

# **Sicher Mailen – Wie geht das?**

**Autor: Hermann Hueck**

**Version 4.0**

**Letzte Änderung: 31.10.2014**



## Vorwort

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 (des 20. Jahrhunderts) 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 (ohne Vorwort und Verzeichnisse) rund 150 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 3 und 4 enthalten die grundlegenden (und auch die einfacheren) Schritte zur

Implementierung von sicherer Email-Nutzung. Das kann jede/jeder schaffen. Sie/er muss dazu nur die ersten ca. 40 Seiten bewältigen und hat damit eine solide Grundlage für die sichere Nutzung der Mail.

Wer das Maximum an Email-Sicherheit erreichen will, der kann sich – auf dieser Grundlage aufbauend – zusätzlich mit der Email-Verschlüsselung in Kapitel 5 bis 10 beschäftigen und diese umsetzen.

Ich habe dieses Dokument 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 sind vor allem die Kapitel 5 bis 10 für den Laien, der sich das erste Mal mit dem Thema Email-Verschlüsselung beschäftigt, durchaus eine Herausforderung. Es handelt sich dabei um eine komplizierte Materie, die sich nicht beliebig vereinfachen lässt.

Dieses Dokument will nicht alle Varianten und Möglichkeiten für alle denkbaren Systeme und Programme beschreiben. Es beruht sehr auf meinen eigenen Erfahrungen. So wird z.B. die PGP-Konfiguration in Thunderbird beschrieben. Andere Mail-Programme werden nicht oder nur kurz erwähnt. Auch kann ich mehr Details zu meinem Mail-Provider liefern. Andere Mail-Provider werden nicht oder nicht so ausführlich beschrieben. Da ich selbst kein iPhone- und iPad-Nutzer bin, sondern Android-Geräte nutze, ist die PGP-Konfiguration für diese Geräte recht ausführlich beschrieben. iOS fällt dabei weitgehend unter den Tisch.

Möchte einer meiner Leser seine Erfahrungen mit einem anderen System (z.B. Linux, iOS oder Windows Phone), einem anderen Programm (z.B. Outlook, Apple Mail, Mailpile etc.) oder mit einem anderen Mail-Provider beisteuern, so bin ich dafür offen. Der Betreffende kann per Mail Kontakt zu mir aufnehmen, um die gewünschte Erweiterung direkt mir zu diskutieren.

**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](mailto:hermann.hueck@secure.mailbox.org) (oder an [hermann.hueck@mailbox.org](mailto:hermann.hueck@mailbox.org)).

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

Kurze Antwort: Die erste Adresse ist sicherer, sie funktioniert nicht immer, jedoch 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 von Kapitel 3 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.

Ein paar Worte zur Weitergabe dieses Dokuments: Das Dokument ist in erster Linie für meine Kommunikationspartner geschrieben, in zweiter Linie jedoch auch für alle Interessierten. Der Download-Link für das Dokument darf also gerne an Freunde, Familienmitglieder, Kollegen etc. weitergegeben werden.

Und nun viel Spaß beim Lesen und Ausprobieren!

Mai 2014

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 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 Kapitel 2 mit den Unterkapiteln 2.4, 2.5 und 2.6 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 der Erläuterung der Transport-Verschlüsselung habe ich diese in das Kapitel 14 ausgelagert. (So werden die Unzulänglichkeiten der Transport-Verschlüsselung und die Protokolle, mit denen man diese Schwachstellen beheben kann, im Kapitel 3.2 gestreift und im Kapitel 14 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 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.

Juli 2014

Hermann Hueck

## Vorwort zur Version 3.0

In Version 3.0 habe ich den Text nach dem zweiten Lektorat durch einen Freund nochmals überarbeitet und verbessert. Mein Dank sei ihm gewiss.

Von einem Leser erhielt ich als Feedback den Hinweis, er vermisste einen Einstieg in PGP in der Art eines kurzen How-To ohne viele Erläuterungen. Ich bin mir nicht ganz sicher, ob der Schnelleinstieg für den IT-Laien der bessere ist. Doch diese Entscheidung soll die künftige Leserin oder der künftige Leser selbst treffen.

Ich habe das ursprüngliche Kapitel über PGP-verschlüsselte Mails in mehrere Kapitel zerlegt und dann zwei PGP-Schnelleinstiege – einen für den PC und einen für das Android-Smartphone oder -Tablet – als neue Kapitel eingefügt. Durch die Aufteilung und durch Erweiterungen sind aus einem Kapitel mittlerweile folgende Kapitel entstanden:

- **Kapitel 5:** Verschlüsselte und signierte Mails mit PGP – Die „graue“ Theorie
- **Kapitel 6 (neu):** PGP auf dem PC – Schnelleinstieg für Ungeduldige
- **Kapitel 7:** PGP auf dem PC – Ausführlicher Einstieg für Wissbegierige
- **Kapitel 8:** PGP – Was noch wichtig oder interessant ist
- **Kapitel 9 (neu):** PGP auf dem Android-Gerät – Schnelleinstieg für Ungeduldige
- **Kapitel 10:** PGP auf dem Android-Gerät – Ausführlicher Einstieg für Wissbegierige

Dabei habe ich in Kauf genommen, dass es zu Dopplungen zwischen den Kapiteln 6 und 7 sowie zwischen den Kapiteln 9 und 10 kam.

Nun kann die Leserin oder der Leser selbst wählen, ob er den Schnelleinstieg oder die ausführlichen Erläuterungen wählt. Genügt ihm der Schnelleinstieg nicht, so kann er durch entsprechende Kapitelverweise jeder Zeit in das entsprechende ausführliche Kapitel springen, um den entsprechenden Konfigurationsschritt nochmals in detaillierter Ausführung zu lesen.

Have just more fun with PGP.

August 2014

Hermann Hueck

## Vorwort zur Version 4.0

Diese Version hat dem Dokument durch redaktionelle Überarbeitung einen weiteren Feinschliff angedeihen lassen.

