Schlüsselbundes (einschließlich des neu erzeugten) lassen sich mit folgendem Kommando anzeigen.

```
gpg -list-keys
```

3.4.3 Das Widerrufszeug

Unbedingt sollte man zu jedem eigenen Schlüssel ein **Widerrufszeug (KRC, Key Revocation Certificate)** erzeugen und an einem sicheren Ort (CD oder USB-Stick, der nur für Schlüssel verwendet wird) speichern. Mit dem Widerrufszeug kann ein öffentlicher Schlüssel widerrufen werden, auch wenn man die Passphrase nicht kennt. Erlangt eine fremde Person Zugriff auf das Widerrufszeug, so kann diese den Schlüssel, für den das Zeug erstellt wurde, widerrufen.

Der widerrufenen Schlüssel wird damit unbrauchbar.

Die Erstellung eines Widerrufszertifikates kann man mit Enigmail erledigen oder mit den Plattformspezifischen Tools oder auch auf der Kommandozeile.

Das Kommando

```
gpg --gen-revoke <key-id> --output revoke-cert.asc
```

erzeugt das Widerrufszertifikat und speichert es in der Datei *revoke-cert.asc*.

Es ist sinnvoll, das Widerrufszertifikat gleich bei der Schlüsselerzeugung mit erzeugen zu lassen.

3.4.4 Weitere Benutzer-IDs (Email-Adressen) hinzufügen

Will man den Schlüssel nicht nur mit einer Email-Adresse nutzen, können weitere Benutzer-IDs hinzugefügt werden. Die Benutzer-ID ist eine Kombination aus Name (normalerweise Vor- und Nachname) und Email-Adresse. Der Name der ersten Email-Adresse darf sich wiederholen; als Email-Adresse gibt man die zweite Email-Adresse an. Ebenso können weitere Benutzer-IDs mit je einer weiteren Email-Adresse hinzugefügt werden.

Man öffnet die *Enigmail*-Schlüsselverwaltung. Im Kontextmenü jedes Schlüssels ist die Option *Benutzer-IDs verwalten ...* zu wählen. In einem neuen Fenster öffnet sich die Liste der Benutzer-IDs. Nach der Erzeugung des Schlüssels enthält die Liste genau einen Eintrag. Nun kann man weitere hinzufügen oder bestehende löschen oder die primäre Benutzer-ID neu festlegen. In der Schlüsselliste des Schlüsselbundes wird immer die primäre Benutzer-ID angezeigt.

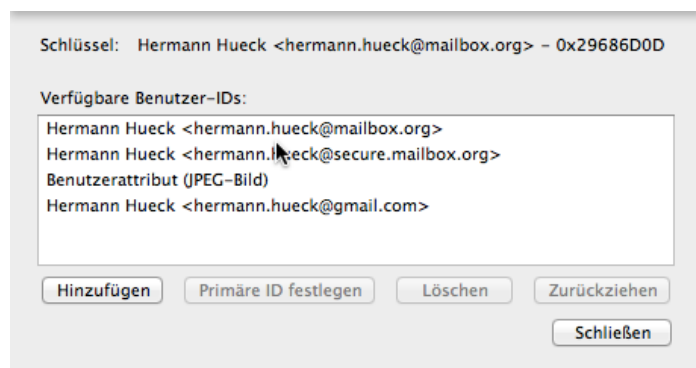


Abbildung 17: Enigmail: Liste der Benutzer-IDs eines Schlüssels

Das Kommando

```
gpg --edit-key <key-id>
```

dient der Änderung eines Schlüssels. Danach lassen sich auch weitere Benutzer-IDs hinzufügen.

3.4.5 Die wichtigsten PGP-Schlüsseleigenschaften

Primäre Benutzer-ID: Hermann Hueck <hermann.hueck@mailbox.org>

Schlüssel-ID: 0x29686D0D

Typ: Schlüsselpaar

Schlüsselgültigkeit: absolut

Besitzervertrauen: absolut

Fingerabdruck: 2982 F80B 65FE 1F59 657D C23C B8BF D074 2968 6D0D

Weitere Benutzer-ID	Gültig
Hermann Hueck <hermann.hueck@secure.mailbox.org>	absolut
Hermann Hueck <hermann.hueck@gmail.com>	absolut

Schlüssel...	ID	Algorit...	Stä...	Erzeugt	Ablaufda...	Verwendung
Primär...	0x29686D0D	RSA	4096	11.04.14	11.04.18	Verschlüsseln, Unterschreiben, Beglaubigen, Aut...
Unters...	0x8785ED05	RSA	4096	11.04.14	11.04.18	Verschlüsseln, Unterschreiben, Authentifizieren

Aktion wählen... ▼

Schließen

Abbildung 18: Enigmail: Schlüsselattribute

Die wichtigsten Schlüsselattribute sind:

- **Schlüssel-ID** (obligatorisch, unveränderlich): Die Schlüssel-ID (oder key id) besteht aus den letzten 8 Zeichen des Fingerabdrucks. Sie ist (anders als der Name des Attributs vermuten lässt) kein eindeutiger Identifikator für den Schlüssel.
- **Fingerabdruck** (obligatorisch, unveränderlich): Der Fingerabdruck (oder finger print) ist eine **eindeutige Kennzeichnung** für den Schlüssel. Der Fingerabdruck besteht aus 40 Zeichen und ist identisch beim privaten und beim öffentlichen Schlüssel. So lässt sich auch die Zusammengehörigkeit der beiden Schlüssel eines Schlüsselpaars nachweisen.
- **Schlüssel-Typ** (obligatorisch, unveränderlich): Der Schlüssel-Typ ist entweder *öffentlich* oder *öffentlich+privat*. Ist der Typ *öffentlich+privat*, dann handelt es sich um ein eigenes Schlüsselpaar. Ist der Typ *öffentlich*, dann ist es der öffentliche Schlüssel einer anderen Person, den man in die eigene Schlüsselverwaltung importiert hat.
- **Erstellungsdatum** (obligatorisch, unveränderlich): Das Erstellungsdatum wird bei der Schlüsselerzeugung festgelegt. Es ist das Datum der Schlüsselerzeugung.
- **Ablaufdatum** (optional, änderbar): Das Ablaufdatum kann bei der Schlüsselerzeugung festgelegt und nachträglich geändert werden. Zum Ablaufdatum des Schlüssels wird der Schlüssel automatisch ungültig. Ein Schlüssel ohne Ablaufdatum ist unbegrenzt gültig.
- **Primäre Benutzer-ID** (obligatorisch, änderbar): Dies ist meist die Benutzer-ID (Name und Email-Adresse), die man bei der Schlüsselerzeugung angegeben hat. Man kann jedoch eine weitere Benutzer-ID erzeugen und diese nachträglich zur primären Benutzer-ID machen.
- **Weitere Benutzer-IDs** (optional, änderbar): Beliebig viele weitere Benutzer-IDs können festgelegt werden. In der Regel definiert man für jede weitere Email-Adresse, mit der man den Schlüssel verwenden will, eine weitere Benutzer-ID.
- **Fotos** (optional, änderbar): Optional können ein oder mehrere Fotos hinzugefügt werden.
- **Kommentar zu jeder Benutzer-ID des Schlüssels** (optional, unveränderlich): Zu jeder Benutzer-ID, die dem Schlüssel hinzugefügt wird, kann optional ein Kommentar angegeben werden. Der Kommentar ist nicht änderbar. Er wird jedoch gelöscht, wenn die Benutzer-ID

gelöscht wird.

- **Beglaubigungen zu jeder Benutzer-ID des Schlüssels** (optional, änderbar): Zu jeder Benutzer-ID kann man eine oder mehrere Beglaubigungen hinzufügen. Eine Beglaubigung ist ein Vertrauensbeweis, den man mit dem eigenen Schlüssel signiert/unterschreibt. Der Schlüssel des Beglaubigenden wird in die Beglaubigung eingetragen. Typischerweise beglaubigt man die öffentlichen Schlüssel anderer PGP-Nutzer, die man in die eigene Schlüsselverwaltung aufgenommen hat und denen man vertraut.
- **Vertrauensstufe oder Besitzervertrauen** (obligatorisch, änderbar): Jeder (eigene oder fremde) Schlüssel in der Schlüsselverwaltung hat eine Vertrauensstufe. Diese drückt aus, wie sehr ich dem betreffenden Schlüssel vertraue. Das Besitzervertrauen ist mit einer von 5 möglichen Vertrauensstufen einstellbar:
 - undefiniert (Ich weiß es nicht.)
 - Nie (Ich vertraue ihm nicht.)
 - Marginal (Ich vertraue ihm nur gering.)
 - Vollständig (Ich vertraue ihm voll.) Dies ist die höchste/beste Vertrauensstufe für die Schlüssel von Kommunikationspartnern.
 - Absolut (Ich vertraue ihm absolut.) Diese Vertrauensstufe sollte man nur für die eigenen Schlüssel verwenden.

3.4.6 Schlüssel exportieren und importieren

In der Schlüsselverwaltung kann man einen Schlüssel in eine Datei exportieren, um ihn z.B. zu auf einen anderen Datenträger zu sichern oder auf einen anderen Rechner zu übertragen. Das macht man typischerweise mit einem eigenen Schlüsselpaar, das aus Private Key und Public Key besteht.

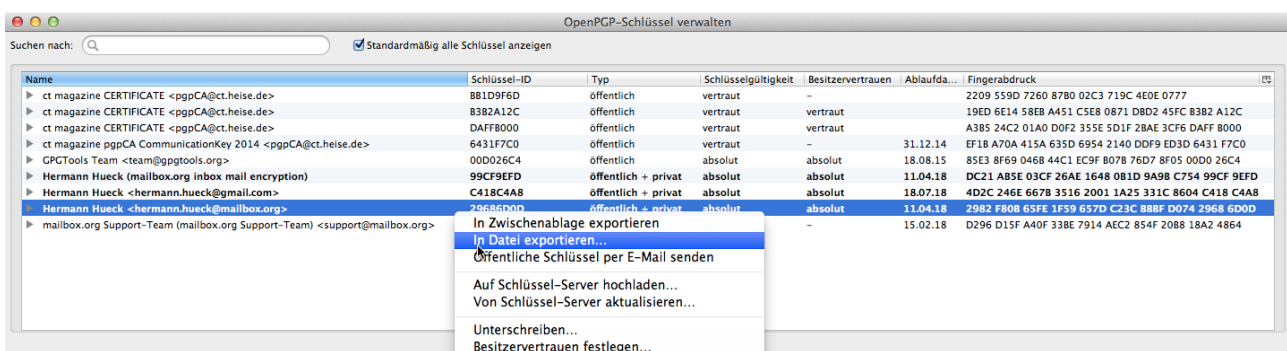


Abbildung 19: Enigmail: Den markierten Schlüssel exportieren

Man kann einen Schlüssel aus einer Datei in die Schlüsselverwaltung importieren. Dies kann der zuvor exportierte eigene Schlüssel sein oder auch ein fremder Public Key, der in Dateiform vorliegt.

3.4.6.1 Schlüssel sicher aufbewahren

Das eigene Schlüsselpaar (bzw. die eigenen Schlüsselpaare, so man mehrere hat) sollte man nicht verlieren. Damit es auch einen Platten-Crash überlebt(en), muss jedes eigene Schlüsselpaar exportiert und auf ein externes Medium (USB-Stick oder CD) gesichert und gut aufbewahrt werden. Das Widerrufszeug kann man dabei gleich mit sichern.

Von diesem Medium kann man den/die Schlüssel wiederherstellen; d.h. wieder in die Schlüsselverwaltung des Rechners oder auch eines zweiten Rechners importieren.

WARNUNG: Was man **NIE TUN** sollte:

- Den privaten Schlüssel als Anhang einer unverschlüsselten Email verschicken – auch nicht an sich selbst, z.B. um ihn auf einen anderen Rechner zu übertragen. Unterwegs könnte er in die falschen Hände geraten.
- Den privaten Schlüssel unverschlüsselt in die Cloud (z.B. in die Dropbox) stellen. Auch das wäre ein bequemer, aber sehr gefährlicher Weg, um den Schlüssel auf einen anderen Rechner zu übertragen.

3.4.7 Konfiguration der Key-Server (Enigmail)

Bevor man den/die (öffentlichen) Schlüssel mit einem Schlüssel-Server synchronisiert (siehe Kap. 3.4.8), ist es sinnvoll, die in *Enigmail* eingestellten Schlüssel-Server zu prüfen und ggf. anzupassen. Ob die dort angegebenen Schlüssel-Server tatsächlich unter ihrem Namen noch erreichbar sind, kann man testen, indem man die Server-Namen in die URL-Zeile des Browsers eingibt.

Unter *OpenPGP* → *Einstellungen* → *Schlüssel-Server* können die Schlüssel-Server festgelegt werden, die beim Import oder Export öffentlicher Schlüssel zur Auswahl stehen sollen. Werden mehrere angegeben, werden sie durch ein Komma getrennt. In der Konfiguration meines Rechners sind z.B. folgende Schlüssel-Server eingetragen.

- a.keyserver.pki.scientia.net
- p80.pool.sks-keyservers.net
- pgp.mit.edu
- subkeys.pgp.net

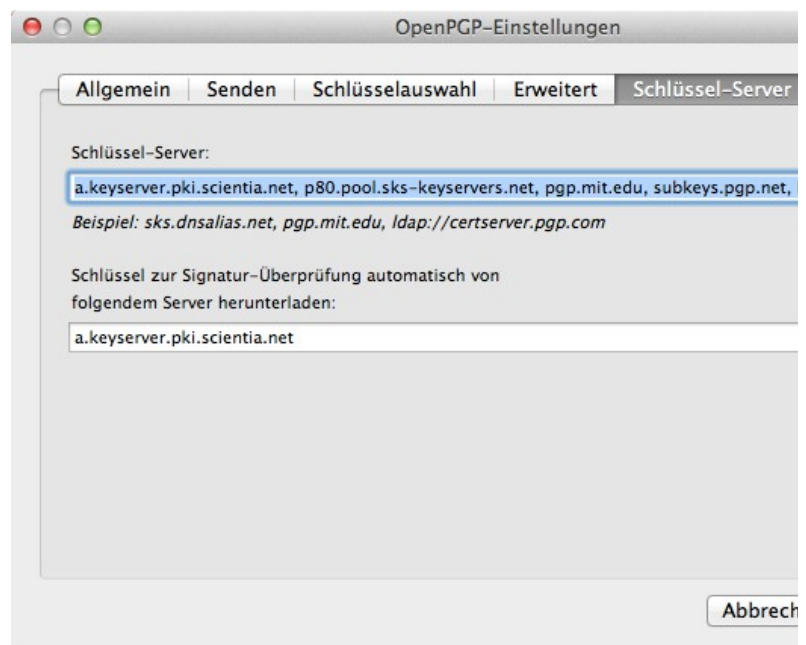


Abbildung 20: Enigmail: Konfiguration der Schlüssel-Server

Auf den Seiten von Heise findet sich ebenfalls eine Empfehlung für die einzustellenden Schlüssel-Server unter der URL: <http://www.heise.de/security/dienste/Keyserver-474468.html>.

Nutzt man die plattform-spezifischen Schlüsselbund-Verwaltungen (*Gpg4win* unter Windows bzw. *GPG Keychain Access* unter OS X) zur Synchronisation der öffentlichen Schlüssel, so müssen auch dort die Key-Server konfiguriert werden. Dies erledigt man in den Einstellungen des betreffenden Programms.

3.4.8 Öffentliche Schlüssel mit Key-Server synchronisieren

Key-Server halten die öffentlichen PGP-Schlüssel vieler Benutzer bereit. Um diese Schlüssel weltweit vorzuhalten, gibt es eine ganze Reihe davon über das globale Internet verteilt. Man muss allerdings nicht jeden Key-Server mit seinem öffentlichen Schlüssel versorgen, sondern man lädt

den Schlüssel auf einen Key-Server hoch. Die Key-Server synchronisieren sich automatisch – normalerweise innerhalb von 24 Stunden (und sie verwenden dazu ein eigenes Kommunikationsprotokoll). Sie heißen deshalb auch **SKS** oder **Synchronizing Key Server**.

Auf den Key-Servern findet man also die öffentlichen Schlüssel aller Benutzer, die PGP zum Verschlüsseln und Signieren von Mails verwenden. Benötigt man den öffentlichen Schlüssel eines bestimmten Benutzers, z.B. um die Mail an ihn zu verschlüsseln, kann man mit der Email-Adresse des Benutzers nach dem zugehörigen Schlüssel suchen, ihn herunterladen und in die eigene Schlüsselverwaltung importieren. Erst wenn der Schlüssel in der Schlüsselverwaltung vorliegt, kann der Mail-Client (*Thunderbird*) den Schlüssel verwenden, um die Mail an den Benutzer zu verschlüsseln oder um die Signatur in der Mail von dem Benutzer zu prüfen.

Hat man den öffentlichen Schlüssel eines anderen Benutzers beglaubigt, sollte man diesen wieder auf den Key-Server hochladen. Der aktualisierte Schlüssel mit der zusätzlichen eigenen Beglaubigung ist in der Regel auch für andere Benutzer wertvoller. So können die öffentlichen Schlüssel auf den Key-Servern im Laufe der Zeit immer mehr Beglaubigungen ansammeln und werden mit jeder neuen Beglaubigung vertrauenswürdiger. Beglaubigungen können auch entzogen werden. Alle Schlüssel des eigenen Schlüsselbundes sollten immer wieder mit dem Schlüssel-Server synchronisiert werden, damit sie, was die Beglaubigungen betrifft, immer auf dem aktuellen Stand sind.

Wie macht man's in der *Enigmail*-Schlüsselbund-Verwaltung?

- **Einzelnen Schlüssel auf Key-Server hochladen:** Im Kontextmenü eines Schlüssels oder mit dem Menüpunkt *Schlüssel-Server* → *Schlüssel hochladen ...* kann man den gewählten Schlüssel hochladen. Markiert man mehrere Schlüssel, kann man diese Aktion auch für mehrere Schlüssel in einem Schritt durchführen. **Vorsicht! Niemals einen Test-Schlüssel hochladen!** Ein hochgeladener Schlüssel kann nie mehr vom Schlüssel-Server gelöscht werden. Er kann nur widerrufen werden. Er bleibt aber als widerrufener und damit ungültiger Schlüssel auf dem Key-Server.
- **Schlüssel auf Key-Server suchen:** Mit dem Menüpunkt *Schlüssel-Server* → *Schlüssel suchen ...* kann man ein Suchkriterium eingeben. Man erhält eine Ergebnisliste mit allen Schlüsseln, die dem Suchkriterium entsprechen und kann unter den Treffern auswählen und die gewählten Schlüssel in den eigenen Schlüsselbund importieren. Zweckmäßigerweise gibt man die Email-Adresse der Person ein, mit der man verschlüsselten Mailverkehr pflegen will.
- **Einen Schlüssel von Key-Server aktualisieren:** Mit dieser Option aus dem Kontextmenü eines Schlüssels oder mit dem Menüpunkt *Schlüssel-Server* → *Schlüssel aktualisieren ...* kann man den betreffenden Schlüssel vom Key-Server aktualisieren. Damit erhält der Schlüssel die aktuellen Beglaubigungen. Auch ein geändertes Ablaufdatum oder den Widerruf eines Schlüssels bekommt die eigene Schlüsselbund-Verwaltung und damit auch *Thunderbird* nur auf diese Weise mit.

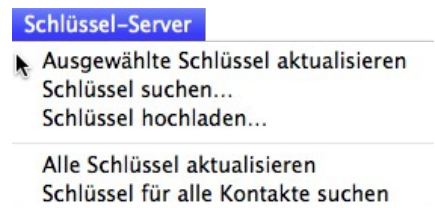


Abbildung 21: Enigmail: Optionen für die Synchronisation mit einem Schlüssel-Server

Markiert man mehrere Schlüssel, kann man diese Aktion auch für mehrere Schlüssel in einem Schritt durchführen.

- **Alle Schlüssel von Key-Server aktualisieren:** Mit dem Menüpunkt *Schlüssel-Server* →

Alle Schlüssel aktualisieren kann man alle Schlüssel der Schlüsselverwaltung vom Key-Server aktualisieren.

- **Schlüssel für alle Kontakte suchen:** Mit dieser Option sucht *Enigmail* die Schlüssel für alle Email-Adressen aus dem *Thunderbird*-Adressbuch und bietet dann alle gefundenen Schlüssel zum Import in den Schlüsselbund an. Man kann immer noch die Schlüssel auswählen, die man importieren möchte. Je nach Größe der Kontaktliste und nach Anzahl der gefundenen Schlüssel kann dieser Vorgang recht lange dauern. Bei mir war es mehr als eine Stunde.

3.4.9 Schlüssel beglaubigen

Die öffentlichen Schlüssel anderer Benutzer, denen man vertraut, kann und sollte man beglaubigen, d.h. sie unterschreiben bzw. signieren. Dadurch vergrößert man das WoT, das Web of Trust, also das Netz aus Vertrauensbeziehungen (siehe Kap. 3.3).

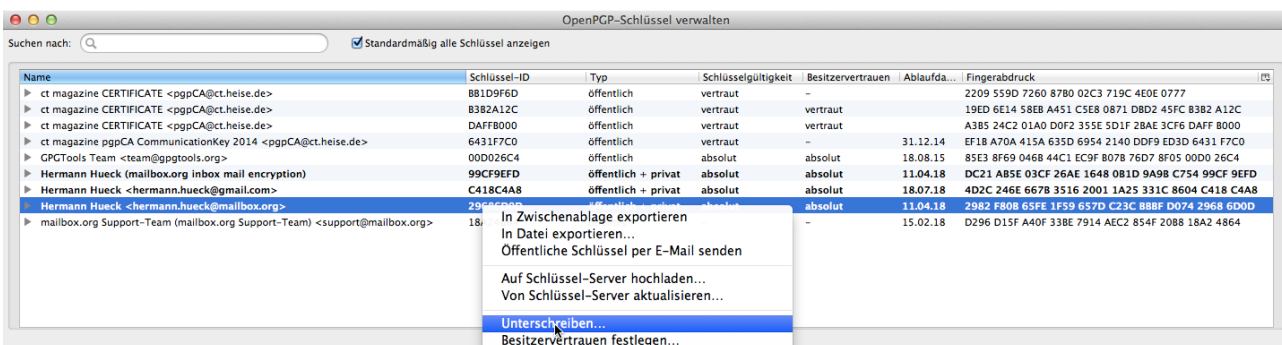


Abbildung 22: Enigmail: Den markierten Schlüssel unterschreiben/beglaubigen

In einem Schlüssel können mehrere Benutzer-IDs (bestehend aus dem Namen und der Email-Adresse des Benutzers) eingetragen sein. Die Benutzer-IDs werden vom Besitzer des Schlüssels gepflegt und können von anderen Benutzern beglaubigt werden. Eine Beglaubigung ist die Bestätigung der Verknüpfung eines Schlüssels mit einer Benutzer-ID. Dazu muss der Beglaubigende die Benutzer-ID mit seinem eigenen Schlüssel signieren/unterschreiben. Die Schlüssel-ID des Beglaubigenden wird (analog zu Stempel und Unterschrift eines Notars) in die Beglaubigung eingetragen.

Damit die Beglaubigung meines Schlüssels durch die Unterschrift mit dem Schlüssel einer anderen Person einen Wert für mich hat, muss ich den öffentlichen Schlüssel des Beglaubigenden ebenfalls vom Schlüssel-Server in meine Schlüsselverwaltung importieren (siehe Kap. 3.4.8). Nach dem Import muss ich diesen Schlüssel ebenfalls signieren und als Besitzervertrauen „vollständiges Vertrauen“ eintragen.

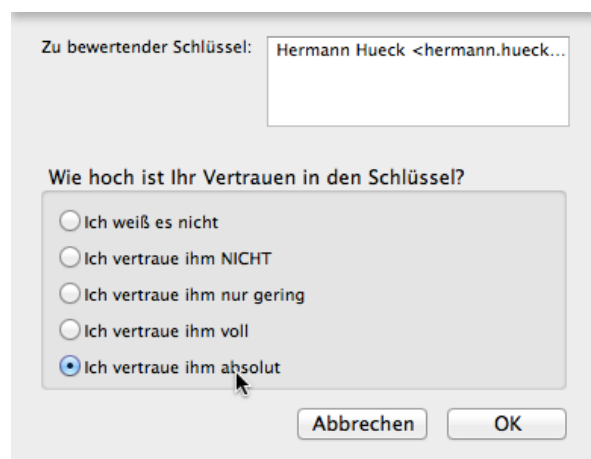


Abbildung 23: Enigmail: Definieren des Besitzervertrauens

3.4.9.1 Verifikation von Schlüssel-Identität und Benutzer-Identität

Bevor ich einen fremden Schlüssel (den öffentlichen Schlüssel eines Kommunikationspartners) beglaubigt, muss man in zunächst in den eigenen Schlüsselbund importieren. Dazu kann der Kommunikationspartner mir eine (am besten signierte) Mail mit seinem öffentlichen Schlüssel als Anhang zusenden oder er exportiert seinen Schlüssel auf einen Schlüssel-Server und ich lade ihn vor dort herunter und importiere ihn in den Schlüsselbund (siehe Kap. 3.4.8).

Der Beglaubigende verbürgt sich mit seiner Unterschrift, d.h. mit seinem eigenen Schlüssel, für die beglaubigte Benutzer-ID und damit implizit auch für die Email-Adresse, die Teil der Benutzer-ID ist. Deshalb sollte er, bevor er die Benutzer-ID eines Schlüssels beglaubigt, sowohl die Identität des Schlüsselinhabers als auch die Email-Adresse und die Identität des Schlüssels, den er beglaubigt, prüfen.

- **Prüfung der Person:** Um die Identität eines Benutzers zu prüfen, kann man sich wie ein Grenzbeamter den Lichtbildausweis (Pass, Personalausweis, Führerschein, BahnCard oder Versicherungskarte) zeigen lassen. Kennt man den Schlüsselbesitzer persönlich, kann allein diese Bekanntschaft schon als Prüfung der Benutzer-Identität gelten. Bei einem Telefonat erkenne ich den guten Freund auch an der Stimme; dies kann als Identitätsnachweis ausreichend sein.
- **Prüfung der Benutzer-ID und Email-Adresse:** Dies geschieht einfach dadurch, dass der Benutzer, dessen Schlüssel ich signieren will, mir eine signierte Mail sendet. Damit erhalte ich die Email-Adresse des Absenders und kann sie mit der Email-Adresse, die in der Benutzer-ID des zu signierenden Schlüssels eingetragen ist, abgleichen. Dazu ist auch der Fingerabdruck des Schlüssels zu prüfen.
- **Prüfung der Schlüssel-Identität mit Hilfe des Schlüssel-Fingerabdrucks:** Der Beglaubigende muss den Fingerabdruck, der vom Schlüsseleigentümer genannt wird, mit dem Fingerabdruck des öffentlichen Schlüssels abgleichen.

Man könnte sich das so vorstellen: Der Schlüsselbesitzer schreibt seine Mail-Adresse und den Fingerabdruck des (privaten) Schlüssels auf ein Papier und übergibt dieses (persönlich oder per Brief oder Fax) an den Beglaubigenden. Der Beglaubigende hat den öffentlichen Schlüssel heruntergeladen oder per Mail zugesandt bekommen und in seine Schlüsselverwaltung importiert. Jetzt kann er prüfen, ob der auf dem Papier übergebene Fingerabdruck mit dem des öffentlichen Schlüssels übereinstimmt. Ist der Schlüsselinhaber ein guter Bekannter, kann er den 40-stelligen Fingerabdruck dem Beglaubigenden auch am Telefon vorlesen.

Wenn man in der *Enigmail*-Schlüsselverwaltung einen Schlüssel unterschreibt/signiert, beglaubigt man immer automatisch alle Benutzer-IDs (und damit alle Email-Adressen) dieses Schlüssels. Dies ist in den allermeisten Fällen auch korrekt und erwünscht. Will man nicht alle, sondern nur eine bestimmte Benutzer-ID eines Schlüssels beglaubigen, so muss man das in der plattform-spezifischen Schlüsselverwaltung *Gpg4win* oder *GPG Keychain Access* oder auf der Kommandozeile erledigen.

Ein paar praktische Beispiele demonstrieren das Beglaubigungsverfahren.

- In einem **Telefonat** kann ich einem Freund den Fingerabdruck meines Schlüssels und meine Email-Adresse vorlesen. Dieser erkennt mich an der Stimme (Nachweis der Benutzer-Identität) und prüft den Fingerabdruck des öffentlichen Schlüssels in einer von mir signierten, an ihn gerichteten Mail (Nachweis der Schlüssel-Identität). Er importiert meinen Public Key in seine Schlüsselverwaltung und er beglaubigt, d.h. er signiert ihn. Danach lädt

er meinen von ihm beglaubigten öffentlichen Schlüssel auf einen Key-Server hoch (siehe Kap. 3.4.8). Ich und andere Benutzer können ihn von dort wieder aktualisieren; so haben wir den beglaubigten Schlüssel dann in unseren Schlüsselverwaltungen. Anschließend müssen beide noch das Besitzervertrauen für den Schlüssel des jeweils anderen auf „vollständiges Vertrauen“ einstellen.

Das geht natürlich genau so gut bei einem persönlichen Treffen. Oder man übermittelt Fingerabdruck und Email-Adresse(n) per Brief oder per Fax.

- Auf einer **Krypto-Party** oder einer **Key-Signing-Party** treffen sich Unbekannte mit ihren Ausweisen und mit ihren Laptops, um sich wechselseitig ihre PGP-Schlüssel zu beglaubigen. Mehr dazu unter <https://www.cryptoparty.in/> und unter <http://de.wikipedia.org/wiki/CryptoParty>
- Die c't vom Heise-Verlag fördert das Web of Trust mit der c't-Kryptokampagne. Sie demonstriert auch sehr gut, wie das PGP-Beglaubigungsverfahren funktioniert. Dies ist Gegenstand des nachfolgenden Unterkapitels.

3.4.9.2 c't-Kryptokampagne

Will man seinen öffentlichen Schlüssel von der c't signieren lassen, ist Folgendes zu tun:

- Man lädt seinen öffentlichen Schlüssel auf einen Key-Server hoch (siehe Kap. 3.4.8) und schickt ihn an die Heise-Mail-Adresse pgpCA@ct.heise.de. Bei Heise wird der öffentliche Schlüssel mit allen Benutzer-IDs gespeichert.
- Als Antwort erhält man für jede Benutzer-ID an die zugehörige Email-Adresse eine Bestätigungsmail. Für einen Schlüssel mit drei Benutzer-IDs erhält man drei Bestätigungsmails. Diese Mails sind mit dem eigenen öffentlichen Schlüssel verschlüsselt. Beim Empfang der Mails in Thunderbird werden diese mit dem privaten Schlüssel entschlüsselt und damit lesbar. In jeder Bestätigungsmail befindet sich ein Bestätigungslink.
- In jeder entschlüsselten Mail von Heise klickt man dann auf den Bestätigungslink und wird auf eine Web-Seite von Heise weitergeleitet. Darauf wird einem mitgeteilt, dass die Verknüpfung der Benutzer-ID (Email-Adresse) mit dem Schlüssel verifiziert wurde und von Heise als gültig anerkannt wird. Die **Prüfung der Benutzer-IDs und Email-Adressen** ist damit abgeschlossen. Jetzt muss noch die Verknüpfung der Person mit dem Schlüssel verifiziert werden. Dazu dienen die folgenden Schritte.
- Man lädt von der Website der c't-Kryptokampagne ein Formular für einen Zertifizierungsantrag herunter und druckt es aus. Man füllt das Formular aus, d.h. man trägt den eigenen Namen, die Email-Adresse der primären Benutzer-ID, die Schlüssel-ID, den Fingerabdruck des Schlüssels, die Personalausweisnummer ein und unterschreibt es.
- Das ausgefüllte Antragsformular legt man persönlich zusammen mit dem Personalausweis beim Heise-Verlag in Hannover oder auf dem Heise-Messestand bei der Cebit in Hannover oder bei der IFA in Berlin vor. Im Verlag oder am Messestand findet die **Personenprüfung** (Nachweis der Benutzer-Identität) statt: es wird geprüft, ob das Lichtbild das des anwesenden Antragstellers ist und ob die PA-Nummer im Ausweis der im Antragsformular entspricht. Der Antrag verbleibt bei Heise und wird im Verlag weiterbearbeitet.
- Beim Heise-Verlag kann ein Bearbeiter dann die **Schlüsselprüfung** (Nachweis der Schlüssel-Identität) durchführen, indem er den auf dem Formularbogen angegebenen

Fingerabdruck mit dem Fingerabdruck des zuvor zugesandten öffentlichen Schlüssels abgleicht. Der Bearbeiter kann den zugesandten Schlüssel signieren und auf einen Key-Server hochladen. Er schickt dem Antragsteller auch eine Mail, die ihm die Erledigung des Antrags mitteilt.

- Den eigenen, frisch signierten/beglaubigten Schlüssel kann man nun von einem Key-Server aktualisieren (siehe Kap. 3.4.8).
- Den Schlüssel des c't-Magazins, mit dem der eigene Schlüssel signiert/beglaubigt wurde, muss man von einem Key-Server herunterladen (siehe Kap. 3.4.8) und in den eigenen Schlüsselbund importieren. Dessen Fingerabdruck ist in jeder Ausgabe des c't-Magazins im Impressum abgedruckt und kann damit verifiziert werden. Nach der Verifikation muss man diesen noch unterschreiben und die Vertrauensstufe auf „volles Vertrauen“ einstellen. Damit erklärt man der Schlüsselverwaltung, dass man diesem Schlüssel vertraut. Damit erst hat die Beglaubigung wirklich einen Wert.

Durch den Umstand, dass der Heise-Verlag eine bekannte Institution ist, ist der Wert dieser Schlüssel-Beglaubigung sehr hoch einzustufen.

Genaue Erläuterungen zur c't-Kryptokampagne und Download des Antragsformulars und FAQ gibt es unter <http://www.heise.de/security/dienste/Krypto-Kampagne-2111.html>.

3.5 Signierter und verschlüsselter Mailverkehr

Der ganze Aufwand der Schlüsselverwaltung und -beglaubigung dient letztlich dazu, signierte und verschlüsselte Mails versenden zu können und empfangener Mails zu prüfen, bzw. sie zu entschlüsseln, um sie dann lesen zu können.

Die Voraussetzung hierfür ist, dass ein Schlüssel einem *Thunderbird*-Email-Account zugeordnet ist. Dann wird genau dieser (private) Schlüssel verwendet, um die aus diesem Account abgehenden Mails zu signieren und die in diesem Account empfangenen, verschlüsselten Mails zu entschlüsseln.

3.5.1 OpenPGP-Einstellungen für jedes Mail-Konto

Mit der Installation von *Enigmail* haben die Konteneinstellungen jedes in *Thunderbird* eingerichteten Mail-Kontos einen neuen Punkt OpenPGP-Sicherheit erhalten, unter dem die kontenspezifischen PGP-Einstellungen vorgenommen werden können.