Die ersten beiden Kapitel wurden jeweils in zwei Kapitel zerlegt, sodass vier Kapitel daraus entstanden sind. Ich hoffe, dadurch eine noch bessere Strukturierung des Inhaltes zu erreichen.

Neu hinzugekommen sind die Kapitel 11 und 13.

Kapitel 11 (PGP/MIME – Und es funktioniert doch.) beschreibt, wie man das modernere Verschlüsselungsformat PGP/MIME nutzen kann. In der vorigen Version hatte ich die Verwendung von PGP/MIME noch nicht empfohlen, da es dafür noch keine Unterstützung unter Android gab. Mit der Android Mail-App *Squeaky Mail* ist seit Sommer 2014 eine Mail-App verfügbar, die PGP/MIME unterstützt. Wie man es mit *Squeaky Mail* auf dem Android-Smartphone und in *Thunderbird* auf dem PC nutzen kann, erläutere ich in diesem neuen Kapitel.

Email ist nicht die einzige Kommunikationsart die wir Internetnomaden heutzutage nutzen. In Kapitel 13 (Verschlüsselte Mails – und was noch?) verlasse ich den Thema Email, um das es in diesen Ausführungen in erster Linie geht. Wer dieses Dokument bis zu Kapitel 12 gelesen hat, hat so viel über sichere Kommunikation und Verschlüsselung gelernt, dass es nur konsequent ist, dieses Wissen auch auf andere Kommunikationsdienste anzuwenden. Deshalb geht es in diesem Kapitel um verschlüsselte SMS, abhörsichere Telefonate und sichere Groupware-Dienste (Adressbuch, Kalender, Aufgabenlisten). Schließlich erörtere ich auch die sichere Nutzung von Cloud-Speicherdienssten à la *Dropbox*. In diesem Kapitel verwerte ich sozusagen das bei der Email Gelernte für andere Kommunikationsdienste. In allen Fällen ist es einfacher zu verstehen und deshalb viel kompakter darstellbar als das große Thema „Sichere Mail“.

Mir ist klar, dass dies immer noch nicht alles ist. Sicherheitsfragen stellen sich unter Anderem auch bei der Nutzung sozialer Netzwerke und vor allem beim Surfen im Web. Dies sind allerdings noch zwei große Themen, die den Rahmen dieses auf die Email ausgerichteten Dokuments deutlich sprengen würden und deshalb hier nicht zur Sprache kommen.

Auch für diese neue Version wünsche ich viel Freude bei der Lektüre und natürlich auch beim Einrichten der eigenen Geräte.

Oktober 2014

Hermann Hueck

## Kapitelübersicht

Was enthalten die Kapitel des Dokuments?

- **Kapitel 1, Sichere Email – warum überhaupt?**, ist das „Motivationskapitel“. Hier versuche ich darzustellen, warum Email-Sicherheit überhaupt wichtig ist.
- **Kapitel 2, Grundlegende Konzepte**, führt in das Thema Email-Sicherheit ein. Es liefert die grundlegenden Konzepte zu diesem Thema. 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 3, (Möglichst) Sicherer Versand unverschlüsselter Mails**, behandelt die Auswahl des passenden Email-Providers und behandelt die *Thunderbird*-Einstellungen, um sich sicher mit dem Provider zu verbinden.
- **Kapitel 4, Tipps für die sichere Mail-Nutzung**, liefert Hinweise und Erläuterungen zur sicheren Email-Konfiguration und -Verwendung. Es enthält alles, was bei der unverschlüsselten Mail-Kommunikation zu beachten ist.
- **Kapitel 5 bis 11 beschäftigen sich mit Mail-Verschlüsselung und Mail-Signierung mit PGP**. 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 5, Verschlüsselte und signierte Mails mit PGP – Die „graue“ Theorie**, führt in die Konzepte der asymmetrischen Verschlüsselung mit PGP ein.
- **Kapitel 6, PGP auf dem PC – Schnelleinstieg für Ungeduldige**, enthält den PGP-Schnelleinstieg für Ungeduldige, eine kochrezeptartige Darstellung der Einrichtung und Nutzung von PGP auf dem PC mit *Thunderbird* und *Enigmail*.
- **Kapitel 7, PGP auf dem PC – Ausführlicher Einstieg für Wissbegierige**, liefert den ausführlichen Einstieg in die Einrichtung und Nutzung von PGP auf dem PC mit *Thunderbird* und *Enigmail*. Dieses Kapitel bietet mehr Erläuterungen, zeigt mehr Alternativen und Optionen auf. Der Text ist mit Screenshots angereichert.
- **Kapitel 8, PGP – Was noch wichtig oder interessant ist**, bietet weitere Informationen zur Nutzung von PGP, die über die Konfiguration von *Thunderbird* und *Enigmail* hinausgehen.
- **Kapitel 9, PGP auf dem Android-Gerät – Schnelleinstieg für Ungeduldige**, enthält den PGP-Schnelleinstieg für Ungeduldige, eine kochrezeptartige Darstellung der Einrichtung und Nutzung von PGP auf dem Android-Smartphone oder -Tablet.
- **Kapitel 10, PGP auf dem Android-Gerät – Ausführlicher Einstieg für Wissbegierige**, liefert den ausführlichen Einstieg in die Einrichtung und Nutzung von PGP auf dem Android-Smartphone oder -Tablet. Dieses Kapitel bietet mehr Erläuterungen, zeigt mehr Alternativen und Optionen auf. Der Text ist mit Screenshots angereichert.
- **Kapitel 11, PGP/MIME – Und es funktioniert doch.**, zeigt wie man PGP/MIME auf Android-Geräten und auf dem PC verwendet.
- **Kapitel 12, Passwortsicherheit und Rechnersicherheit**, enthält einige Grundsätze zur Passwortsicherheit und zur Rechnersicherheit, die – wenn man sie beachtet – auch der Email-Sicherheit zugute kommen.
- **Kapitel 13, Verschlüsselte Mails – und was noch?**, beschäftigt sich mit anderen

Anwendungsbereichen jenseits der Email (SMS, Telefonie, Groupware-Dienste und Cloud-Dienste) und gibt Hinweise, wie diese Dienste möglichst sicher genutzt werden können. Die bei der Email „gelernten“ Sicherheitsmaßnahmen (insbesondere die Verschlüsselung) kommen nun bei anderen Kommunikationsdiensten zur Anwendung.