Detaillierte Information zu dieser Konfiguration inklusive Screenshot ist hier zu finden:

http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Einstellungen#OpenPGP-Sicherheit

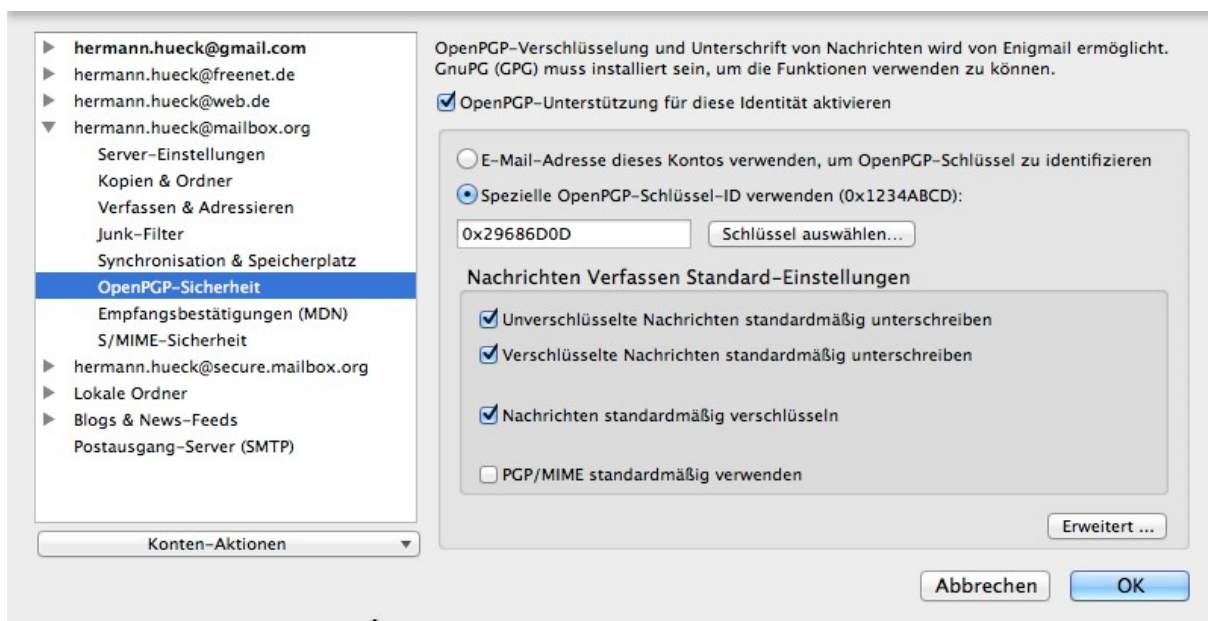


Abbildung 24: Enigmail: PGP-Einstellungen für ein Mail-Konto

Will man dieses Konto mit PGP nutzen, muss man mindestens den Hauptschalter umlegen (Haken setzen bei: *OpenPGP-Unterstützung für diese Identität* (sprich: Email-Adresse) *aktivieren*) und einen (privaten) Schlüssel des Schlüsselbundes für dieses Email-Konto auswählen.

Alle weiteren Einstellungen sind optional.

Die folgenden vier Einstellungen sind die Standardeinstellungen für den Mail-Versand und können auch vor dem Versenden einer Mail noch geändert werden:

- *Unverschlüsselte Nachrichten standardmäßig unterschreiben*. Dies halte ich für sinnvoll.
- *Verschlüsselte Nachrichten standardmäßig unterschreiben*. Dies halte ich auch für sinnvoll.
- *Nachrichten standardmäßig verschlüsseln*. Dies ist in der Regel nur sinnvoll, wenn man viele öffentliche Schlüssel von Kommunikationspartnern im eigenen Schlüsselbund hat. Fangt man mit der Verwendung von PGP gerade erst an, ist diese Voreinstellung nicht zu empfehlen.
- *Immer PGP/MIME verwenden*. Standardmäßig wird die Mail mit allen ihren Anhängen in einem Paket verschlüsselt. Mit PGP/MIME werden der Mail-Inhalt und jeder Anhang separat verschlüsselt. Eine solche Mail besteht aus mehreren verschlüsselten Teilen. PGP/MIME das modernere Format für den Mailversand. Da es noch viele Tools gibt, die dies nicht unterstützen, ist aktuell davon eher abzuraten. Wenn das Mail-Tool des Partners PGP/MIME nicht unterstützt, kann dieser die verschlüsselte Mail nicht entschlüsseln, obwohl er den richtigen Schlüssel zur Entschlüsselung der Nachricht in seinem Schlüsselbund hat. Auch die Apps auf dem Smartphone oder Tablet unterstützen dieses Format aktuell noch nicht (siehe Kap. 3.10.6).

Zwei weitere Einstellungen betreffen die (für den Benutzer normalerweise nicht angezeigten) Kopfzeilen der Mail, die sog. Mail-Header:

- *Sende OpenPGP-Schlüssel-ID*. Diese Option bestimmt, ob im Mail-Header eine Kopfzeile mit der Schlüssel-ID erstellt wird. Diese Option ist sinnvoll, jedoch nicht erforderlich.

- *Sende URL, um Schlüssel zu empfangen.* Diese Option bestimmt, ob im Mail-Header eine Kopfzeile mit der Download-URL des öffentlichen Schlüssels erstellt wird. Wählt man diese Option, ist auch die URL, an der der eigene öffentliche Schlüssel von einem Schlüssel-Server heruntergeladen werden kann, anzugeben. Diese Option ist sinnvoll, jedoch nicht erforderlich. Sie hilft dem Mail-Programm des Empfängers meiner Mail, meinen Schlüssel auf einem Schlüssel-Server zu finden und ggf. automatisch herunterzuladen.

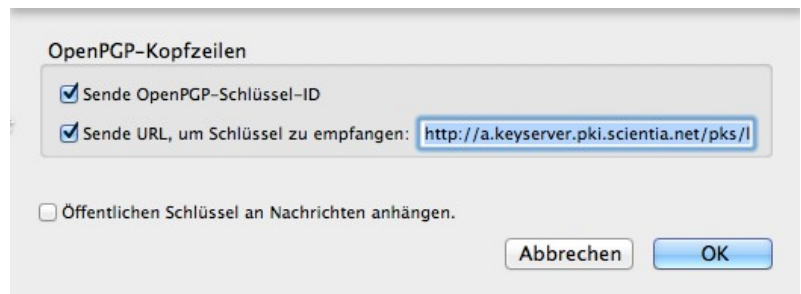


Abbildung 25: Enigmail: Weitere PGP-Optionen

Eine weitere Einstellung betrifft den Schlüssel-Versand:

- *Öffentlichen Schlüssel an Nachrichten anhängen.* Durch Setzen dieser Option wird der eigene öffentliche Schlüssel jeder ausgehenden Nachricht dieses Mail-Accounts automatisch als Attachment hinzugefügt. Damit kann man den eigenen öffentlichen Schlüssel an die Kommunikationspartner verteilen. Das ist jedoch nicht unbedingt erforderlich, da der Schlüssel von jedem auch vom Key-Server heruntergeladen werden kann. *Enigmail* lädt den Schlüssel eines Kommunikationspartners, der sich noch nicht im Schlüsselbund befindet, automatisch herunter und importiert ihn.

3.5.2 Versand und Empfang signierter und verschlüsselter Emails

Das Senden und Empfangen signierter und verschlüsselter Mails ist nach den ganzen Vorarbeiten eine einfache Angelegenheit geworden und funktioniert fast genau so wie der Versand und Empfang ohne Verschlüsselung und Signatur.

3.5.2.1 Versand

Zum Verfassen einer neuen Mail klickt man wie üblich auf den Button *Verfassen* oder man wählt eine bereits empfangene Mail und klickt auf *Antworten* oder auf *Weiterleiten*. Das Fenster zum Verfassen der Mail öffnet sich.

Das Fenster hat jetzt (nach der Installation von *Enigmail*) ein neues Schloss-Symbol mit der Beschriftung *OpenPGP*. Beim Klick auf das Symbol legt man die PGP-Versand-Optionen fest. Man kann bestimmen, ob vor dem Versand ...

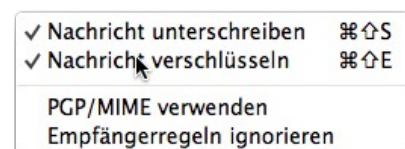


Abbildung 26: PGP-Versand-Optionen

- die Nachricht (mit dem privaten Schlüssel der versendenden Email-Adresse) unterschrieben werden soll,
- die Nachricht (mit dem öffentlichen Schlüssel der Email-Adresse des Empfängers) verschlüsselt werden soll
- und ob PGP/MIME als Verschlüsselungsformat verwendet werden soll.

3.5.2.2 Empfang

Nachrichten werden wie üblich empfangen. *Thunderbird* zeigt oben im Nachrichtenfenster die

PGP-Informationen der Nachricht an, falls die Nachricht signiert und/oder verschlüsselt ist.

Ist die Nachricht signiert und unverschlüsselt, kann man die Nachricht natürlich lesen. Allerdings kann *Thunderbird/Enigmail* die Signatur nicht überprüfen, wenn der öffentliche Schlüssel des Absenders nicht im Schlüsselbund enthalten ist. Diese Information wird angezeigt; *Thunderbird* bietet an, den fehlenden öffentlichen Schlüssel vom Schlüssel-Server herunterzuladen und in den Schlüsselbund zu importieren. Dann prüft *Thunderbird* die Nachricht erneut und zeigt die PGP-Informationen erneut an. Ggf. kann man anschließend (wenn man dem gerade importierten Schlüssel vertraut) den Schlüssel noch signieren, das Besitzervertrauen auf die geeignete Stufe einstellen und den Schlüssel dann wieder auf den Schlüssel-Server hochladen.

Ist die Nachricht (mit dem eigenen öffentlichen Schlüssel) verschlüsselt, wird sie von *Thunderbird* mit dem eigenen privaten Schlüssel aus dem Schlüsselbund entschlüsselt und der entschlüsselte Text wird angezeigt.

Mehr dazu unter http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Einstieg

3.5.2.3 Text-Mails statt HTML-Mails

Bei Verwendung von Inline-PGP im Gegensatz zu PGP/MIME (siehe Kap. 3.5.1) kann es Probleme mit HTML-Mails geben, die dazu führen können, dass das eine verschlüsselte Mail nicht mehr entschlüsselt werden kann.

Deshalb empfiehlt es sich für das Verfassen von Mails das Text-Format einzustellen. In den Einstellungen jedes Mail-Kontos gibt es im Konfigurationsordner *Verfassen und Adressieren* die Option „*Nachrichten im HTML-Format verfassen*“. Diese Option sollte standardmäßig abgeschaltet sein.

Damit verzichtet man beim Verfassen auf Text-Formatierungen wie Überschriften, Fettschrift, Kursivschrift, Aufzählungslisten und unterschiedliche Schriftarten.

Stellt man von Inline-PGP wieder auf PGP/MIME um, kann man für das Verfassen der Mails auch das HTML-Format verwenden.

3.5.2.4 Die verschlüsselt versendeten Mails lesbar machen

Mail verschlüsselt und versandt. So weit, so gut. Allerdings kann man die verschlüsselt versendete Mail nun selbst nicht mehr lesen. Dieses Problem lässt sich mit einer weiteren *Enigmail*-Einstellung beheben.

Diese Einstellung ist in *Thunderbird* zu finden unter *OpenPGP* → *Einstellungen* → *Senden*. Dort ist das Häkchen mit der Beschriftung „*Zusätzlich mit eigenem Schlüssel verschlüsseln*“ auszuwählen.

Wird diese Option nicht gesetzt, wird die Mail nur mit dem öffentlichen Schlüssel des Empfängers der Mail verschlüsselt. Dies hat zur Folge, dass auch die Mail-Kopie im Ordner *Gesendet* mit dem öffentlichen Empfänger-Schlüssel verschlüsselt wird. Da diese jedoch nur mit dem privaten Schlüssel des Empfängers entschlüsselt werden kann, kann man die Mails, die man verschlüsselt versandt hat, selbst nicht mehr entschlüsseln und lesen.

Setzt man jedoch das Häkchen, wird die Mail-Kopie, die im eigenen *Gesendet*-Ordner gespeichert wird, mit dem eigenen öffentlichen Schlüssel verschlüsselt und kann auch mit dem eigenen privaten Schlüssel wieder entschlüsselt werden. Diese Option bewirkt also, dass man die Kopien der verschlüsselt versendeten Mails im Ordner *Gesendet* auch nachträglich noch lesen kann.

3.5.2.5 Mit signierten Mails beginnen ...

Hat man PGP gerade erst eingerichtet, hat man in der Regel nur wenige Kommunikationspartner, mit denen man verschlüsselte Nachrichten austauschen kann. Um verschlüsselte Mails zu versenden, benötigt man ja die öffentlichen Schlüssel der Kommunikationspartner.

Beim Signieren der Mails ist man nicht auf den Kommunikationspartner angewiesen. Man benötigt dazu nur den eigenen privaten Schlüssel. Deshalb kann man, wenn man alles eingerichtet hat, damit beginnen alle ausgehenden Mails zu signieren.

Der jeweilige Partner kann die Mails, auch wenn sie signiert sind, immer lesen. Will er allerdings die Signaturen empfangener Mails überprüfen, muss auch er sowohl *Gpg4win* oder *GPG Keychain Access* als auch *Thunderbird/Enigmail* oder einen anderen Mail-Client mit PGP-Unterstützung installieren.

Durch das Versenden signierter Mails informiert man seine Kommunikationspartner, dass das eigene System PGP beherrscht. Man teilt mit, dass man die Voraussetzungen für die PGP-Nachrichtenverschlüsselung geschaffen hat und bewirbt damit die Verwendung von PGP.

Im Laufe der Zeit sammelt man immer mehr öffentliche Schlüssel von Kommunikationspartnern ein. Man kann diese Schlüssel signieren/beglaubigen, die Vertrauensstufe auf ein adäquates Level einstellen und so allmählich den Anteil der verschlüsselten Kommunikation erhöhen.

3.5.2.6 Empfängerregeln

Beginnt man gerade mit der Verwendung von PGP, hat man in der Regel nur die öffentlichen Schlüssel weniger Kommunikationspartner im Schlüsselbund. Deshalb ist es, wie oben beschrieben, sinnvoll, Mails beim Versand standardmäßig zu signieren, jedoch nicht zu verschlüsseln.

Für die wenigen Empfänger, deren öffentlicher Schlüssel sich im eigenen Schlüsselbund befindet, lässt sich jedoch pro Empfänger eine von der Standardeinstellung abweichende Empfängerregel erstellen.

Dazu wählt man in *Thunderbird/Enigmail* den Menüpunkt *OpenPGP* → *Empfängerregeln...* Darauf öffnet sich ein Dialog mit einer Liste von Empfängerregeln, die anfangs noch leer ist. Man klickt nun auf den Button *Hinzufügen*. Ein weiterer Dialog öffnet sich. In diesem gibt man eine bestimmte Empfänger-Email-Adresse ein und kann für diese eine Regel erstellen: Man wählt den zu verwendenden öffentlichen Schlüssel und legt mit weiteren Optionen fest, ob

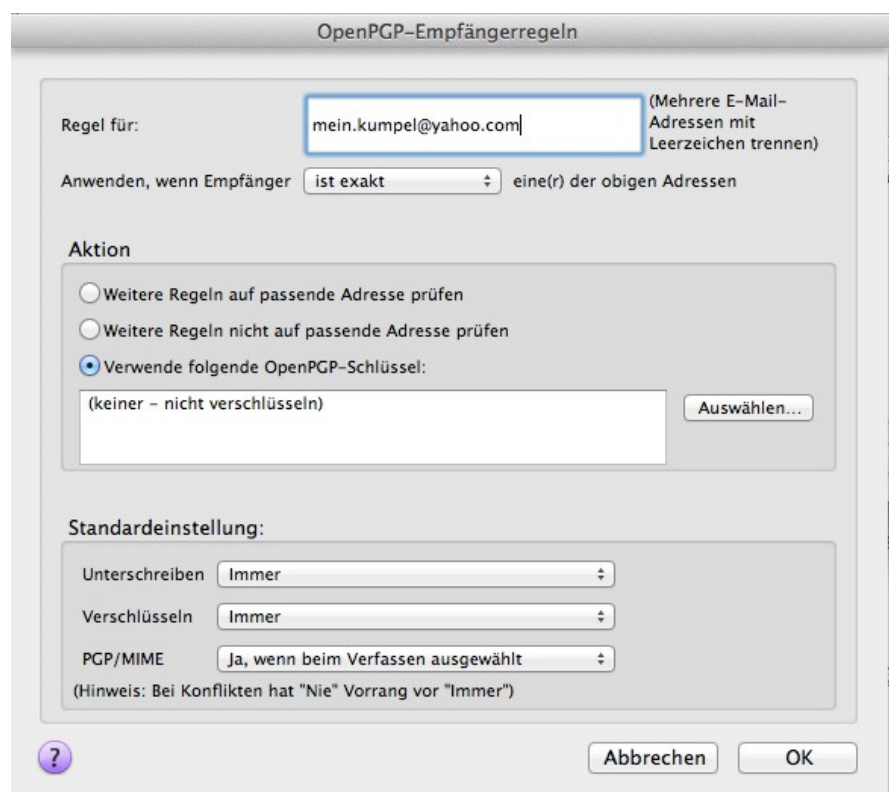


Abbildung 27: Neue Empfängerregel erstellen

die Mails an diesen Adressaten signiert werden sollen, ob sie verschlüsselt werden sollen und ob PGP/MIME zu verwenden ist. Der Dialog ist selbsterklärend.

3.6 *Fallstricke beim Einsatz von GnuPG*

Diese sind das Thema eines kurzen, jedoch sehr lesenswerten Online-Artikels auf der Web-Site von Heise und für jedermann erreichbar unter:

<http://www.heise.de/ix/heft/Im-zweiten-Anlauf-2197613.html> .

In diesem Artikel wird übrigens die Verwendung von PGP/MIME empfohlen (siehe Kap. 3.5.1). Das werde auch ich tun, sobald dieses Format auch für mein Android-Smartphone und -Tablet verfügbar ist.

3.7 *Webmail? Vergiss es!*

Per Webmail – also mit dem Browser auf den Mail-Account zuzugreifen – ist natürlich eine bequeme Sache. Man kann es an jedem Rechner mit Internetzugang tun und muss dazu nicht zu Hause vor dem eigenen System sitzen.

Doch Webmail und signierte und verschlüsselte Kommunikation – das beißt sich.