- In **Kapitel 14, Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?**, geht der Frage nach, wie sicher verschlüsselte Übertragungskanäle denn wirklich sind. Es zeigt auf, welche technischen Qualitätskriterien eine hochwertige Transport-Verschlüsselung aufweisen sollte und wie man dies überprüfen kann. Dieses recht technisch ausgerichtete Kapitel ist zum Verständnis des restlichen Dokuments nicht erforderlich. Es richtet sich eher an den mehr technisch interessierten Leser.
- **Kapitel 15** enthält das **Glossar**, in dem Fachbegriffe und Abkürzungen erläutert werden.

# Versionshistorie

Version	Datum	Beschreibung
1.0	01.05.2014	<ul style="list-style-type: none"> <li>Initiale Fassung</li> </ul>
2.0	25.07.2014	<ul style="list-style-type: none"> <li>Redaktionelle und inhaltliche Überarbeitung nach erstem Lektorat</li> <li>Einführung der Versionshistorie</li> <li>Vorwort zur Version 2.0</li> <li>Neue Kapitel 2.4, 2.5 und 2.6 liefern wichtige Grundbegriffe der Netzwerkkommunikation</li> <li>Neues Kapitel 3.2: Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?</li> <li>Neues Kapitel 14.6: Die Qualität der Transport-Verschlüsselung der Mail-Provider prüfen</li> <li>Neues Kapitel 7.2.3: Text-Mails statt HTML-Mails</li> <li>Neues Kapitel 7.2.4: Die verschlüsselt versendeten Mails lesbar machen</li> <li>Neues Kapitel 7.3.3: Empfängerregeln</li> <li>Ergänzung in Kap. 8.1: PGP-Unterstützung für den Chrome-Browser</li> <li>Neues Kapitel 8.5: Das verschlüsselte Postfach von mailbox.org</li> <li>Neues Kapitel 14: Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?</li> <li>Kap. 15: Glossar erweitert</li> </ul>
3.0	14.08.2014	<ul style="list-style-type: none"> <li>Erneute Überarbeitung nach zweitem Lektorat</li> <li>Kapitel 3, das in Version 2.0 noch das ganze PGP-Kapitel war, wurde aufgeteilt in die Kapitel 5, 7, 8 und 10</li> <li>Neues Kapitel 1.3: Scannen von Mails – zu welchem Zweck?</li> <li>Neues Kapitel 6: PGP auf dem PC – Schnelleinstieg für Ungeduldige</li> <li>Neues Kapitel 9: PGP auf dem Android-Gerät – Schnelleinstieg für Ungeduldige</li> <li>Kap. 15: Glossar erweitert</li> </ul>
4.0	31.10.2014	<ul style="list-style-type: none"> <li>Erneute Überarbeitung nach drittem Lektorat</li> <li>Kapitel 1 (früher: <i>Einleitung</i>) aufgeteilt in zwei Kapitel. Dadurch wurden die nachfolgenden Kapitelnummern um eins hochgezählt. <ul style="list-style-type: none"> <li>Kapitel 1: Sichere Email – warum überhaupt?</li> <li>Kapitel 2: Grundlegende Konzepte</li> </ul> </li> <li>Kapitel 2 (früher: <i>Sicherer Versand unverschlüsselter Mails</i>) aufgeteilt in zwei Kapitel. Dadurch wurden die nachfolgenden Kapitelnummern um eins hochgezählt. <ul style="list-style-type: none"> <li>Kapitel 3: (Möglichst) Sicherer Versand unverschlüsselter Mails</li> <li>Kapitel 4: Tipps für die sichere Mail-Nutzung</li> </ul> </li> <li>Kapitel 4.4: Gefahren beim Empfang von HTML-Mails abwenden: aktualisiert und erweitert</li> <li>Neues Kapitel 11: PGP/MIME – Und es funktioniert doch.</li> <li>Neues Kapitel 13: Verschlüsselte Mails – und was noch?</li> <li>Kapitel 14.6: Die Qualität der Transport-Verschlüsselung der Mail-Provider prüfen: aktualisiert und stark erweitert</li> <li>Kap. 15: Glossar erweitert</li> <li>Neu: Zusammenfassung am Ende jedes Hauptkapitels</li> </ul>

# Inhaltsverzeichnis

1 Sichere Email – warum überhaupt?.....	1
1.1 Verlust der Privatsphäre im Internet.....	1
1.2 Sichere Email – warum?.....	3
1.3 Scannen von Mails – zu welchem Zweck?.....	4
1.4 Kein Google-Bashing.....	6
1.5 Die Nachteile der Mail-Verschlüsselung.....	6
1.6 Zusammenfassung.....	7
1.7 Links zu diesem Kapitel.....	7
2 Grundlegende Konzepte.....	9
2.1 Die Daten und Metadaten allgemein.....	9
2.2 Die Daten und die Metadaten der Email.....	9
2.3 Sichere Email – Zuständigkeiten.....	10
2.4 Client und Server.....	10
2.4.1 Spezialisierte Clients und Server.....	11
2.4.2 Client und Server sind Kommunikationsrollen.....	11
2.4.3 Kommunikationsphasen.....	11
2.5 Protokoll.....	12
2.6 Verschlüsselung.....	13
2.6.1 Verschlüsselung einer Datei – ein Beispiel.....	14
2.6.2 Transport-Verschlüsselung vs. Daten-Verschlüsselung.....	14
2.6.3 Transport-Verschlüsselung.....	15
2.6.4 Daten-Verschlüsselung.....	16
2.7 Zusammenfassung.....	17
2.8 Wer mehr wissen will .....	17
2.9 Links zu diesem Kapitel.....	18
3 (Möglichst) Sicherer Versand unverschlüsselter Mails.....	19
3.1 Die Mail auf dem Transportweg.....	19
3.2 Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?.....	22
3.3 Der richtige Email-Provider.....	24
3.3.1 Auswahlkriterien.....	24
3.3.2 Meine Auswahl.....	25
3.4 Ein paar Bemerkungen zu Gmail.....	26
3.5 Konfiguration des Mail-Clients.....	27
3.5.1 Thunderbird-Konfiguration.....	27
3.6 Die passende Email-App auf dem Smartphone.....	29
3.7 Sicherer Mail-Alias bei mailbox.org.....	29
3.8 Zusammenfassung.....	30
3.9 Links zu diesem Kapitel.....	30
4 Tipps für die sichere Mail-Nutzung.....	32
4.1 Rechnersicherheit.....	32
4.2 Thunderbird-Updates.....	32
4.3 Gefahren in Mail-Anhängen abwenden.....	32
4.4 Gefahren beim Empfang von HTML-Mails abwenden.....	33
4.4.1 Die Ausführung von JavaScript muss deaktiviert sein.....	34
4.4.2 Die Ausführung von Java-Applets muss deaktiviert sein.....	35
4.4.3 Vorsicht bei Links.....	36
4.4.4 User-Tracking verhindern.....	36
4.5 Warnung vor der Nutzung von Webmail.....	39
4.6 Zusammenfassung.....	39
4.7 Links zu diesem Kapitel.....	40

5 Verschlüsselte und signierte Mails mit PGP – Die „graue“ Theorie.....	41
5.1 Asymmetrische Verschlüsselung.....	41
5.2 Welches Verschlüsselungsverfahren – S/MIME oder PGP?.....	43
5.3 Private Key und Public Key – Wie funktioniert's?.....	44
5.4 Einschränkungen bei der Nutzung verschlüsselter Mails.....	46
5.5 Zusammenfassung.....	46
5.6 Endlich loslegen.....	47
5.7 Links zu diesem Kapitel.....	47
6 PGP auf dem PC – Schnelleinstieg für Ungeduldige.....	49
6.1 PGP-Schlüsselverwaltung mit Thunderbird/Enigmail auf dem PC.....	49
6.2 Schlüssel und Widerrufszertifikat sichern.....	51
6.3 Thunderbird für PGP-Nutzung konfigurieren.....	52
6.4 PGP mit Thunderbird nutzen.....	54
6.4.1 Mailversand.....	54
6.4.2 Mailempfang.....	55
6.4.3 Schlüsselbund-Pflege.....	55
6.4.4 Empfängerregeln.....	55
6.5 PGP auf einem weiteren PC einrichten.....	55
6.6 Zusammenfassung.....	56
6.7 Links zu diesem Kapitel.....	56
7 PGP auf dem PC – Ausführlicher Einstieg für Wissbegierige.....	57
7.1 Verwaltung des Schlüsselbunds.....	57
7.1.1 Tools zur Schlüsselverwaltung.....	57
7.1.2 Erzeugung eines neuen Schlüsselpaars.....	60
7.1.2.1 Schlüsselerzeugung mit Enigmail / Thunderbird.....	60
7.1.2.2 Limitationen von Enigmail / Thunderbird.....	61
7.1.2.3 Schlüsselerzeugung mit Gpg4win.....	62
7.1.2.4 Schlüsselerzeugung mit der GPG Suite.....	62
7.1.2.5 Schlüsselerzeugung auf der Kommandozeile.....	62
7.1.3 Das Widerrufszertifikat.....	62
7.1.4 Weitere Benutzer-IDs (Email-Adressen) hinzufügen.....	63
7.1.5 Die wichtigsten PGP-Schlüsseleigenschaften.....	63
7.1.6 Schlüssel exportieren und importieren.....	65
7.1.6.1 Schlüssel sicher aufbewahren.....	65
7.1.7 Konfiguration der Key-Server (Enigmail).....	65
7.1.8 Öffentliche Schlüssel mit Key-Server synchronisieren.....	66
7.1.9 Schlüssel beglaubigen.....	68
7.1.9.1 Verifikation von Schlüssel-Identität und Benutzer-Identität.....	68
7.1.9.2 c't-Kryptokampagne.....	70
7.2 Thunderbird für PGP-Nutzung konfigurieren.....	71
7.2.1 Globale Enigmail-Einstellungen für alle Konten.....	71
7.2.2 Enigmail-Einstellungen für jedes Mail-Konto.....	72
7.2.3 Text-Mails statt HTML-Mails.....	74
7.2.4 Die verschlüsselt versendeten Mails lesbar machen.....	74
7.2.5 Mails automatisch entschlüsseln/überprüfen.....	75
7.3 PGP mit Thunderbird nutzen.....	75
7.3.1 Mailversand.....	75
7.3.2 Mailempfang.....	75
7.3.3 Empfängerregeln.....	76
7.3.4 Mit signierten Mails beginnen .....	76
7.3.5 Pflege des Schlüsselbundes.....	77
7.4 Verschlüsselte Mails auf dem Zweitrechner.....	77