Theoretisch wäre das denkbar ... nämlich dann, wenn man den privaten Schlüssel seinem Mail-Provider anvertraut. Manche Mail-Provider bieten das sogar an. Doch widerspricht das dem elementaren Grundsatz asynchroner Verschlüsselung: **Gib den privaten Schlüssel niemals aus der Hand!** Der Fremde, der meinen Schlüssel missbrauchen wollte (und über etwas technisches Know-how und die geeigneten Tools verfügt) ...

- Er könnte Mails in meinen Namen signieren und versenden.
- Er könnte an mich gerichtete, verschlüsselte Mails abfangen und entschlüsseln.
- Er könnte in meinem Namen andere Schlüssel unterschreiben/beglaubigen.
- Er könnte den Schlüssel widerrufen. Dadurch würde er ungültig. Und ich könnte meinen Schlüssel nicht mehr benutzen.

Der private Schlüssel beim Provider ... der Provider könnte ihn missbrauchen, aber ein Hacker oder die NSA, die beim Provider einbricht und meinen Schlüssel stiehlt, könnte ihn genau so missbrauchen.

Also muss der eigene private Schlüssel möglichst gut geschützt (durch eine starke Passphrase) im eigenen Schlüsselbund auf dem eigenen Rechner eingesperrt bleiben.

Wer über Webmail auf den eigenen Mail-Account zugreift, kann heute nur unverschlüsselt kommunizieren:

- Er kann verschlüsselte Mails nicht entschlüsseln und lesen.
- Er kann keine Signaturprüfung bei eingegangenen Mails vornehmen.
- Er kann keine signierten Mails versenden.
- Und er kann auch keine verschlüsselten Mails versenden.

Ist man zu Hause, kann man den mit PGP verschlüsselten und signierten Mailverkehr mit Thunderbird abwickeln. Um unterwegs nicht auf den verschlüsselten Mailverkehr verzichten zu müssen, kann man sich auf dem Smartphone oder Tablet einen Mail-Client mit PGP-Unterstützung

einrichten (siehe Kap. 3.9 und Kap. 3.10). Dann ist man für den Mailzugriff nicht auf fremde Rechner angewiesen. Alternativ (und sicher weniger komfortabel) lässt sich auch *Thunderbird portable* mit *Enigmail* auf einen USB-Stick installieren. Diesen USB-Stick kann man leicht mitnehmen und an jedem fremden Windows-Rechner anschließen und betreiben (siehe nachfolgendes Kap. 3.7.1).

In einigen Monaten könnte Webmail mit PGP doch in den Bereich des Möglichen rücken. Google arbeitet an einer Chrome-Erweiterung, die die PGP-Verschlüsselung auch mit dem Chrome-Browser ermöglichen soll. Diese Erweiterung soll auch auf den PGP-Schlüsselbund des aktuellen Rechners zugreifen können.

Man wird diese PGP-Erweiterung für Chrome also nur auf den Rechnern nutzen können, auf denen auch eine Kopie des Schlüsselbundes liegt. Den Vorteil von Webmail, dass sie auf jedem beliebigen Rechner nutzbar ist, hat man mit der Erweiterung allein also noch nicht wiedergewonnen. Um die Erweiterung nicht nur am eigenen sondern auch an fremden Rechnern nutzen können, müsste man eine Kopie des Schlüsselbundes auch immer auf einem USB-Stick oder auf einer Speicherkarte mit sich herumtragen.

An dieser Stelle will ich nochmals die generelle Webmail-Warnung aussprechen. Fremde Rechner hat man meist nicht unter der eigenen Kontrolle. Das Risiko, an einem verseuchten Rechner zu sitzen, ist in der Regel zu hoch, um dort sicherheitskritische Tätigkeiten wie den Zugriff auf den Webmail-Account oder den Bank-Account durchzuführen (siehe auch Kap. 2.7).

3.7.1 Die Webmail-Alternative – *Thunderbird portable*

Will man unterwegs unbedingt Zugriff auf die verschlüsselten Mails und hat kein entsprechend eingerichtetes Gerät (Laptop, Tablet oder Smartphone) mit Internetzugang dabei, dann gibt es noch eine Alternative.

Man installiert *Thunderbird portable*, *Enigmail* und *Gpg4win* auf einen USB-Stick und kann diese Installation mit dem eigenen Schlüsselbund überall hin mitnehmen. An jedem fremden Windows-PC lässt sich der Stick anschließen und betreiben.

Ich will diese Alternative (die ich mehr der Vollständigkeit wegen aufgenommen habe) hier nicht vertiefen, sondern verweise zum Download und zur weiteren Information auf die nachstehenden Links.

Thunderbird portable Download:

<http://www.heise.de/download/thunderbird-portable.html>

Infos zu *Thunderbird portable*: http://www.thunderbird-mail.de/wiki/Portable_Thunderbird

3.8 Verschlüsselte Mails auf dem Zweitrechner

Besitzt man einen zweiten Rechner, auf dem man ebenfalls auf die verschlüsselte Email-Kommunikation zugreifen will, so muss man die gesamte beschriebene Einrichtungsprozedur auf diesem System wieder holen.

Außerdem macht das nur Sinn, wenn man die Mail-Konten auf dem Erstrechner bereits mit IMAP

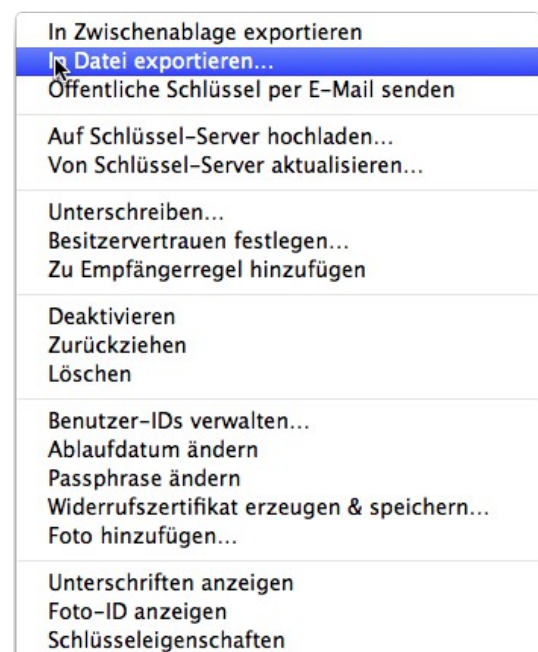
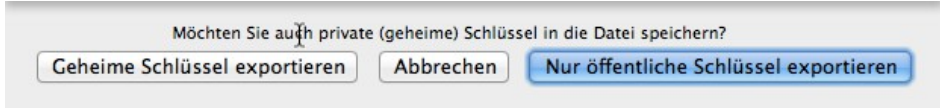


Abbildung 28: Enigmail: Schlüssel-Export

konfiguriert hat. Dabei bleiben die Mails zentral beim Provider gespeichert. Es können beliebig viele eigene Geräte auch Smartphones und Tablets für den IMAP-Zugriff auf den Mail-Account eingerichtet werden. Auch auf dem Zweitrechner darf der Mail-Abruf nicht mit POP, sondern muss mit IMAP konfiguriert werden.

Auf dem Zweitrechner erzeugt man keine neuen Schlüssel, sondern man muss den gesamten Schlüsselbund des Erstrechners auf den zweiten übertragen. Entscheidend ist die Übertragung der eigenen privaten Schlüssel. Die öffentlichen fremden Schlüssel könnte man sich auch von einem Schlüssel-Server holen. Einfacher ist es sicherlich, gleich alle Schlüssel in einem Aufwasch zu übertragen.

Dabei kann man so vorgehen:

- Die *Enigmail*-Schlüsselverwaltung öffnen
 - Die eigenen Schlüsselpaare (Schlüssel-Typ: privat + öffentlich) markieren. Dann im Kontextmenü den Eintrag *In Datei exportieren ...* auswählen. In dem folgenden Fenster angeben, dass man auch die privaten, geheimen Schlüssel exportieren will. Schließlich in eine Datei speichern.
- 
- Abbildung 29: Enigmail: Auch die geheimen Schlüssel exportieren?*
- Die fremden Schlüssel (Schlüssel-Typ: öffentlich) markieren. Dann im Kontextmenü den Eintrag *In Datei exportieren ...* auswählen. Schließlich in eine Datei speichern.
 - Beide Dateien auf den anderen Rechner übertragen. Wichtig ist dabei, dass die Übertragung über einen **sicheren Übertragungskanal** erfolgt.
 - Die Übertragung über das Internet (in einer unverschlüsselten Mail an sich selbst oder über unverschlüsselten Cloud-Speicher (z.B. Dropbox) verbietet sich von selbst. Die Übertragung über verschlüsselten Cloud-Speicher kommt durchaus in Frage. (Dies muss man allerdings zuvor eingerichtet haben. Z.B. mit dem Tool *BoxCryptor* einen Dropbox-Ordner verschlüsseln und trotzdem mit allen angemeldeten Geräten auf den verschlüsselten Speicher zugreifen. Die verschlüsselten Dateien liegen in der Cloud und werden nur lokal auf den Geräten entschlüsselt.)
 - Die Übertragung durch das eigene Heimnetz (LAN/WLAN) ist weit weniger kritisch, allerdings auch nicht ganz risikolos. (Man kann nie ganz sicher sein, welche ungebetenen Gäste unbemerkt im eigenen Netz herumturnen. Gerade nach den jüngsten Skandalen um unsichere Router ist diese Gefahr nicht unrealistisch.)
 - Ist die Übertragung über das Netzwerk unvermeidlich, so kann man die Schlüssel vorher in eine Passwort-geschützte Zip-Datei verpacken und dann auch über ein unsicheres Netzwerk übertragen. Selbstverständlich ist dazu ein starkes Passwort zu wählen.
 - Sehr sicher ist die Übertragung mittels eines externen Datenträgers (USB-Stick, Speicherkarte, externe Festplatte). Diese Methode ist außerdem recht einfach und deshalb in der Regel zu bevorzugen.
 - Nach der Übertragung kann man die übertragenen Schlüssel aus den beiden Dateien (eigene private Schlüssel und fremde öffentliche Schlüssel) in die Schlüsselverwaltung des Zweitrechners importieren. Nach dem erfolgreichen Import kann und sollte man die Dateien

mit den exportierten Schlüsseln auf beiden Rechnern und auf dem Datenträger löschen.

- Erst wenn die Schlüssel im Schlüsselbund vorliegen, kann man auf dem Zweitrechner die Zuordnung der privaten Schlüssel zu den eingerichteten Mail-Konten vornehmen.

3.9 Verschlüsselte Mails auf iPhone oder iPad

Da ich selbst kein iPhone- oder iPad-Nutzer bin, kann ich aus Erfahrung zur Email-Verschlüsselung mit PGP unter iOS nichts sagen.

Ich erlaube mir an dieser Stelle ausnahmsweise, einen Absatz aus dem Artikel „*Verschlüsseln und Signieren mit PGP*“ aus dem c't-Sonderheft „*Sichere E-Mail*“ (siehe Kap. 1.6) zu zitieren. Der zitierte Abschnitt findet sich auf Seite 91 im Artikel: „Verschlüsseln und Signieren mit PGP“.

Für iOS existiert nur eine halbwegs praktikable PGP-App, die für Privatanwender in Frage kommt. **iPGMail** ist mangels Schnittstellen in iOS nicht in die bordeigene Mail-App integriert. Möchte man Mails dechiffrieren, heißt es, sie per Cut & Paste in iPGMail zu importieren. Möchte man selbst Mails verschlüsseln, klappt das direkt in iPGMail. Geheime Schlüssel und öffentliche Schlüsselbunde lassen sich via USB und iTunes importieren. Vorsicht: Einmal entschlüsselte Nachrichten speichert die App im Klartext, außerdem bietet sie noch die Möglichkeit, Texte und Schlüssel beispielsweise in die Cloud zu laden. Selbstredend, dass sie dort nichts verloren haben.

Das c't-Sonderheft mit dem Artikel erschien Anfang 2014. Möglicherweise wird diese Lösung in den nächsten Monaten weiterentwickelt, sodass auch unter iOS bald ein praktikables Verfahren zur Verwendung von PGP zur Verfügung steht.

3.10 Verschlüsselte Mails auf Android-Smartphone oder -Tablet

Um PGP auch auf dem Android-Smartphone oder -Tablet für Mail-Empfang und -Versand zu nutzen, benötigt man auch hier die richtigen Tools (eine Mail-App und eine App zur Schlüsselbund-Verwaltung). Nach der Installation der Apps muss man die Schlüssel aus dem Schlüsselbund des PC auf das Gerät übertragen und in die Schlüsselbund-Verwaltung importieren.

3.10.1 Die passenden Android-Apps

Für die Schlüsselverwaltung kommen aktuell zwei Apps in Frage:

- **APG**: OpenPGP-Unterstützung für Android:
<https://play.google.com/store/apps/details?id=org.thialfihar.android.apg>
- **OpenKeyChain**: Eine Weiterentwicklung von APG. Die neuen Features (Schlüsseltausch via NFC und QR-Code) dieser App werden wieder in APG zurückportiert:
<https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain>

Im Google Play Store gibt es nicht wenige Mail-Apps. Sucht man darunter aber eine mit Unterstützung für PGP-Verschlüsselung, wird die Auswahl recht übersichtlich. Es stehen nur *K-9 Mail* und dessen Abkömmlinge zur Auswahl:

- **K-9 Mail**: <https://play.google.com/store/apps/details?id=com.fsck.k9>
- **K-@ Mail**: <https://play.google.com/store/apps/details?id=com.onegravity.k10.free>
- **K-@ Mail Pro**: <https://play.google.com/store/apps/details?id=com.onegravity.k10.pro2>
- **Kaiten Mail**: <https://play.google.com/store/apps/details?id=com.kaitenmail.adsupported>
- **Kaiten Mail (kommerziell)**: <https://play.google.com/store/apps/details?id=com.kaitenmail>

Bei diesen Mail-Apps ist aktuell (Juli 2014) noch keine stabile Unterstützung des moderneren Verschlüsselungsformates PGP/MIME gegeben. Die Entwickler arbeiten aber daran, sodass die Situation in einem halben Jahr möglicherweise besser aussieht.

Auf meinen Android-Geräten sind *APG* und *K-@ Mail Pro* installiert. Ich beziehe mich in der folgenden Beschreibung auf diese beiden Apps. Die Benutzung der anderen Apps dürfte sich davon nur geringfügig unterscheiden.

3.10.2 Installation der Apps

Die Apps installiert man wie üblich aus dem Google Play Store. Zweckmäßigerweise installiert man die Schlüsselbund-Verwaltungs-App (bei mir *APG*) vor der Mail-App (bei mir *K-@ Mail Pro*). Die Mail-App findet dann die Schlüsselbund-Verwaltung auf dem Android-Gerät vor und trägt diese bei der Installation als zuständige Kryptographie-App in die eigene Konfiguration ein.

Danach richtet man den/die Account/Accounts in der Mail-App ein. Die Sicherheitseinstellungen sind analog zu den *Thunderbird*-Einstellungen (siehe Kap. 2.4.1) vorzunehmen.

Sendet man jetzt auf dem PC eine verschlüsselte Mail an sich selbst und versucht sie auf dem Smartphone bzw. Tablet zu lesen, so kann diese jetzt noch nicht entschlüsselt werden (siehe Abb. 30).

3.10.3 Übertragung der Schlüssel

Bevor sie in die Schlüsselbund-Verwaltung importiert werden, müssen die Schlüssel vom PC auf das Android-Gerät übertragen werden. Dabei ist grundsätzlich so vorzugehen wie bei der Übertragung der Schlüssel auf den Zweitrechner (siehe Kap. 3.8). Auch hier nochmals die Warnung, die Schlüssel niemals über einen unverschlüsselten Netzwerkkanal zu übertragen!

Für Android-Geräte bietet sich zusätzlich zu den in Kapitel 3.8 genannten Möglichkeiten ein weiterer, sehr einfacher Übertragungsweg an. Man kann das Gerät direkt an die USB-Schnittstelle des Rechners anschließen und dann vom Rechner auf das Android-Dateisystem zugreifen. Das Smartphone oder Tablet muss dazu als Mediengerät (nicht als Kamera) angeschlossen sein. Die Art des Anschlusses lässt sich in den USB-Einstellungen des Geräts einsehen oder ändern.

Unter Windows geht die Übertragung ohne weitere Vorbereitungen. Nach dem Anschluss mit einem USB-Kabel an den Rechner, kann man mit dem Windows-Explorer die exportierten Schlüssel direkt in einen Ordner des Android-Geräts (z.B. den Download-Ordner) übertragen.

Auf dem Mac muss zunächst eine Dateiübertragungssoftware von

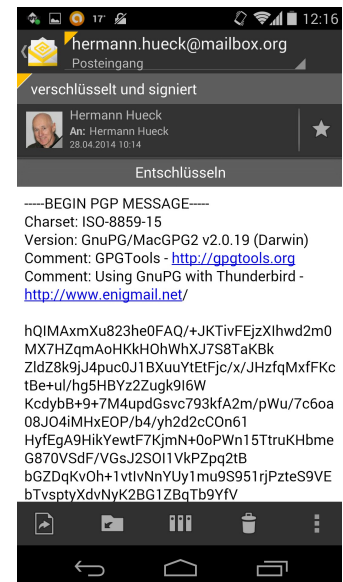


Abbildung 30: Android: *K-@ Mail*: Nur Zeichensalat – Die verschlüsselte Mail ist nicht lesbar.

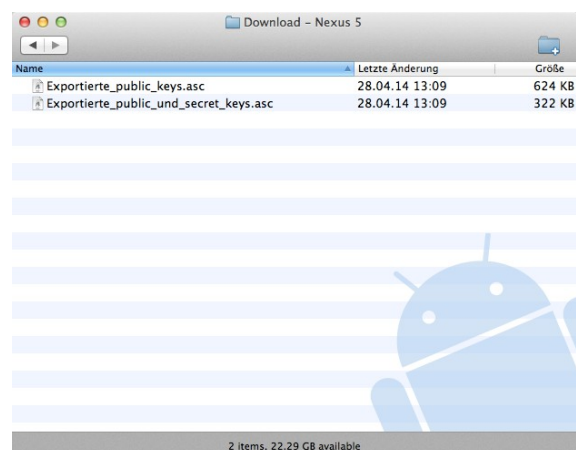


Abbildung 31: Android-Filetransfer auf dem Mac, Smartphone über USB angeschlossen: Die Schlüssel werden ins Download-Verzeichnis kopiert.