7.5 Zusammenfassung.....	79
7.6 Links zu diesem Kapitel.....	80
<b>8 PGP – Was noch wichtig oder interessant ist.....</b>	<b>82</b>
8.1 Fallstricke beim Einsatz von GnuPG.....	82
8.2 Webmail? Vergiss es!.....	82
8.3 Die Webmail-Alternative – Thunderbird To Go auf dem USB-Stick.....	83
8.4 Verschlüsselte Mails auf iPhone oder iPad.....	84
8.5 Das verschlüsselte Postfach von mailbox.org.....	84
8.6 Zusammenfassung.....	85
8.7 Links zu diesem Kapitel.....	85
<b>9 PGP auf dem Android-Gerät – Schnelleinstieg für Ungeduldige.....</b>	<b>87</b>
9.1 PGP-Schlüsselverwaltung mit APG auf dem Android-Gerät.....	87
9.2 K-@ Mail für PGP-Nutzung konfigurieren.....	89
9.3 PGP auf dem Android-Gerät nutzen.....	89
9.3.1 Versand.....	90
9.3.2 Empfang.....	90
9.3.3 Schlüsselbund-Pflege.....	90
9.4 Zusammenfassung.....	90
9.5 Links zu diesem Kapitel.....	90
<b>10 PGP auf dem Android-Gerät – Ausführlicher Einstieg für Wissbegierige.....</b>	<b>91</b>
10.1 Apps installieren.....	91
10.1.1 Die passenden Android-Apps.....	91
10.1.2 Installation der Apps.....	92
10.2 Verwaltung des PGP-Schlüsselbundes mit APG.....	92
10.2.1 Konfiguration der Schlüsselserver.....	92
10.2.2 Übertragung der Schlüssel auf das Android-Gerät.....	92
10.2.3 Schlüssel-Import in den Schlüsselbund.....	93
10.3 Konfiguration der Mail-App für die Nutzung von PGP.....	94
10.3.1 K-@ Mail Einstellungen.....	94
10.3.2 Text-Mails statt HTML-Mails.....	94
10.4 Nutzung von PGP.....	95
10.4.1 Mailversand.....	95
10.4.2 Empfang.....	95
10.4.3 Schlüsselbund-Pflege.....	96
10.5 Zusammenfassung.....	96
10.6 Links zu diesem Kapitel.....	97
<b>11 PGP/MIME – Und es funktioniert doch.....</b>	<b>98</b>
11.1 PGP/MIME auf dem Android-Gerät.....	98
11.2 PGP/MIME auf dem PC.....	99
11.3 Das verschlüsselte Postfach von mailbox.org.....	99
11.4 Zusammenfassung.....	99
11.5 Links zu diesem Kapitel.....	99
<b>12 Passwortsicherheit und Rechnersicherheit.....</b>	<b>100</b>
12.1 Sichere Passwörter.....	100
12.2 Sicherheit von Rechner, Tablet und Smartphone.....	101
12.2.1 Windows sicher betreiben.....	101
12.2.2 Mac OS X sicher betreiben.....	101
12.2.3 Linux sicher betreiben.....	102
12.2.4 iOS sicher betreiben.....	103
12.2.5 Android sicher betreiben.....	103
12.3 Zusammenfassung.....	104
12.4 Links zu diesem Kapitel.....	104

13 Verschlüsselte Mails – und was noch?.....	105
13.1 Grundsätzliches.....	105
13.1.1 Zero Knowledge.....	105
13.1.2 Open Source.....	106
13.1.3 Was sind Cloud-Dienste?.....	106
13.2 Kommunikationsdienste.....	108
13.2.1 Verschlüsselte „SMS“ mit TextSecure.....	108
13.2.1.1 SMS und Instant Messaging (WhatsApp).....	108
13.2.1.2 WhatsApp unter dem Blickwinkel der Sicherheit und des Privatsphäre-Schutzes.....	109
13.2.1.3 Alternativen zu WhatsApp.....	110
13.2.1.4 TextSecure auf dem Android-Smartphone nutzen.....	111
13.2.1.4.1 Einrichtung.....	111
13.2.1.4.2 Nachrichten-Versand.....	112
13.2.1.4.3 Das Passwort.....	112
13.2.1.4.4 Gerätewechsel.....	112
13.2.2 Abhörsichere Telefonate mit RedPhone.....	113
13.3 Cloud-Dienste.....	113
13.3.1 Groupware-Dienste beim Email-Provider in Anspruch nehmen.....	114
13.3.1.1 Zugriff mit CardDAV und CalDAV.....	114
13.3.1.2 Zugriff auf das zentrale Adressbuch auf dem PC mit Thunderbird.....	115
13.3.1.3 Zugriff auf den zentralen Kalender auf dem PC mit Thunderbird.....	116
13.3.1.4 Zugriff auf eine zentrale Aufgabenliste auf dem PC mit Thunderbird.....	117
13.3.1.5 Zugriff auf das zentrale Adressbuch auf dem Android-Gerät.....	117
13.3.1.6 Zugriff auf den zentralen Kalender auf dem Android-Gerät.....	118
13.3.1.7 Zugriff auf die zentralen Aufgabenlisten auf dem Android-Gerät.....	118
13.3.1.8 CardDAV und CalDAV mit anderen Programmen und Geräten.....	119
13.3.2 Cloud Storage.....	119
13.3.2.1 Viele Cloud Storage Anbieter.....	119
13.3.2.2 Funktionsweise der Synchronisation.....	119
13.3.2.3 Weitere Merkmale der Cloud Storage Dienste.....	120
13.3.2.4 Sicherheitsfragen.....	121
13.3.2.5 Cloud-Speicher mit Ende-zu-Ende-Verschlüsselung.....	121
13.3.2.6 Cloud-Speicher unverschlüsselt und ein zusätzliches Verschlüsselungsprogramm.....	123
13.4 Zusammenfassung.....	126
13.5 Links zu diesem Kapitel.....	128
14 Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?.....	130
14.1 SSL (Secure Socket Layer) und TLS (Transport Layer Security).....	131
14.1.1 Poodle – Sicherheitslücke in SSLv3.....	131
14.2 PFS (Perfect Forward Secrecy).....	133
14.3 HSTS (HTTP Strict Transport Security).....	133
14.4 DANE (DNS-based Authentication of Named Entities).....	133
14.5 Heartbleed.....	135
14.6 Die Qualität der Transport-Verschlüsselung der Mail-Provider prüfen.....	136
14.6.1 starttls.info testet Qualität der Transportverschlüsselung.....	136
14.6.2 tsla.info testet DANE-Unterstützung.....	137
14.6.3 Weitere Tests unter de.ssl-tools.net.....	138
14.6.4 Testergebnisse.....	138
14.7 Zusammenfassung.....	140
14.8 Links zu diesem Kapitel.....	140
15 Glossar.....	142

# Abbildungsverzeichnis

Abbildung 1: Die Mail auf ihrem Transportweg.....	21
Abbildung 2: Thunderbird-Konfiguration für den Postausgang-Server (SMTP).....	28
Abbildung 3: Thunderbird-Konfiguration für den Posteingang-Server (IMAP).....	28
Abbildung 4: Erweiterte Einstellungen von Thunderbird: Konfigurationsvariablen für JavaScript.	35
Abbildung 5: Thunderbird-Plugins: Das Java-Plugin (grau unterlegt) ist deaktiviert.....	35
Abbildung 6: Thunderbird-Einstellungen: Nachladen externer Inhalte deaktivieren.....	36
Abbildung 7: Thunderbird: Werbemail von WEB.DE vor dem Laden der externen Inhalte.....	37
Abbildung 8: Thunderbird: Werbemail von WEB.DE nach dem Laden der externen Inhalte.....	38
Abbildung 9: Asymmetrisch verschlüsselte Kommunikation.....	42
Abbildung 10: Thunderbird: Der Enigmail-Schlüsselbund mit einigen Schlüsseln.....	57
Abbildung 11: Thunderbird: Verwaltung der Add-ons öffnen.....	58
Abbildung 12: Thunderbird nach der Enigmail-Installation: Das neue Enigmail-Menü.....	59
Abbildung 13: Thunderbird: Konten-spezifische Enigmail-Konfiguration.....	59
Abbildung 14: Enigmail: Neues Schlüsselpaar erzeugen.....	60
Abbildung 15: Enigmail: Dialog zum Erzeugen eines neuen Schlüsselpaars.....	60
Abbildung 16: Enigmail: Der Schlüsselbund mit dem neu erzeugten Schlüssel (markiert).....	61
Abbildung 17: Enigmail: Liste der Benutzer-IDs eines Schlüssels.....	63
Abbildung 18: Enigmail: Schlüsselattribute.....	63
Abbildung 19: Enigmail: Den markierten Schlüssel exportieren.....	65
Abbildung 20: Enigmail: Konfiguration der Schlüssel-Server.....	66
Abbildung 21: Enigmail: Optionen für die Synchronisation mit einem Schlüssel-Server.....	67
Abbildung 22: Enigmail: Den markierten Schlüssel unterschreiben/beglaubigen.....	68
Abbildung 23: Enigmail: Definieren des Besitzervertrauens.....	68
Abbildung 24: Enigmail: Globale Einstellungen für den Mailversand.....	71
Abbildung 25: Enigmail: PGP-Einstellungen für ein Mail-Konto.....	72
Abbildung 26: Enigmail: Weitere PGP-Optionen.....	74
Abbildung 27: PGP-Versand-Optionen.....	75
Abbildung 28: Neue Empfängerregel erstellen.....	76
Abbildung 29: Enigmail: Schlüssel-Export.....	77
Abbildung 30: Enigmail: Auch die geheimen Schlüssel exportieren?.....	78
Abbildung 31: Android: K-@ Mail: Nur Zeichensalat – Die verschlüsselte Mail ist nicht lesbar....	92
Abbildung 32: Android-Filetransfer auf dem Mac, Smartphone über USB angeschlossen: Die Schlüssel werden ins Download-Verzeichnis kopiert.....	93
Abbildung 33: Android: Die importierten Schlüssel im APG-Schlüsselbund.....	93
Abbildung 34: Android: APG-Schlüsseldetails: Ein Schlüssel mit drei Benutzer-IDs.....	93
Abbildung 35: Android: K-@ Mail: Kryptographie-Optionen.....	94
Abbildung 36: Android: K-@ Mail: Die Mail kann jetzt gelesen werden.....	95
Abbildung 37: Squeaky Mail: Kryptographie-Optionen: PGP/MIME ist wählbar. (vgl. Abb. 37)...	98
Abbildung 38: TextSecure: Einstellungen.....	111
Abbildung 39: Thunderbird: Ein Remote-Adressbuch erstellen.....	115
Abbildung 40: Thunderbird: Einen Netzwerk-Kalender erstellen.....	116
Abbildung 41: CardDAV-Sync: Neues CardDAV-Konto erstellen.....	117
Abbildung 42: CalDAV-Sync: Neues CalDAV-Konto erstellen.....	118
Abbildung 43: Boxcryptor: Der verschlüsselte Inhalt des verschlüsselten Ordners Boxcryptor.bc	125
Abbildung 44: Boxcryptor: Der entschlüsselte Inhalt des virtuellen Boxcryptor-Laufwerks.....	125
Abbildung 45: Daten in der Dropbox-App verschlüsselt, in der Boxcryptor-App entschlüsselt....	126
Abbildung 46: Poodle-Test mit einem verwundbaren (links) und einem nicht verwundbaren Browser (rechts).....	132
Abbildung 47: starttls.info: Abfrage von secure.mailbox.org (Übersicht).....	136
Abbildung 48: starttls.info: Abfrage von secure.mailbox.org (Detail-Ansicht für ersten Server)...	137



# 1 Sichere Email – warum überhaupt?

## 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 detaillierte 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. Bezahlte Werbung wird von denjenigen, deren Werbung sie auf ihrer Internet-Plattform platziert. Wir Nutzer bekommen die Werbung zu sehen, wenn wir die Plattform der Provider nutzen. Die Plattformen vieler Mail-Provider (WEB.DE, GMX, freenet 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 den Ergebnisseiten unserer Google-Suche. Dazu 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 sofort oder bei Bedarf auszuwerten. 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 Reise durch das Internet ist im Laufe der vergangenen zwei Jahre deutlich besser geworden und macht weiter Fortschritte. Die Mail-Inhalte sind jedoch allermeist unverschlüsselt; die Mails werden unverschlüsselt bei den Providern gespeichert.)
- **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-Mails und virenverseuchte Mails). Oder sie greifen unseren Mail-Provider direkt an und erlangen auf diesem Wege ebenfalls 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 jährten 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. Es ist wie mit dem Geist in der Flasche. Einmal aus der Flasche entwichen kehrt er nie mehr dorthin zurück.