<http://www.android.com/filetransfer/> heruntergeladen und installiert werden. Man schließt das Gerät an, startet die Filetransfer-Anwendung und öffnet darin den Download-Ordner. Man zieht die exportierten Schlüssel-Dateien in das Anwendungsfenster; diese werden dabei in den Download-Ordner kopiert.

3.10.4 Schlüssel-Import

Nun ist der Schlüsselbund zu öffnen; d.h. die App APG wird gestartet. Man wählt *Schlüssel importieren* → *Datei* und wählt dann den Ordner aus, in dem die Schlüssel-Datei(en) liegen. APG zeigt die importierbaren Schlüssel an. Die Schlüssel, die in den Schlüsselbund importiert werden sollen, können ausgewählt werden. Beim Tippen auf den Button „*Ausgewählte Schlüssel importieren*“ werden die betreffenden Schlüssel in den Schlüsselbund importiert. Bei mehreren Schlüssel-Dateien ist der Vorgang entsprechend zu wiederholen.

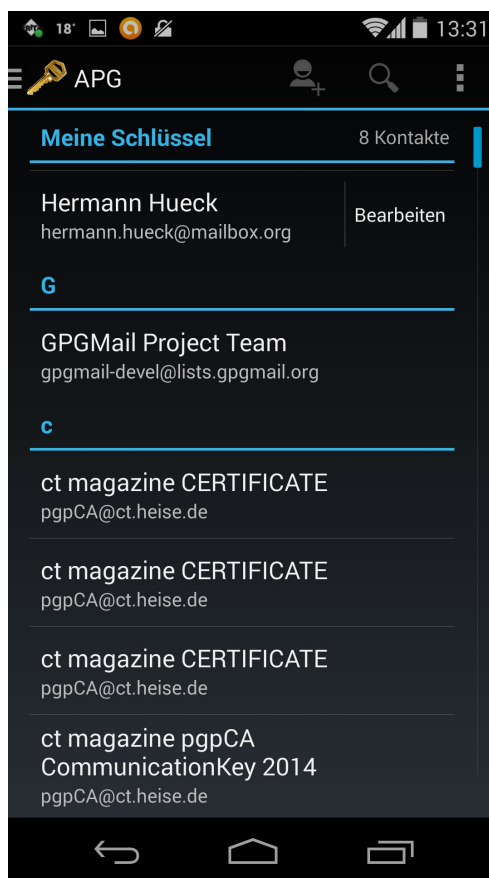


Abbildung 32: Android: Die importierten Schlüssel im APG-Schlüsselbund

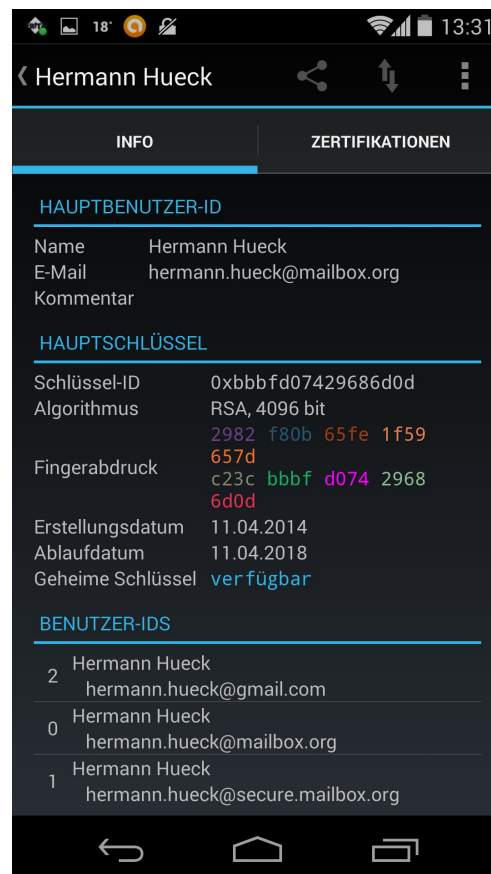


Abbildung 33: Android: APG-Schlüsseldetails: Ein Schlüssel mit drei Benutzer-IDs

Nach dem erfolgreichen Import werden die Schlüsseldateien auf dem Gerät nicht mehr benötigt. Insbesondere Dateien, die private Schlüssel enthalten, sind unbedingt zu löschen.

Öffentliche Schlüssel können auch von einem Key-Server importiert werden. In APG wählt man die Option *Schlüssel importieren* → *Schlüsselserver*. Danach wählt man den Schlüsselserver aus und gibt einen Suchbegriff für den Schlüssel ein, z.B. eine Email-Adresse oder einen Teil einer Email-

Adresse. Aus der Liste der gefundenen Schlüssel kann man wieder diejenigen markieren, die man importieren will. Beim Tippen auf den Button „*Ausgewählte Schlüssel importieren*“ werden die betreffenden öffentlichen Schlüssel in den Schlüsselbund importiert.

3.10.5 Konfiguration der Schlüsselservers in APG

In APG unter *Einstellungen* → *Allgemein* → *Schlüsselservers* können auch die Schlüssel-Server festgelegt werden, die beim Import oder Export öffentlicher Schlüssel verwendet werden sollen. Auf meinem Android-Smartphone und -Tablet sind (analog zu *Enigmail* auf dem Mac) folgende Schlüssel-Server definiert:

- a.keyserver.pki.scientia.net
- p80.pool.sks-keyservers.net
- pgp.mit.edu
- subkeys.pgp.net

3.10.6 PGP-Mail-Empfang und -Versand

3.10.6.1 Einstellungen

Die verschlüsselte Mail, die vor dem Schlüssel-Import noch nicht lesbar war, lässt sich nun entschlüsseln und lesen.

In der Mail-App gibt es zu jedem konfigurierten Account eine Kryptographie-Konfiguration mit drei möglichen Parametern. Diese ist pro Mail-Account folgendermaßen auf sinnvolle Werte einzustellen.

- OpenPGP Provider: *APG*.
- Automatisches Signieren abgehender Mails: Einschalten.
- Verschlüsselung abgehender Mails: Automatisch verschlüsseln, wenn ein öffentlicher Schlüssel des Empfängers vorhanden ist.

Eine weitere Einstellung betrifft das Verfassen der Mails. Die Android-Mail-Apps unterstützen bislang nicht das modernere Verschlüsselungsformat PGP/MIME. Man kann also nur Inline-PGP verwenden. Inline-PGP wiederum ist nicht mit Mails im HTML-Format verträglich. Analog zu *Thunderbird* (siehe Kap. 3.5.2.3) ist deshalb für das Verfassen von Mails das Text-Format einzustellen.

- In den Einstellungen der App *K-@ Mail* wählt man den Menüpunkt *Nachrichten senden* → *Formatierung*. Im Dialogfenster, das sich daraufhin öffnet, ist die Option „*Einfacher Text (keine Bilder und Formatierungen)*“ zu selektieren. Diese Einstellung ist für jedes Konto, das zum Versenden verschlüsselter Mails verwendet werden soll, vorzunehmen.

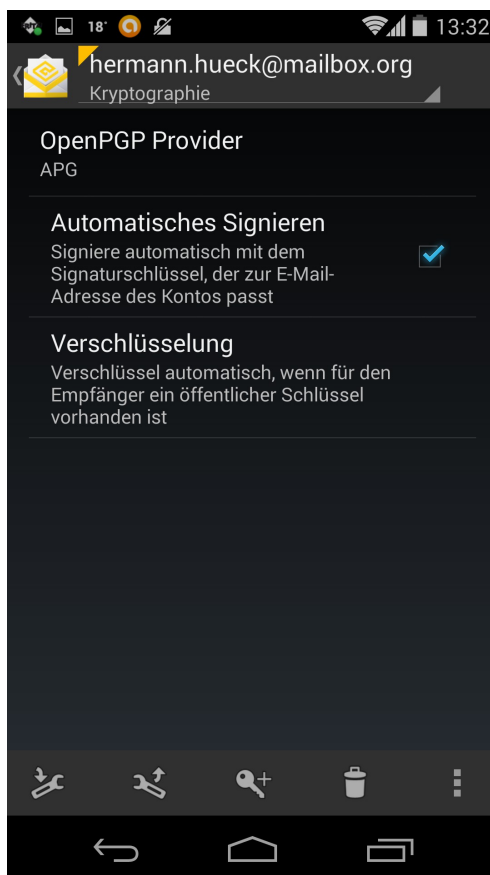


Abbildung 35: Android: K-@ Mail: Kryptographie-Optionen

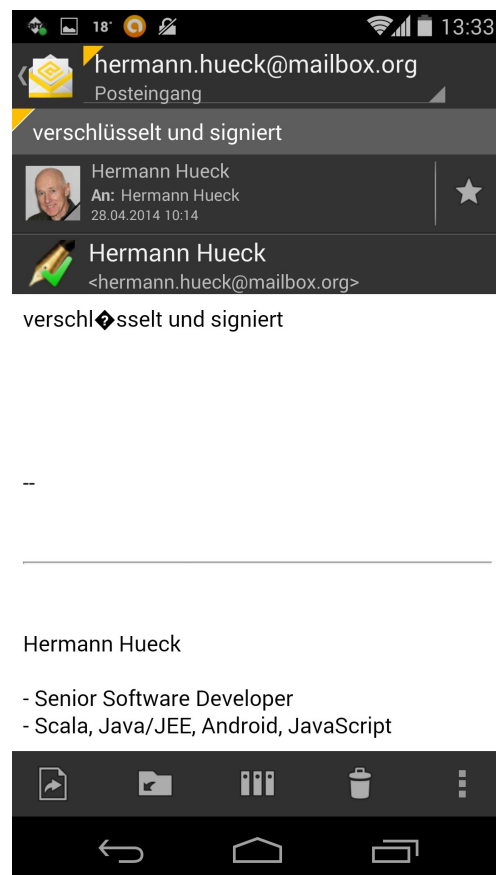


Abbildung 34: Android: K-@ Mail: Die Mail kann jetzt gelesen werden.

3.10.6.2 Versand

Das Senden einer Mail funktioniert wie üblich.

Vor dem Versenden der Mail kann man jeweils mit einer Checkbox festlegen, ob die aktuelle Mail signiert und verschlüsselt werden soll. Mit den genannten Voreinstellungen muss man hier meist nichts mehr ändern.

Außerdem muss man auswählen, mit welchen öffentlichen Schlüsseln die Mail verschlüsselt werden soll. Man wählt die Schlüssel des bzw. der Adressaten der Mail und – **WICHTIG !!!** - auch den **eigenen Schlüssel**. So wird die Mail, die im Ordner *Gesendet* gespeichert wird, mit dem eigenen öffentlichen Schlüssel verschlüsselt und kann nachträglich auch mit dem eigenen privaten Schlüssel wieder entschlüsselt werden.

Das Verschlüsselungsformat PGP/MIME wird aktuell von *APG* und *OpenKeyChain* noch nicht unterstützt. Man kann seine Verwendung auf dem Android-Gerät bislang noch nicht einstellen. Dieser Umstand hat die Konsequenz, dass man auch auf dem PC mit *Thunderbird/Enigmail* auf PGP/MIME verzichten muss, falls die Mails sowohl auf dem Rechner als auch auf dem Android-Gerät lesbar sein soll.

Sendet man eine mit PGP/MIME verschlüsselte Mail, wird auch eine verschlüsselte Kopie der eigenen Mail im *Gesendet*-Ordner gespeichert. Diese wird mit dem eigenen öffentlichen Schlüssel

verschlüsselt und kann grundsätzlich mit dem eigenen privaten Schlüssel auch dechiffriert werden. Mit *Thunderbird* könnte man diese Mail wieder entschlüsseln und lesen. Auf dem Android-Tablet oder -Smartphone wäre die PGP/MIME formatierte Mail nicht dechiffrierbar und auch nicht lesbar, da die Unterstützung für dieses Format bislang fehlt.

3.10.7 Empfang

Wird eine verschlüsselte Mail mit *K-@ Mail* auf dem Android-Gerät empfangen, ist sie zunächst noch nicht lesbar und sieht etwa so aus wie in Abbildung 30 dargestellt. Die App erkennt jedoch, dass es sich um eine verschlüsselte Mail handelt und zeigt über der Mail eine Schaltfläche mit der Aufschrift „*Entschlüsseln*“. Beim Tippen auf diese Schaltfläche verlangt die App die Eingabe der Passphrase. Gibt man diese richtig ein, bekommt die App den Zugriff auf den privaten Schlüssel und kann damit die Mail entschlüsseln und lesbar machen (siehe Abbildung 34).

3.11 Das verschlüsselte Postfach von *mailbox.org*

mailbox.org bietet an, Mails, die der Absender nicht verschlüsselt hat, sofort nach dem Eintreffen beim Provider zu verschlüsseln und verschlüsselt im Posteingang abzulegen. Ist die eingegangene Mail erst verschlüsselt gespeichert, kann auch das Team von *mailbox.org* oder ein Hacker, der in den Server von *mailbox.org* einbricht, die Mail nicht mehr entschlüsseln und lesen. Da nur der Empfänger den passenden privaten Schlüssel hat, kann nur er sie entschlüsseln und lesen.

Damit das klappt, muss man in den Einstellungen seines Accounts bei *mailbox.org* die Option „*PGP-Verschlüsselung für eingehende Mails aktivieren*“ und seinen öffentlichen Schlüssel in das darunter liegende Eingabefeld kopieren. Der Provider verschlüsselt dann alle eingehenden Mails, die der Absender nicht bereits verschlüsselt hat.

Er verwendet dabei das modernere Format PGP/MIME. In *Thunderbird* kann ich diese Mails entschlüsseln und lesen. Auf meinem Android-Smartphone oder -Tablet sind die Mails allerdings nicht mehr lesbar, da die Android-Mail-Apps dieses Format bislang noch nicht unterstützen. Da ich die Mails auch auf meinen Android-Geräten lesen will, habe ich das verschlüsselte Postfach wieder deaktiviert.

Nutzt man das verschlüsselte Postfach, kann man nicht mehr feststellen, die Mail mit meinem öffentlichen Schlüssel verschlüsselt hat, der Absender der Mail oder der Provider *mailbox.org*. Möglich wird die Unterscheidung, wenn man ein zweites Schlüsselpaar erzeugt (siehe Kap. 3.4.2). Bei der Eingabe der Benutzer-ID für dieses Schlüsselpaar kann man das Email-Feld leer lassen. Den zweiten öffentlichen Schlüssel verwendet man nur für das verschlüsselte Postfach von *mailbox.org*. Man exportiert diesen Schlüssel nicht auf einen Key-Server.

Nun kann der Provider *mailbox.org* mit meinem zweiten öffentlichen Schlüssel die Mails verschlüsseln. Die Kommunikationspartner, die mir verschlüsselte Mails senden, verwenden meinen ersten öffentlichen Schlüssel, der weltweit auf den Key-Servern verfügbar ist. Durch die Verwendung unterschiedlicher Schlüssel kann ich bei einer eingehenden Mail unterscheiden, ob sie vom Absender oder von meinem Provider verschlüsselt wurde.

Eines darf man dabei nicht vergessen: Nutzt man das verschlüsselte Postfach, dann sind diese Mails über Webmail nicht mehr lesbar, da der Browser sie nicht entschlüsseln kann.