Ein weiterer Grundsatz ist die **Verwendung sicherer Passwörter** für Dienste, bei denen wir uns anmelden müssen (siehe auch Kap. 12.1).

Datensparsamkeit und sichere Passwörter sind allgemeine Sicherheitsmaßnahmen, die unter Anderem auch der Email-Sicherheit zugute kommen. Sie 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 (mehr zur Passwort-Sicherheit in Kap. 12.1).

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 weitere Hinweise zur Email-Sicherheit füllen die Kapitel 3 und 4 dieses Dokuments. Um die Email-Kommunikation komplett abzusichern, müssen die Mails verschlüsselt werden. Dies ist allerdings nicht trivial und ist Thema in den Kapiteln 5 bis 10.

Warum schreibe ich dieses Dokument? Zum großen Teil 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. Google ist Großmeister, wenn es um das Sammeln, Auswerten und geschickte Verknüpfen von Daten geht.

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 Eduard Snowden das Einzige, worauf wir uns heute im Internet noch verlassen können.

Eduard 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.

### 1.3 Scannen von Mails – zu welchem Zweck?

Unverschlüsselte Mails können vom Provider gelesen werden. Vor allem die großen amerikanischen Provider (Google, Microsoft, Yahoo, Facebook) gewinnen aus den gescannten Mails Informationen über die Kommunikationspartner und erstellen daraus und aus Informationen, die sie aus anderen Quellen (Websuche, Smartphone, soziale Netzwerke, Übertragung des Standorts und vieles mehr) beziehen, persönliche Profile ihrer Nutzer. Sie nutzen diese, um dem Nutzer auf ihn zugeschnittene Werbung zu präsentieren (z.B. in den Ergebnissen der Google-Suche). Werbung ist eine sehr wichtige Einnahmequelle dieser Unternehmen. Gerade bei Google und Facebook ist Werbung die Basis des Geschäftsmodells.

Dies hat auch sein Gutes. Dadurch, dass er die Mails lesen kann, kann der Provider z.B. auch Virenschutz für Mails bereitstellen. Verschlüsselte Mails lassen sich nicht auf Viren untersuchen.

Spam-Mails lassen sich nur aus unverschlüsselten Mails herausfiltern. Dies ist für uns Mail-Nutzer sehr wertvoll. Spam-Mails landen so automatisch im Spam-Ordner.

Doch der Mail-Scan geht noch weiter. Anfang August 2014 wurde bekannt, dass sowohl Google als auch Microsoft Mails nach kinderpornographischen Inhalten durchsuchen. Beide Unternehmen haben die betreffenden Personen an die amerikanischen Behörden gemeldet.

Links auf Heise Online zu diesem Thema:

- <http://www.heise.de/newsticker/meldung/Kinderpornographie-Google-bringt-Polizei-auf-die-Spur-eines-Nutzers-2282356.html>
- <http://www.heise.de/newsticker/meldung/Kinderpornographie-Google-verteidigt-E-Mail-Scan-2284919.html>
- <http://www.heise.de/newsticker/meldung/Auch-Microsoft-durchsucht-Mail-Konten-nach-Kinderpornografie-2287862.html>

So ehrenwert und moralisch diese Email-Scans motiviert sein mögen, man fragt sich, wo die Grenze ist. Untersucht Google die Mails auch nach terroristischen Inhalten? Oder nach Drogendelikten? Oder Wohnungseinbrüchen und Autodiebstählen? Oder nach Fahrrad- und Handtaschendiebstählen? Oder wird sogar nach gesellschaftskritischen Inhalten gesucht? Werden Google und Microsoft und die anderen großen Provider (Yahoo, Facebook und Twitter) zur neuen Weltpolizei?

Durch diese Nachrichten sahen sich auch die deutschen Email-Anbieter veranlasst, Stellung zu beziehen und erklärten, dass sie die Mails nicht nach Kinderpornographie, sondern lediglich nach

Viren und Spam durchsuchen würden.

- <http://www.heise.de/newsticker/meldung/Deutsche-E-Mail-Anbieter-durchsuchen-Mails-nicht-nach-Kinderpornos-2288305.html>

Gleichgültig, welche positiven Nebenwirkungen das Durchsuchen von Mails auch haben mag, eines haben wir dabei schon längst verloren: unsere Privatsphäre.

Nach diesen Nachrichten hat am 08. August auch die Bundesdatenschutzbeauftragte dazu Stellung bezogen: „*Die inhaltliche Auswertung von E-Mails stellt zweifelsfrei einen nicht unerheblichen Grundrechtseingriff dar.*“

- <http://www.heise.de/newsticker/meldung/Datenschutzbeauftragte-Scannen-von-E-Mails-ist-Grundrechtseingriff-2289059.html>

Doch damit nicht genug. Lagern unsere Mails unverschlüsselt auf den Servern der Provider, dann steht ihr Inhalt nicht nur den Providern zur Verfügung, sondern auch Hackern, denen es gelingt, in die Server des Providers einzubrechen – seien es nun Internetkriminelle oder die Hacker der NSA, des GCHQ oder des BND.

Außerdem muss man nicht nur bei amerikanischen, sondern auch bei deutschen Providern damit rechnen, dass Behörden im Zuge polizeilicher Ermittlungen per Gerichtsbeschluss die Herausgabe der Email-Kommunikation eines Benutzers verlangen können und dem Provider verbieten, den Benutzer davon in Kenntnis zu setzen.