(Beschreibung in Kapitel 3.11 und unter <https://mailbox.org/ihr-e-mail-postfach/#Datenschutz>)

3.12 Links zu diesem Kapitel

- Deutsche Wikipedia-Einträge zu S/MIME, PGP und OpenPGP:
<http://de.wikipedia.org/wiki/S/MIME>
<http://de.wikipedia.org/wiki/OpenPGP>
http://de.wikipedia.org/wiki/Pretty_Good_Privacy
- Stift-Film über asymmetrische Verschlüsselung und Signierung mit PGP:
<https://mailbox.org/stiftfilm-wie-funktioniert-e-mail-verschluesselung-mit-pgp/>
- Kurze Einführung in PGP bei WEB.DE: <https://hilfe.web.de/sicherheit/pgp.html> .
- Gpg4win, Download unter <http://www.gpg4win.de/> oder unter <http://www.heise.de/download/gpg4win-e2d914fd06f82399ae36052e8362b95c-1398246471-2619466.html>
- GPG Keychain Access von GPGTools; Download unter <http://www.heise.de/download/gpg-keychain-access-1178953.html>
- GPG Suite von GPGTools; Download unter <https://gpgtools.org/> oder bei Heise unter <http://www.heise.de/download/gpg-suite-a8929973af027b9846bd8eeb330da2d4-1398337947-2678714.html> .
- Manual-Seite für das gpg-Kommando:
<https://www.gnupg.org/documentation/manpage.html>
- Enigmail-Installationshilfe:
http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP#Enigmail_installieren
<https://www.verbraucher-sicher-online.de/anleitung/bildfolge-enigmail-installieren-in-mozilla-thunderbird>
- Detaillierte Beschreibung der Verschlüsselung und Schlüsselverwaltung mit Thunderbird/Enigmail:
http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP
- Detaillierte Beschreibung der Verschlüsselung und Schlüsselverwaltung mit Gpg4win:
<http://gpg4win.de/handbuecher/einsteiger.html>
- Detaillierte Beschreibung der Verschlüsselung und Schlüsselverwaltung mit GPG Keychain Access: <http://support.gpgtools.org/kb>
- Schlüsselgenerierung mit Thunderbird/Enigmail:
http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Schl%C3%BCsselverwaltung#Ein_Schl.C3.BCsselpaar_erzeugen
- Schlüsselgenerierung unter Windows mit Gpg4win:
http://gpg4win.de/handbuecher/einsteiger_7.html
- Schlüsselgenerierung unter Mac OS X mit GPG Keychain Access:
<http://support.gpgtools.org/kb/faq-gpg-keychain-access/generate-a-key>
- Krypto-Parties und Key-Signing-Parties:
<https://www.cryptoparty.in/>
<http://de.wikipedia.org/wiki/CryptoParty>

- c't-Kryptokampagne:
<http://www.heise.de/security/dienste/Krypto-Kampagne-2111.html>
- Schlüssel-Server-Empfehlungen bei Heise:
<http://www.heise.de/security/dienste/Keyserver-474468.html>
- OpenPGP-Sicherheit in *Thunderbird/Enigmail* konfigurieren:
http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Einstellungen#OpenPGP-Sicherheit
- Nachrichten-Empfang und -Versand mit OpenPGP:
http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Einstieg
- Fallstricke bei der Verwendung von GnuPG:
<http://www.heise.de/ix/heft/Im-zweiten-Anlauf-2197613.html> .
- *Thunderbird portable* Download:
<http://www.heise.de/download/thunderbird-portable.html>
- Infos zu *Thunderbird portable*:
http://www.thunderbird-mail.de/wiki/Portable_Thunderbird
- OpenPGP-Unterstützung für Android:
APG:
<https://play.google.com/store/apps/details?id=org.thialfihar.android.apg>
OpenKeyChain:
<https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain>
- Android Mail-Apps mit Unterstützung für PGP-Verschlüsselung:
K-9 Mail: <https://play.google.com/store/apps/details?id=com.fsck.k9>
K-@ Mail: <https://play.google.com/store/apps/details?id=com.onegravity.k10.free>
K-@ Mail Pro: <https://play.google.com/store/apps/details?id=com.onegravity.k10.pro2>
Kaiten Mail: <https://play.google.com/store/apps/details?id=com.kaitenmail.adsupported>
Kaiten Mail (kommerziell): <https://play.google.com/store/apps/details?id=com.kaitenmail>
- Dateiübertragung zwischen Mac OS X und Android:
<http://www.android.com/filetransfer/>
- Das verschlüsselte Postfach von *mailbox.org*:
<https://mailbox.org/ihr-e-mail-postfach/#Datenschutz>

4 Passwortsicherheit und Rechnersicherheit

Der Anhang behandelt einige allgemeine Sicherheitsfragen wie Passwortsicherheit und Rechnersicherheit, die nicht direkt mit der Email-Sicherheit zu tun haben aber dennoch die Sicherheit der Mails erhöhen. Selbstverständlich trägt ein sicherer Rechner auch zur Sicherheit der Mails bei.

Einige gute Tipps zur Rechner- und Passwortsicherheit findet man unter <https://www.verbraucher-sicher-online.de/computer-und-netze>

4.1 Sichere Passwörter

Ein paar Grundregeln für sichere Passwörter.

- Ein Passwort sollte mindestens 12 Zeichen lang sein.
- Ein Passwort sollte Buchstaben, Ziffern und Sonderzeichen enthalten. Manche Dienste lassen keine Sonderzeichen zu. Diesen Mangel kann man ausgleichen, indem man die Länge des Passwortes erhöht, z.B. auf 16 Zeichen.
- Ein Passwort sollte nicht leicht zu erraten sein. (keine Geburtstage, Spitznamen oder Ähnliches; also keine Eselsbrücken verwenden, die andere auch erraten können)
- Ein Passwort sollte nicht in Wörterbüchern vorkommen. (Wörterbuch-Attacken probieren einfach alle Wörter eines Wörterbuches aus. Ein Mensch würde Jahre dazu benötigen, ein leistungsfähiger Computer benötigt dazu nur Minuten und verwendet dabei auch noch die Wörterbücher mehrerer Sprachen.)
- Für verschiedene Dienste sind verschiedene Passwörter zu verwenden. Nicht selten verwenden Benutzer heute 50 und mehr Passwörter für verschiedene Dienste. Wird das Passwort eines Dienstes geknackt oder gestohlen, dann sind nicht 50 Dienste, sondern nur ein Dienst kompromittiert.
- Auch wenn es mühsam ist, Passwörter sollten regelmäßig geändert werden – bei sicherheitskritischen Accounts wie Email-Account oder Online-Banking-Account. Eine allgemeine Regel für das richtige Intervall der Passwortänderung gibt es nicht. Als Faustregel würde ich sagen, jedes halbe Jahr. Bei anderen Accounts, die man selten verwendet und deren Kompromittierung weniger schmerzt, kann das Intervall auch größer sein. Die Entscheidung für den richtigen Zeitabstand muss man letztendlich selbst treffen.

Hat man viele Passwörter, empfiehlt sich der Einsatz eines Passwort-Managers. Ein Passwort-Manager ist installierbares Programm, das Passwörter verwaltet. Die Passwortliste des Passwort-Managers sollte man mit einem starken Passwort, dem sog. Master-Passwort, schützen. Dieses sollte man sich jedoch gut einprägen, da es der einzige Zugang zu den anderen Passwörtern ist. Ein Passwort-Manager ist mit einem Schlüsselkasten vergleichbar, das Master-Passwort ist dabei der Schlüssel zum Schlüsselkasten.

4.2 Sicherheit von Rechner, Tablet und Smartphone

Um ein System sicher zu betreiben, muss man drei wesentliche Aspekte beachten:

- Das Betriebssystem aktuell halten
- Die installierten Programme aktuell halten (Das kann durchaus aufwändig sein.)

- Je nach Betriebssystem einen aktuellen Virens scanner installieren und die Viren-Signaturen aktuell halten, also mehrmals täglich aktualisieren (bzw. aktualisieren lassen).

Die nachfolgenden Unterkapitel beschreiben die Sicherheitsgrundsätze für die heute im Privatbereich gängigen Betriebssysteme.

4.2.1 Windows sicher betreiben

- **Betriebssystem:** In der *Systemsteuerung* → *Windows-Update* sind automatische Windows-Updates einzustellen. Der Rechner prüft dann (wenn er eine Verbindung ins Internet hat) selbstständig, ob bei Microsoft Updates vorliegen und aktualisiert sich. Werden Windows-Updates installiert, ist häufig ein Neustart des Systems erforderlich.
- **Programme:** Die installierten Programme müssen aktuell gehalten werden. Das ist bei Windows meist ein schwieriges Unterfangen, da jedes Programm von der Website seines Herstellers aktualisiert werden muss. Es gibt kein zentrales Software-Repository.
 - Besonderes Augenmerk sollte man auf die Aktualität der häufig verwendeten Kommunikationsprogramme werfen. Diese sind besonders sicherheitskritisch. Diese Programme sind auch auf den meisten Privat-PCs installiert: *Chrome*, *Firefox*, *Thunderbird* oder ein alternativer Email-Client. Auch *Adobe Flash Player*, *Adobe Reader* und *Java* sind besonders sicherheitskritisch und deshalb stets aktuell zu halten.
 - Alle installierten Programme regelmäßig auf Aktualität zu prüfen, kann sehr zeitaufwändig werden. Manche sagen, die Programme auf einem Windows-System aktuell zu halten, ist schlimmer als einen Sack Flöhe zu hüten. Ein guter Helfer dabei ist *Secunia PSI*. Dieses Programm führt Buch über alle installierten Programme und informiert den Benutzer, wenn eines oder mehrere Programme nicht mehr aktuell sind. Wöchentlich einen System-Scan durch Secunia durchführen zu lassen, kann eine sehr sinnvolle Maßnahme sein. (Siehe http://secunia.com/vulnerability_scanning/personal/)
- **Virens scanner:** Einen Windows-Rechner ohne Virens scanner zu betreiben oder die Viren-Signaturen nicht mehrmals täglich zu aktualisieren, darf als grobe Fahrlässigkeit eingestuft werden. Die Viren-Signaturen aktualisiert der installierte Virens scanner per Voreinstellung in der Regel automatisch.

4.2.2 Mac OS X sicher betreiben

- **Betriebssystem:** In den *Systemeinstellungen* → *App Store* sind automatische Updates einzustellen. Der Rechner prüft dann (wenn er eine Verbindung ins Internet hat) selbstständig, ob im Mac OS X App Store Updates vorliegen und aktualisiert sich. Auf diesem Wege wird sowohl das Betriebssystem als auch die aus dem App Store installierten Programme aktualisiert.
- **Programme:** Unter Mac OS X müssen nicht alle Programme aus dem OS X App Store installiert werden. Manche Programme stammen aus anderen Quellen. Diese erfahren keine automatische Aktualisierung. Sie müssen jeweils einzeln aktualisiert werden.
 - Wie unter Windows ist ein besonderes Augenmerk auf folgende häufig verwendete und sicherheitskritische Programme zu werfen: *Chrome*, *Firefox*, *Thunderbird* oder der alternative Email-Client, *Adobe Flash Player*, *Adobe Reader* und *Java*. Sie alle sind sehr beliebt, werden jedoch nicht aus dem OS X App Store installiert und damit auch nicht automatisch aktualisiert.

- Ein taugliches Software-Überwachungsprogramm wie *Secunia PSI* unter Windows gibt es für OS X meines Wissens nicht. Hier bleibt einem die manuelle Aktualitätsprüfung nicht erspart. Der Aufwand kann sich auf manchen Systemen erheblich reduzieren, wenn man alle Programme, die man nicht wirklich braucht, deinstalliert. So muss man diese auch nicht mehr aktuell halten.
- **Virens Scanner:** Früher konnte man einen Mac auch ohne Virens Scanner weitgehend sicher betreiben. Angriffe galten immer nur Windows-Rechnern und fast niemals den Macintosh-Rechnern. Durch die zunehmende Verbreitung (aktuell ca. 10 % der verkauften Privatrechner, Tendenz steigend) werden auch die Rechner von Apple zu einem immer beliebteren Angriffsziel. So ist es heute wie unter Windows durchaus auch auf dem Mac empfehlenswert, einen Virens Scanner zu installieren und die Viren-Signaturen mehrmals täglich zu aktualisieren bzw. aktualisieren zu lassen.

4.2.3 Linux sicher betreiben

- **Betriebssystem und Programme:** Für jede Linux-Distribution gibt ein zentrales Software-Repository im Netz, aus dem das Betriebssystem und die installierten Programme aktualisiert werden können. Mit einem lokal installierten Programm (unter Ubuntu Linux ist dies die „Softwareaktualisierung“) kann man das System aktuell halten. Mit wenigen Mausklicks aktualisiert man Betriebssystem und Programme. Wird dabei der Linux-Kernel aktualisiert, ist ein Neustart des Rechners erforderlich.
- **Virens Scanner:** Linux hat einen sehr geringen Marktanteil und ist durch die relativ geringe Verbreitung im Privatbereich für Virenhersteller kein lohnendes Angriffsziel. Da keine oder kaum Linux-Viren erstellt werden, ist die Herstellung von Linux-Virens Scannern ebenso wenig lukrativ. Die Hersteller von Virens Scannern bieten diese für Windows und meist für Mac OS X an. Es gibt jedoch keine Virens Scanner für Linux. Der Linux-Anwender muss zwar auf den Virens Scanner verzichten, hat trotzdem ein System, das durch Viren fast nicht gefährdet ist.

4.2.4 iOS sicher betreiben

iPhones und iPads beziehen (solange sie nicht durch einen Jailbreak entsperrt wurden) ihre Apps ausschließlich aus dem Apple App Store. Dieser wird von Apple stark kontrolliert und reguliert, was mancher als Nachteil empfinden mag. Ein Vorteil ist sicherlich, dass es sehr unwahrscheinlich ist, dass Schadprogramme in den App Store eingeschleust werden und sich lange dort halten können. Deshalb ist die Gefährdung von iOS-Geräten auch relativ gering. Die Installation eines Virens Scanners ist weder sinnvoll noch möglich. Mit regelmäßigen Aktualisierungen des iOS-Betriebssystems und der installierten Apps ist man weitgehend auf der sicheren Seite.

4.2.5 Android sicher betreiben

Bei Android ist die Bedrohungslage deutlich höher als bei iOS.

- **Betriebssystem:** Die Aktualisierungen des Betriebssystems kommen nicht wie bei iOS von einer zentralen Stelle für alle Android-Geräte, sondern von den Herstellern der jeweiligen Geräte. Diese sind allerdings meist sehr nachlässig mit der Versorgung der älteren Geräte mit Updates. So werden Sicherheitslücken häufig sehr spät oder gar nicht geschlossen. Als Nutzer hat man allerdings fast keine Möglichkeit, dieses Risiko zu verhindern.

Allerdings gibt es eine Alternative. Man kann das alternative Android-System *CyanogenMod* (<http://www.cyanogenmod.org/>) auf sein altes Gerät aufspielen. *CyanogenMod* gibt es für viele Geräte unterschiedlichster Hersteller. Die Chancen, ein altes Android-Gerät (z.B. das Samsung Galaxy S), das von seinem Hersteller längst nicht mehr mit Updates versorgt wird, wieder mit einer aktuellen Android-Version auszustatten, ist recht hoch.

- **Programme:** Der Google Play Store stellt Aktualisierungen nur für die Apps ein, nicht für das Android-Betriebssystem. Die Regulierung ist zwar nicht so streng wie bei Apple. Selbst wenn es einer Malware-App gelingt, durch die Kontrollen von Google durchzuschlüpfen, ist die Wahrscheinlichkeit der Entdeckung der Schadfunktionen doch recht hoch. Google entfernt diese zeitnah aus dem Play Store. Solange man seine Apps ausschließlich aus dem Google Play Store installiert und aktualisiert, bleibt das Malware-Risiko sehr überschaubar. Bezieht man die Apps auch aus alternativen App Stores, erhöht sich dieses Risiko deutlich.
- **Virens Scanner:** Man kann das Restrisiko durchaus noch verkleinern, indem man sich eine Virens Scanner-App auf das Android-Gerät installiert. Viele Hersteller von Virens Scannern für Windows bieten mittlerweile auch eine Virens Scanner-App für Android im Play Store an. Die Virens Scanner-Apps sind allerdings noch nicht so ausgereift und leistungsfähig wie ihre großen Schwestern unter Windows. Auch Experten sind diesbezüglich geteilter Meinung. Ich würde sagen, eine Virens Scanner-App auf dem Smartphone oder Tablet macht durchaus Sinn; einen sehr großen Sicherheitsgewinn sollte man sich aber nicht davon versprechen. Wichtiger ist der Verzicht auf alternative App Stores (s.o.).

4.3 Links zu diesem Kapitel

- Tipps zur Rechnersicherheit:
<https://www.verbraucher-sicher-online.de/computer-und-netze>
- Secunia PSI:
http://secunia.com/vulnerability_scanning/personal/
- *CyanogenMod*: Alternative Android-Firmware, verfügbar für sehr viele Geräte:
<http://www.cyanogenmod.org/>

5 Einige technische Erläuterungen

In diesem Kapitel liefere ich einige Hintergründe für technisch Interessierte. Sie sind in den vorigen Kapiteln nicht enthalten, da sie den Textfluss der vorigen Kapitel stören könnten und IT-Laien möglicherweise nicht interessieren oder gar abschrecken. Auch sind diese Kapitel zum Verständnis des Dokuments und zum Einrichten sicherer Mail nicht unbedingt erforderlich.

5.1 Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?

Dieses Kapitel vertieft die Inhalte, die in Kapitel 2.1.1 nur angerissen wurden.

Verschlüsselte Übertragungskanäle (oder Transport-Kanäle) bieten nur relative Sicherheit.

Werden die Daten, die auf einem verschlüsselten Transport-Kanal übertragen werden, mitgelesen, so bekommt der Mitlesende nur einen unverständlichen Kauderwelsch zu sehen. Wäre er im Besitz des Schlüssels, mit dem die Übertragung verschlüsselt wurde, könnte er den Datenkauderwelsch entschlüsseln und die übertragenen Inhalte im Klartext mitlesen.

Genau das versuchen sowohl Geheimdienste wie die NSA und GCHQ als auch Hacker. Sie versuchen die verschlüsselten Kanäle anzugreifen. Dies geht umso leichter, je schlechter die Verschlüsselung eines Transport-Kanals implementiert ist. Je besser die Verschlüsselung des Kanals „gemacht“ ist, umso schwieriger ist es, sie zu aufzubrechen. Bei der Mail-Übertragung ist es Sache der Provider, die Kanäle optimal zu verschlüsseln und damit die Hürden für die Kompromittierung des Kanals möglichst hoch zu setzen.

Ein mit SSL/TLS verschlüsselter Kanal wird schon beim Verbindungsaufbau durch eine sog. MITM-Attacke (Man in the Middle Attacke) angegriffen oder „gekapert“. Dabei klinkt sich der Angreifer schon beim Verbindungsaufbau in die TLS-Verbindung ein und kommuniziert dann verschlüsselt sowohl mit dem Client als auch mit dem Server. Der Client „denkt“, er kommuniziert mit dem Server und kommuniziert tatsächlich mit dem MITM. Der Server „denkt“ auch, er kommuniziert mit dem Client und kommuniziert tatsächlich mit dem MITM. Der MITM leitet die vom Client empfangenen Requests an den Server weiter und ebenso die vom Server empfangenen Responses zurück an den Client. Dabei kann der MITM die Requests und Responses mitlesen und ggf. auch ändern/manipulieren. Wichtig zum Kapern einer TLS-Verbindung ist ein gefälschtes Zertifikat.

Für „gut gemachte“ Verschlüsselung gibt es einige Qualitäts-Kriterien, an denen man auch die Mail-Provider messen kann (siehe Kap. 2.2). Bei Einhaltung dieser Qualitätskriterien ist das Kapern von TLS-Verbindung erheblich schwieriger.

- Verwendung der neuesten TLS-Version 1.2 (siehe Kap. 5.1.1)
- Unterstützung von PFS (siehe Kap. 5.1.2)
- Unterstützung von HSTS für die Webmail-Schnittstelle (siehe Kap. 5.1.3)
- Unterstützung von DANE (siehe Kap. 5.1.4)

Ich werde diese eher technischen Kriterien in den nachfolgenden Unterkapiteln auch hier nur oberflächlich skizzieren, um ein ganz grobes Verständnis zu ermöglichen.

Wer in die technischen Tiefen hinabsteigen will, den verweise ich wieder auf das c't-Sonderheft (siehe Kap. 1.6). Dort sind die betreffenden Informationen im Kasten auf Seite 25 unter dem Titel „Technische Eckpunkte für Server-Verschlüsselung“ und auf Seite 110 im Artikel „SSL-

Verbindungen besser sichern“ zu finden.

Eine weitere Informationsquelle über DANE (siehe Kap. 5.1.4) ist im c't Heft Nr. 11 des Jahres 2014 der Artikel auf Seite 194 mit dem Titel „Geleitschutz – DANE verbessert sicheren Transport zwischen Mailservern“. Dieser Artikel kann online auch separat bestellt werden. Man muss nicht das ganze Heft kaufen.

5.1.1 SSL (Secure Socket Layer) und TLS (Transport Layer Security)

SSL ist das alte Protokoll zur Verschlüsselung von Transport-Kanälen. Auch die neueste dritte Version des Protokolls (SSLv3) sollte nicht mehr verwendet werden. Der Nachfolger von SSL ist TLS. Die neueste Protokoll-Version ist die Version 1.2 (TLSv1.2). Nur diese Version sollte noch zum Einsatz kommen. Häufig spricht man von SSL-Transport-Verschlüsselung auch, wenn tatsächlich das neuere TLS zum Einsatz kommt.

SSL und TLS stellen für andere Protokolle einen verschlüsselten Übertragungskanal oder Transport-Kanal bereit.

Beispielsweise ist HTTPS (**HTTP Secure**) das durch den verschlüsselten Kanal übertragene HTTP-Protokoll. Das HTTP-Protokoll definiert das Format der Nachrichten zwischen einem Web-Browser (HTTP-Client) und einem Web-Server (HTTP-Server) (siehe Kap. 1.5.1.1). Bei HTTP werden die Daten über einen unverschlüsselten Transport-Kanal übertragen. Bei HTTPS erfolgt die Übertragung über einen mit SSL oder TLS verschlüsselten Kanal.

Ähnliches gilt für andere Protokolle. FTP (**F**ile **T**ransfer **P**rotokoll) ist das Protokoll zur unverschlüsselten Dateiübertragung (siehe Kap. 1.5.1.1). FTPS (**F**TP **S**ecure) verwendet eine SSL/TLS-verschlüsselte Transportverbindung für die Nachrichten, die bei der Dateiübertragung zwischen FTP-Client und FTP-Server ausgetauscht werden.

Nicht anders ist es bei den Protokollen für die Mail-Übertragung (siehe Kap. 1.5.1.1). SMTP und IMAP verzichten auf Verschlüsselung beim Datentransport, während SMTPS und IMAPS für den selben Zweck einen SSL/TLS-verschlüsselten Übertragungskanal verwenden.

SSL und TLS basieren auf Server-Zertifikaten, mit denen sich die Server vor dem Aufbau der verschlüsselten Transport-Verbindung bei Clients ausweisen können. Auch die neueste TLS-Version kann kompromittiert werden, wenn Zertifikate gefälscht werden. Dies ist nicht ganz einfach, aber doch möglich. Diesem Problem versucht, DANE beizukommen (siehe Kap. 5.1.4).

Weitere Informationen zu SSL und TLS unter:

http://de.wikipedia.org/wiki/Transport_Layer_Security

Die Protokolle HTTP, FTP, SMTP und IMAP sind im Kapitel 1.5.1.1 im Glossar kurz erläutert. Detailliertere Informationen finden sich auf den deutschen und englischen Wikipedia-Seiten.

5.1.2 PFS (Perfect Forward Secrecy)

Die Geheimdienste zeichnen auch verschlüsselte Kommunikation auf, die sie nicht dechiffrieren können, da ihnen der Schlüssel fehlt. Erhalten sie den Schlüssel allerdings zu einem späteren Zeitpunkt (z.B. durch den Einbruch in einen Server oder durch per Gerichtsbeschluss angeordnete Beschlagnahmung des Schlüssels), können sie die aufgezeichnete Kommunikation auch nachträglich noch entschlüsseln. Genau das verhindert PFS. Dabei werden temporäre sog. Sitzungsschlüssel erzeugt, die nur während einer Kommunikationssitzung zwischen zwei Partnern gültig sind. Damit können die Kommunikationspartner die Datenübertragung während der Sitzung verschlüsseln und entschlüsseln. Der Sitzungsschlüssel wird nach Ablauf der

Kommunikationssitzung verworfen. Ein Schlüssel, der nicht mehr existiert, kann nicht gestohlen werden und seine Herausgabe lässt sich auch durch einen Richterspruch nicht erzwingen.

Weitere Infos zu PFS: http://de.wikipedia.org/wiki/Perfect_Forward_Secrecy

5.1.3 HSTS (HTTP Strict Transport Security)

Dieses Protokoll spielt nur bei der Verwendung der Webmail-Schnittstelle, also beim Zugriff auf die Mails mit dem Web-Browser (siehe Kap. 2.7), eine Rolle. HSTS erzwingt den Zugriff auf den Web-Server des Mail-Providers mit HTTPS auch, wenn ein Benutzer diesen über eine HTTP-URL adressieren will. Gibt also ein sorgloser Benutzer im Browser die URL <http://www.mailprovider.de> ein, schaltet der Browser automatisch auf die URL <https://www.mailprovider.de> um. Dass der Browser bei der Adressierung des Servers Protokoll HTTP nicht verwenden und automatisch auf das verschlüsselte HTTPS umschalten soll, dies teilt ihm der Web-Server des Providers über das HSTS-Protokoll mit.

Weitere Infos zu HSTS: <http://de.wikipedia.org/wiki/HTTPS#HSTS>

5.1.4 DANE (DNS-based Authentication of Named Entities)

Mit SSL oder TLS verschlüsselte Transport-Kanäle basieren auf Zertifikaten, mit denen die Server sich gegenüber den Clients ausweisen. Ein Client überprüft das Zertifikat eines Servers, bevor er eine verschlüsselte Transport-Verbindung zu ihm aufbaut. Z.B. prüft ein Web-Browser das Zertifikat eines Web-Servers, bevor er eine HTTPS-Sitzung mit diesem beginnt. Genau so muss das Mail-Programm (z.B. *Thunderbird*) das Zertifikat des Mail-Servers prüfen. Auch der Mail-Server des Absender-Providers (Er ist hier in der Rolle des Client.) muss das Zertifikat des Empfänger-Providers (Er ist in der Rolle des Servers.) prüfen.

Mit gefälschten Zertifikaten lässt sich auch Schindluder treiben. Um das Fälschen der Zertifikate zu verhindern, gibt es weltweit etwas mehr als 200 sog. CAs (Certificate Authorities). Eine CA ist eine Art digitaler Notar, der Zertifikate beglaubigen darf. So müssen z.B. die Browser nur noch diese ca. 200 CAs (genau genommen deren öffentliche Schlüssel) kennen und nicht die Zertifikate von Millionen von Web-Servern. Entsprechendes gilt für die verschlüsselte Übertragung von Mails. Das ganze System steht und fällt mit der Vertrauenswürdigkeit der CAs.

Dieses auf den CAs beruhende System ist angreifbar. Alle ca. 200 CAs können für jeden beliebigen Server (z.B. *mail.google.com*) ein Zertifikat ausstellen. Wird eine einzige CA kompromittiert (z.B. durch einen Hackerangriff oder durch die von einer staatlichen Ermittlungsbehörde oder einem Geheimdienst erzwungene Kooperation), kann ein korrekt beglaubigtes, jedoch falsches Zertifikat ausgestellt werden. Mit diesem gefälschten Zertifikat könnte ein falscher Gmail-Server betrieben werden. Der Browser und das Mail-Programm würden dem falschen Server vertrauen und mit ihm einen verschlüsselten Kommunikationskanal aufbauen in dem Glauben, es handele sich um den echten Gmail-Server.

Kommt DANE zum Einsatz, kann nicht mehr jede CA ein Zertifikat für jeden beliebigen Server erstellen. DANE legt genau fest, welche CA ein Zertifikat für einen Server erstellen darf. Mit DANE können auch sog. selbst-signierte Zertifikate verwendet werden. Dabei wird die CA als Aussteller des Zertifikats überflüssig.

Dieses Verfahren hat den Vorteil, dass der Eigentümer einer Domain (z.B. *google.com*) die Zertifikats-Hoheit für alle Server dieser Domain hat (z.B. für *mail.google.com*, *www.google.com*, *plus.google.com*, *developer.google.com*, etc.). Bei DANE hat Google die Zertifikats-Hoheit für die in der Domain *google.com* betriebenen Server, GMX hat die Zertifikats-Hoheit für alle in den

eigenen Domains (*gmx.net*, *gmx.de*, etc.) betriebenen Server. Dasselbe gilt für alle anderen Domain-Eigentümer, die auch für die Verwaltung der Server der jeweiligen Domains zuständig sind. Das Ausstellen falscher Zertifikate (für Server aus anderen Domains) wird dadurch erheblich erschwert.

Die Zertifikate werden automatisch über das DNS (**Domain Name System**) verteilt. Dieses System ist weit schwerer zu manipulieren als die CAs.

Ein anderes System, das das Dilemma mit den TLS-Server-Zertifikaten zu beheben versucht, ist **EmiG (Email made in Germany)**. Fünf große deutsche Email-Provider (*Telekom*, *Strato*, *1&1*, *GMX*, *WEB.DE*) haben mit EmiG ein eigenes Verfahren entwickelt, bei dem die Provider dieses Verbundes sich gegenseitig zertifizieren. Damit wollen sie den sicher verschlüsselten Mail-Transport zwischen den Providern dieses Verbundes gewährleisten. Grundsätzlich können weitere Provider dem Verbund beitreten. Dazu wird ein neuer Provider zunächst vom TÜV Rheinland zertifiziert. Der TÜV prüft, ob der neue Provider die technischen Voraussetzungen für die Teilnahme am EmiG-Verbund erfüllt.

Zu der Zeit, als EmiG entwickelt wurde, war die Entwicklung des DANE-Standards noch nicht abgeschlossen. Die Teilnehmer des EmiG-Verbundes erklären jedenfalls, sie würden die Einführung von DANE erneut prüfen.

Langfristig dürfte EmiG eine deutsche Insellösung bleiben. DANE ist als globaler Standard besser aufgestellt. *Posteo* und *mailbox.org* sind seit Mai 2014 die beiden ersten deutschen Provider, die den DANE-Standard implementieren, der Anbieter *Mail.de* unterstützt DANE seit Juni. EmiG ist außerdem auf TLS-verschlüsselte Email-Übertragung (SMTPS) ausgerichtet. DANE kann die TLS-verschlüsselte Übertragung für alle Protokolle (SMTPS, IMAPS, HTTPS, FTPS, etc.) sicherer machen.

Um DANE (und EmiG) weiter zu vertiefen, müsste zunächst das DNS erläutert werden. Darauf verzichte ich hier und verweise auf die kurze Beschreibung im Glossar und auf folgende Quellen:

- Artikel „Geleitschutz – DANE verbessert sicheren Transport zwischen Mailservern“ in der c't 2014, Heft 11. Der Artikel kann auch einzeln beim Heise-Verlag erworben werden unter http://www.heise.de/artikel-archiv/ct/2014/11/194_Geleitschutz
- Deutscher Wikipedia-Eintrag zu DNS: http://de.wikipedia.org/wiki/Domain_Name_System
- Englischer Wikipedia-Eintrag zu DANE: <http://en.wikipedia.org/wiki/DANE>
- Erläuterung zu DANE bei der Internet Society: <http://www.internetsociety.org/deploy360/resources/dane/>
- Heise-Online Bericht über den ersten Einsatz von DANE in Deutschland bei Posteo: <http://www.heise.de/netze/meldung/Verschlueselter-Mail-Transport-Posteo-setzt-als-erster-Provider-DANE-ein-2187144.html>
- Zunehmende Verbreitung von DANE: <http://www.heise.de/newsticker/meldung/Mail-Sicherheit-Domain-Anbieter-dotplex-nimmt-DANE-ins-Programm-2263544.html>
- Heise-Online Bericht über EmiG: <http://www.heise.de/netze/meldung/So-funktioniert-E-Mail-made-in-Germany-2188248.html>

5.2 Links zu diesem Kapitel

- Infos zu SSL und TLS unter: http://de.wikipedia.org/wiki/Transport_Layer_Security
- Infos zu PFS: http://de.wikipedia.org/wiki/Perfect_Forward_Secrecy
- Infos zu HSTS: <http://de.wikipedia.org/wiki/HTTPS#HSTS>
- Infos zu DNS: http://de.wikipedia.org/wiki/Domain_Name_System
- Infos zu DANE: <http://en.wikipedia.org/wiki/DANE>
und <http://www.internetsociety.org/deploy360/resources/dane/>
- Posteo als erster deutscher Provider mit DANE-Unterstützung:
<http://www.heise.de/netze/meldung/Verschluesserter-Mail-Transport-Posteo-setzt-als-erster-Provider-DANE-ein-2187144.html>
- Zunehmende Verbreitung von DANE:
<http://www.heise.de/newsticker/meldung/Mail-Sicherheit-Domain-Anbieter-dotplex-nimmt-DANE-ins-Programm-2263544.html>
- Heise-Online Bericht über EmiG:
<http://www.heise.de/netze/meldung/So-funktioniert-E-Mail-made-in-Germany-2188248.html>

6 Glossar

Begriff oder Abkürzung	Erläuterung
Android	Google-Betriebssystem für mobile Geräte (Smartphones und Tablets). Heute gibt es auch Fernseher, Laptops, Smartwatches und andere Geräte wie Kühlschränke, Waschmaschinen, Brillen und Kleinroboter, die auf dem Android-Betriebssystem basieren.
Asymmetrische Verschlüsselung	Für das Verschlüsseln und das Entschlüsseln gibt es zwei unterschiedliche, aber zusammengehörende Schlüssel – den Public Key (siehe dort) und den Private Key (siehe dort). Was mit dem einen verschlüsselt wurde, kann nur mit dem anderen entschlüsselt werden.
Betriebssystem-Kernel	(oder Betriebssystem-Kern) siehe Kernel
CA	Certificate Authority : eine zugelassene Zertifizierungsstelle, die digitale Schlüssel für die verschlüsselte Kommunikation im Internet zertifiziert (beglaubigt).
Client	Ein Programm, das den von einem Server angebotenen Dienst nutzt. Z.B. nutzt ein SMTP-Client den von einem SMTP-Server angebotenen Dienst des Mail-Versands. Das Protokoll definiert die zulässigen die Nachrichten, die der Client und der Server miteinander austauschen. Das SMTP-Protokoll beschreibt die Nachrichten zwischen SMTP-Client und SMTP-Server. (Als Client wird nicht nur das Client-Programm, sondern häufig auch der Rechner, auf dem das Client-Programm läuft, bezeichnet.)
Cloud-Dienst	Cloud-Dienste sind Dienste im Internet mit einer definierten Zugriffsschnittstelle. Die Dienste werden von sehr großen Rechenzentren mit Tausenden von Rechnern zur Verfügung gestellt. Die größten Anbieter sind Amazon, Google und Microsoft. Es gibt jedoch viele weitere. Die bekanntesten Cloud-Dienste sind die Speicherdienste, die auch als <i>Online-Festplatten</i> bekannt sind. Der bekannteste Vertreter ist <i>Dropbox</i> . Dabei werden die Daten des Benutzers wie auf einer Festplatte gespeichert. Tatsächlich werden die Daten jedoch beim Cloud-Anbieter im Internet gespeichert. Der Dienst erlaubt z.B. die automatische Synchronisierung der Daten desselben Benutzers zwischen verschiedenen Geräten.
DANE	DNS-based Authentication of Named Entities : Siehe Kapitel 5.1.4
Daten einer Nachricht	Der eigentliche Inhalt einer Nachricht. Bei einem Brief ist es das, was sich im Umschlag befindet, bei einer Mail ist es der Nachrichtentext.
DNS	Domain Name System : Ein System, mit dem Rechnernamen auf IP-Adressen abgebildet werden. Die Kommunikationsprogramme adressieren sich mit IP-Adressen. Die Menschen verwenden jedoch Rechnernamen. Z.B. spezifiziert man bei der Eingabe einer URL im Browser am Anfang der URL den Rechnernamen (Beispiel: www.google.de). Mit Hilfe des DNS findet der

Begriff oder Abkürzung	Erläuterung
	Browser automatisch die IP-Adresse von www.google.de heraus und verwendet diese, um diesen Server zu adressieren und eine Verbindung zu ihm herzustellen.
DSA	Digital Signature Algorithm : ein Standard der US-Regierung für digitale Signaturen.
Ende-zu-Ende-Verschlüsselung	Bei einer Ende-zu-Ende-Verschlüsselung wird eine Nachricht vom Absender verschlüsselt und erst beim Empfänger wieder entschlüsselt. Auf den verschiedenen Stationen und Teilstrecken der Nachrichtenübermittlung kann die Nachricht nicht entschlüsselt werden.
FTP	File Transfer Protocol : Protokoll zur Übertragung von Dateien über das Netzwerk
GCHQ	Government Communication Headquarters. Britischer Geheimdienst, zuständig für Spionage im Internet.
GPG, GnuPG	GNU Privacy Guard ist die PGP-Implementierung von GNU. (GNU (GNU is Not Unix.) ist eine Organisation, die die Lizenzkosten-freie Verbreitung von Software propagiert.)
HTTP	Hypertext Transfer Protocol : Protokoll zur Übertragung verlinkter Web-Seiten (Hypertext) zwischen vom Web-Server (HTTP-Server) zum Web-Browser (HTTP-Client)
HTTPS	HTTP Secure : Sicheres HTTP wird verwendet wenn Browser und Web-Server über einen verschlüsselten Transport-Kanal miteinander kommunizieren.
HSTS	HTTP Strict Transport Security : Ein Verfahren, das sicherstellt, dass der Browser bei der Kommunikation mit einem bestimmten Web-Server immer HTTPS verwendet, auch wenn der Benutzer eine HTTP-URL eingibt. Siehe Kapitel 5.1.3
IMAP	Internet Mail Access Protocol : ein Protokoll zum Zugriff und auch die Verwaltung der empfangenen und beim Provider gespeichert Mails. Anders als bei POP (siehe dort) bleiben die Mails beim Provider gespeichert. Dadurch ist der Zugriff mit vielen Geräten auf denselben Mail-Bestand möglich.
Implementation	Umsetzung, Realisierung, Erfüllung (eines Vertrages)
iOS	Betriebssystem der mobilen Geräte von Apple (iPhone, iPad und iPod).
Jailbreak	Bei einem iPhone oder iPad sind die Apps in ihren Rechten beschränkt. Sie dürfen nur das tun, wozu sie berechtigt sind. Beispielsweise können nur Apps aus dem Apple App Store installiert werden. Ein Jailbreak (Ausbruch aus dem Gefängnis) ermöglicht es, diese Beschränkungen aufzuheben und gewährt den uneingeschränkten Zugriff auf das Gerät. Nach dem Jailbreak stehen z.B. auch alternative App Stores, die nicht Apples Segen haben, offen.
Kernel	Kern des Betriebssystems. Ein Betriebssystem besteht aus dem Kern und den (Benutzer-)Programmen. Der Benutzer bedient die Benutzer-Programme, z.B. den Browser, den Mail-Client, den Datei-Manager und viele weitere. Benutzer-

Begriff oder Abkürzung	Erläuterung
	Programme werden gestartet und können (wenn sie nicht mehr gebraucht werden) beendet werden. Sie sind nicht immer aktiv. Der Kern des Betriebssystems ist immer aktiv vom Starten bis zum Herunterfahren des Rechners. Der Kern erledigt (für die Benutzerprogramme) dauernd zentrale Aufgaben wie Zuweisung der Rechenzeit, Verwaltung des Speichers, Zugriffe auf Festplatten und andere Speichermedien, Zugriffe auf das Netzwerk. Man könnte sagen, der Kernel ist der zentrale Manager, der verhindert, dass sich die verschiedenen Benutzerprogramme gegenseitig in die Quere kommen. So verhindert er z.B., dass diese ihre Daten auf der Festplatte oder im Hauptspeicher gegenseitig überschreiben. Oder er verhindert, dass die Mail, die für den Mail-Client bestimmt ist, plötzlich vom Browser gelesen wird.
KRC	Key Revocation Certificate: Mit dem Widerrufszeugnis kann ein Schlüssel (auch ohne die Passphrase) widerrufen, d.h. für ungültig erklärt werden. Wie der private Schlüssel sollte auch das Widerrufszeugnis nicht in die Hände fremder Personen fallen.
Mail-Alias	Alternative Mail-Adresse, die zusätzlich zur Haupt-Mail-Adresse bei vielen Mail-Providern für denselben Mail-Account eingerichtet werden kann. Beispielsweise kann sich Joseph Mayer mit der Mail-Adresse joseph.mayer@web.de den Mail-Alias joseph@mayer.de einrichten, falls dieser Alias nicht bereits vergeben ist.
Mail-Provider	Siehe Provider
Malware	(Malicious Software) Schädliche Software, die unerwünschte Aktivitäten auf einem Rechner ausführt. Dazu gehören Viren, Trojaner, Spähprogramme, etc.
Metadaten einer Nachricht	Die Nachrichtenattribute außer dem Inhalt. Bei einem Brief ist es das, was sich auf dem Umschlag befindet (Absender, Empfänger, Briefmarke, Poststempel), bei einer Mail sind es Absender-Adresse, Empfänger- und CC-Adressen, Betreff, Nachrichtenformat, Versandzeitpunkt, Verschlüsselungsinformation, etc.
MIME	Multi Purpose Internet Mail Extensions: Mail-Übertragungsformat, bei dem Inhalt und Anhänge als getrennte Blöcke in der Mail enthalten sind. Das besondere Merkmal dabei ist, dass jeder Block, je nach Art des Blockinhalts (Klartext, HTML-Text, Image, Video, Audio oder auch ein PDF-Dokument) einen anderen Inhaltstyp (oder <i>content type</i>) hat. Der Inhaltstyp jedes Blocks wird durch einen sog. <i>mime type</i> oder <i>Medientyp</i> in den Metadaten der Mail beschrieben. Beispiele für <i>Medientypen</i> sind: <i>text/plain</i> für Klartext, <i>text/html</i> für HTML-Text, <i>image/jpeg</i> für Bilder im JPEG-Format, <i>image/gif</i> für Bilder im GIF-Format, <i>audio/basic</i> , <i>video/mpeg</i> oder <i>application/pdf</i> für PDF-Dokumente. Dadurch, dass die Art eines Anhangs in der Mail gespeichert ist, weiß das Mail-Programm des Empfängers mit welchem Zusatz-Programm es den Inhalt eines Anhangs darstellen kann. Zum Beispiel kann <i>Thunderbird</i> zur Darstellung eines PDF-Anhangs den Adobe-Reader starten und zur Darstellung eines Videos den installierten Media-Player. Klartext und HTML-Text kann <i>Thunderbird</i> selbst darstellen und benötigt dazu kein Zusatz-

Begriff oder Abkürzung	Erläuterung
	<p>Programm.</p> <p>Ebenso gibt es auch <i>mime types</i> für signierte und verschlüsselte Mail-Blöcke (Anhänge): <i>multipart/signed</i> für einen signierten Block, <i>multipart/pcs7-mime</i> für einen mit S/MIME verschlüsselten Block und <i>multipart/encrypted</i> für einen mit PGP verschlüsselten Block. <i>Thunderbird</i> weiß, dass diese Blöcke nach dem Empfang einer Signatur-Prüfung zu unterziehen sind, bzw. dass sie entschlüsselt werden müssen. Ist <i>Enigmail</i> installiert, so kann <i>Thunderbird</i> das selbst erledigen und benötigt kein externes Zusatz-Programm.</p>
NSA	National Security Agency. Amerikanischer Geheimdienst, zuständig für Spionage im Internet.
Open Source	Open Source Programme nennt man Programme, deren Quelltext öffentlich verfügbar gemacht wird. Damit ist grundsätzlich jeder Softwareentwickler, der das entsprechende fachliche Know-how dazu hat, in der Lage, die exakte Funktionsweise des Programms zu überprüfen.
Passphrase	In der PGP-Terminologie wird das Passwort, das zum Zugriff auf den privaten Schlüssel verwendet wird, als Passphrase bezeichnet. Für jeden privaten Schlüssel wird eine eigene Passphrase festgelegt.
Passwort-Manager	<p>Passwort-Verwaltungsprogramm. Der heutige Internet-Benutzer hat häufig fünfzig und mehr (hoffentlich unterschiedliche) Passwörter für unterschiedliche Accounts. Diese kann er sich in der Regel nicht auswendig merken. Klassischerweise schreibt er sich alle Passwörter auf eine Liste, die er aber nicht verlieren darf. Er kann die Passwörter auch einem Passwort-Manager anvertrauen und auf dem Rechner speichern. Der Passwort-Manager speichert die Passwörter verschlüsselt ab und gibt sie nur wieder preis, wenn man das Master-Passwort richtig eingibt. Der Benutzer muss sich nur noch das Master-Passwort des Passwort-Managers merken.</p> <p>Der Passwort-Manager kann mit einem Schlüsselkasten verglichen werden und das Master-Passwort mit dem Schlüssel, das den Schlüsselkasten öffnet.</p>
PFS	Perfect Forward Secrecy: Ein Verfahren, das auch das nachträgliche Entschlüsseln einer aufgezeichneten, verschlüsselten Kommunikation verhindert. Siehe Kapitel 5.1.2
PGP	Pretty Good Privacy: Ein Verfahren zur Übertragung Ende-zu-Ende verschlüsselter Mails mit asymmetrischer Verschlüsselung. Dabei wird das MIME-Format nicht verwendet. Mail-Inhalt und Mail-Anhänge werden in inline (d.h. als ein zusammenhängender Block) verschlüsselt. Bei PGP werden die öffentlichen Schlüssel von den Benutzern wechselseitig beglaubigt. Dadurch entsteht ein sogenanntes WoT oder Web of Trust (siehe dort).
PGP/MIME	Ein Verfahren zur Übertragung Ende-zu-Ende verschlüsselter Mails mit asymmetrischer Verschlüsselung. Dabei wird das MIME-Format (siehe dort) eingehalten. Dabei wird der Mail-Inhalt und jeder Anhang der Mail separat verschlüsselt. PGP/MIME ist das modernere Verfahren, das jedoch von machen Tools noch nicht unterstützt wird. Die Verschlüsselung funktioniert genauso wie beim klassischen Inline-PGP (siehe dort), nur das

Begriff oder Abkürzung	Erläuterung
	Nachrichtenformat ist anders.
POP	Post Office Protocol: Protokoll zum Abruf von Mails. Anders als bei IMAP (siehe dort) werden die Mails auf den Rechner des Benutzers heruntergeladen und normalerweise vom Server des Providers gelöscht. POP lässt sich nur mit einem Gerät sinnvoll verwenden und wird deshalb heute nur noch selten eingesetzt.
Private Key	Der private Schlüssel eines Schlüsselpaars verbleibt immer beim Eigentümer des Schlüssels und ist geheim. Der Schlüsseleigentümer muss darauf achten, den Schlüssel niemals herauszugeben oder zu verlieren. Der private Schlüssel wird vom Schlüsseleigentümer benutzt, um die an ihn gerichteten Nachrichten zu entschlüsseln und um Nachrichten, die er versendet, zu signieren. (Siehe auch asymmetrische Verschlüsselung)
Protokoll	<p>Ein Protokoll ist ein Satz von Regeln, welche das Format, den Inhalt, die Bedeutung und die Reihenfolge von Nachrichten zwischen verschiedenen Instanzen (z.B. zwischen Client und Server) festlegen. Nur dadurch dass beide dieselben Nachrichten „verstehen“, können sie sinnvoll miteinander kommunizieren. Beispiele:</p> <ul style="list-style-type: none"> • Das HTTP (Hypertext Transfer Protocol) regelt das Format der Nachrichten zwischen dem HTTP-Client (Browser) und dem HTTP-Server (Web-Server). • Das FTP (File Transfer Protocol) regelt die Nachrichten zur Übertragung von Dateien zwischen dem FTP-Client und dem FTP-Server. • Das SMTP (Simple Mail Transfer Protocol) regelt die Nachrichten zur Übertragung von Mails zwischen dem SMTP-Client (<i>Thunderbird</i>) und dem SMTP-Server des Providers. <p>Es gibt sehr viele weitere Protokolle, die jeweils unterschiedlichen Zwecken dienen.</p>
Provider	Dienstanbieter. Firma, die einen Dienst bereitstellt. Z.B. sind GMX und WEB.DE Mail-Provider. Sie stellen den Mail-Dienst bereit. Natürlich gibt es für andere Dienste andere Provider. (engl.: to provide = bereitstellen, anbieten, (Leistung oder Dienst) erbringen)
Public Key	Der öffentliche Schlüssel eines Schlüsselpaars wird möglichst vielen anderen Benutzern zugänglich gemacht (typischerweise auf sog. Key-Servern wo sie von jedem heruntergeladen werden können. Der öffentliche Schlüssel wird von anderen Benutzern als dem Schlüsseleigentümer benutzt, um die Nachrichten an den Schlüsseleigentümer zu verschlüsseln und um die Signatur von Nachrichten, die vom Schlüsseleigentümer stammen, zu verifizieren. (Siehe auch asymmetrische Verschlüsselung)
Quelltext oder Quellcode	Der Softwareentwickler entwickelt ein Computer-Programm, indem er sog. Quelltext (oder Quellcode, engl. Source Code) schreibt. Dieser Quelltext ist in einer Programmiersprache geschrieben und ist menschen-lesbar. Die Maschine (der Computer) versteht diesen Text nicht und kann deshalb das Programm in

Begriff oder Abkürzung	Erläuterung
	dieser Form nicht ausführen. Dazu muss der Quellcode erst von einem Übersetzer (Compiler) übersetzt werden. Das Ergebnis der Übersetzung (Compilation) ist der Zielcode, bzw. der Maschinencode. Der Maschinencode ist für Menschen nicht lesbar, jedoch die Maschine (der Computer) kann ihn lesen und so das Programm ausführen.
RSA	Asymmetrisches kryptographisches Verfahren benannt nach seinen Erfindern R ivest, S hamir und A dleman
Schlüsselbund	Der PGP-Schlüsselbund (auch PGP-Schlüsselverwaltung genannt) enthält alle (öffentlichen und privaten) Schlüssel, die zur verschlüsselten und/oder signierten Kommunikation verwendet werden. Typischerweise enthält er das eigene Schlüsselpaar (bestehend aus dem eigenen öffentlichen und privaten Schlüssel). Außerdem enthält er die öffentlichen Schlüssel aller Kommunikationspartner.
Server	Programm, das einen bestimmten Dienst anbietet. Der Client nutzt den angebotenen Dienst. Ein SMTP -Server z.B. steht in der Regel beim Mail-Provider und bietet dem SMTP -Client den Dienst an, Mails zu versenden. Das Protokoll definiert die Nachrichten, die der Client und der Server miteinander austauschen. Das SMTP-Protokoll beschreibt die Nachrichten zwischen SMTP-Client und SMTP-Server. (Als Server wird nicht nur das Server-Programm, sondern häufig auch der Rechner, auf dem das Server-Programm läuft, bezeichnet.)
S/MIME	Secure MIME : ein Verfahren zur Übertragung Ende-zu-Ende verschlüsselter Mails mit asymmetrischer Verschlüsselung. Dabei wird das MIME-Format (siehe dort) eingehalten. Bei S/MIME werden die öffentlichen Schlüssel von bestimmten Zertifizierungsstellen, den CAs (siehe dort) beglaubigt.
Software-Repository	Zentrale Software-Aufbewahrungsstelle. Bei den Linux-Distributionen gibt es ein solches zentrales Software-Repository, an dem praktisch alle Software-Pakete eine Linux-Distribution wie Ubuntu Linux aufbewahrt wird. Dies macht die Aktualisierung des Systems sehr einfach. Diese ist mit ein paar Mausklicks erledigt. Bei Windows gibt es kein solches Repository. Deshalb kann es recht aufwändig werden, alle Programme auf einem Windows-System aktuell zu halten. Bei iOS und Android ist der Apple App Store bzw. der Google Play Store ein solches Software-Repository.
SKS	Synchronizing Key Server : ein Key-Server, der sich automatisch mit anderen Key-Servern synchronisiert. Dadurch haben die Key-Server weltweit einen nahezu gleichen Datenbestand.
SMS	Short Message Service: Kurznachrichtendienst, der es ermöglicht Textnachrichten über das Sprachnetz zu versenden. Ein Smartphone mit Internetzugang ist dafür nicht erforderlich. Das „gute, alte“ Handy (das heute kaum noch zu kaufen gibt) genügt.
SMTP	Simple Mail Transfer Protocol : ein Protokoll zur Versenden von Mails.
SSL	Secure Socket Layer : alte Bezeichnung für TLS

Begriff oder Abkürzung	Erläuterung
STARTTLS	Dies ein Verfahren, eine unverschlüsselte Verbindung in eine mit SSL oder TLS verschlüsselte Verbindung umzuwandeln und damit „aufzuwerten“.
Symmetrische Verschlüsselung	Eine Nachricht wird immer mit dem Schlüssel entschlüsselt, mit dem sie auch verschlüsselt wurde. Dieses Verfahren ist bei Nachrichtenübertragungen unzuweckmäßig, da der Absender zum Verschlüsseln und der Empfänger zum Entschlüsseln denselben Schlüssel benötigen.
TLS	Transport Layer Security : ein Verfahren, um einen sicheren Übertragungskanal zwischen zwei Instanzen zum Transport von Daten aufzubauen. Das Verfahren für verschiedene Protokolle eingesetzt, z.B. für HTTPS (verschlüsselte Übertragung zwischen Browser und Web-Server). Das Verfahren kommt auch zum Einsatz, beim Abruf von Mails (mit POP oder IMAP), beim Versand von Mails (mit SMTP) oder bei der Übertragung von Mails zwischen den Providern. Mit TLS kann bei der Mail-Übertragung keine Ende-zu-Ende-Verschlüsselung (siehe dort) sichergestellt; es sind nur die Teilstrecken der Übertragung zwischen zwei Instanzen verschlüsselt.
User-Tracking	Verfolgung eines Benutzers im Internet. Dabei werden die Spuren, die ein Benutzer bei der Nutzung des Internet hinterlässt, verfolgt (getrackt) und gespeichert. So lässt sich bei längerer Beobachtung des Benutzerverhaltens im Internet ein recht genaues Persönlichkeitsprofil des betreffenden Benutzers erstellen. Dies lässt sich benutzen, um dem Benutzer gezielt Werbung auf Web-Seiten anzuzeigen, die genau auf ihn und seine Interessen zugeschnitten sind.
Vertrauen	Vertrauen sich die Personen A und B (ohne die Vermittlung einer weiteren Person), so spricht man von direktem Vertrauen . Wenn Person A der Person B vertraut und B vertraut C, kann auch A der Person C vertrauen, obwohl er C gar nicht kennt. In diesem Fall spricht man von transitivem Vertrauen . (Siehe auch WoT, Web of Trust)
Webmail	Zugriff auf den Mail-Account mit dem Internet-Browser wie Chrome, Firefox, Internet-Explorer, usw. Der Zugriff auf die Mails mit dem Browser ist eine Alternative zum Mail-Zugriff mit einem speziellen Mail-Client, z.B. Thunderbird, Outlook, Apple Mail und viele andere.
WoT	Web of Trust : ein Vertrauensgeflecht, das sich aus direkten Vertrauensbeziehungen (siehe dort) und transitiven Vertrauensbeziehungen (siehe dort) zusammensetzt.