Es geht also darum, unsere **Privatsphäre zurückzugewinnen!** Was können wir dafür tun?

Wir benötigen als Erstes einen **vertrauenswürdigen und zuverlässigen Provider**. Dieser sollte kein Interesse daran haben, die Inhalte unserer Mails zu scannen, um persönliche Profile seiner Nutzer zu erstellen. Dieser sollte außerdem sein Handwerk verstehen, sodass seine Server, auf denen unsere Mails lagern, möglichst gut vor Hackerangriffen geschützt sind.

Doch auch der beste Provider kann mir nicht garantieren, dass seine Server niemals gehackt werden. Und auch er wird meine Mails bei einer durch Gerichtsbeschluss angeordneten, polizeilichen Anfrage herausgeben müssen.

Wer mehr Privatsphäre will, muss zu recht unbedeutenden Maßnahmen greifen und seine **Mails verschlüsseln**. Die Inhalte verschlüsselter Mails können nicht gelesen werden, nicht vom Provider, nicht von Hackern, nicht von NSA und GCHQ und auch nicht von staatlichen Ermittlungsbehörden. Nur ich selbst kann meine Mails entschlüsseln, solange nur ich selbst den Schlüssel dazu habe.

„**Ich habe doch nichts zu verbergen.**“ wird sich mancher sagen. Dies ist richtig. Aus einer einzelnen Mail oder SMS lassen sich nicht viele Informationen gewinnen. Wird jedoch meine gesamte Email-Kommunikation über Monate und Jahre systematisch gesammelt, gespeichert und ausgewertet, so lässt sich aus vielen unwichtigen Informationen dennoch ein sehr detailliertes, persönliches Profil aggregieren, das nicht nur mein Alter, mein Geschlecht, meine Nationalität und meinen Beruf kennt. Dieses Profil gibt auch Auskunft über meine Hautfarbe, meine Vorlieben, meine politische Gesinnung, meine sexuellen Neigungen usw.

Ein solches Profil von mir, das bei Google oder Facebook oder bei der NSA oder dem GCHQ liegt, ein Profil, das ich selbst nicht verfasst habe und auch nicht korrigieren kann, dies betrachte ich als Verletzung meiner Privatsphäre. Und deshalb würde ich auch gerne die vielen kleinen, unwichtigen Informationshäppchen wie meine Emails verschlüsseln.

## 1.4 Kein Google-Bashing

Auch wenn es im vorigen Kapitel so klingen mag, ich will Google nicht einfach nur als den großen „Internet-Bösewicht“ oder die „Datenkrake“ brandmarken. Viele Internet-Errungenschaften, die wir heute ganz selbstverständlich und zu unserem Vorteil nutzen, haben wir Google zu verdanken. Auch ich schätze und nutze diese Vorteile.

Dies ist nicht nur die Google-Suchmaschine (mit über 90 % Marktanteil in Deutschland) und das Smartphone-Betriebssystem Android (mit ca. 80 % Marktanteil in Deutschland). Der Google Browser Chrome hat nur einen Marktanteil von ca. 20 % allerdings mit steigender Tendenz. Er gilt jedoch (meines Erachtens zu Recht) als der schnellste, sicherste und technologisch fortschrittlichste Browser. Google bietet auch einen sehr ausgereiften und komfortablen Mail-Service und viele weitere Dienste, von denen wir vielleicht manche kennen: YouTube, Google Maps, Google Calendar, Google Drive (ein Cloud-Dienst a la *Dropbox*), Google Plus (ein Soziales Netzwerk wie Facebook). Andere Dienste sind häufig im Hintergrund tätig und deshalb dem normalen Internetnutzer nicht explizit bekannt. Sie stellen uns jedoch – meist ohne dass wir uns dessen bewusst sind – sehr viel Komfort bei der Nutzung des Web bereit.

Gerade durch das große und vielfältige Angebot von kooperierenden Diensten kann Google besonders effizient Daten über die Netzbürger sammeln und miteinander verknüpfen. Es ist heute kaum noch möglich, das Web zu nutzen, ohne dabei Google mit persönlichen Daten zu beliefern.

Ich möchte den Komfort, den mir Google's Dienste bieten, häufig nicht missen. Ich will mir jedoch auch darüber im Klaren sein, welchen Preis ich als Google's Datenlieferant dafür bezahle, um evtl. eine Grenze ziehen zu können, wenn mir der Preis zu hoch ist.

Was für Google gilt, gilt grundsätzlich auch für einige andere Anbieter von Internet-Diensten. Google hat allerdings die Fähigkeit, Daten zu sammeln und sinnvoll zu verknüpfen, am besten perfektioniert.

## 1.5 Die Nachteile der Mail-Verschlüsselung

Der erste Schritt zur sicheren Mail ist die Wahl des richtigen Email-Providers. Der zweite Schritt ist die Einrichtung der Mail-Verschlüsselung. Wer Mail verschlüsselt, muss – neben einigen Unbequemlichkeiten in der Benutzung – einige weitere Nachteile in Kauf nehmen.

- Der Zugriff auf die Mails mit dem Browser (Webmail) ist nicht mehr möglich. Dies ist wohl für viele Nutzer die schwerwiegendste Einschränkung. (Warum die klassische Webmail und die Verschlüsselung sich technisch nicht vereinbaren lassen, wird in Kapitel 8.2 erläutert.) Durch die Nutzung eines Smartphones, das man immer dabei haben kann, kann man jedoch gut auf Webmail verzichten.
- Die Volltextsuche durch die Mail-Inhalte ist nicht möglich. Dieser Nachteil trifft nur diejenigen wirklich, die ihre Mails lange aufbewahren und später noch nach bestimmten alten Mails suchen wollen oder müssen. Die meisten Mails haben nach ein paar Tagen ihre Bedeutung verloren und enden ohne hin in der Vergessenheit oder im Papierkorb. Da viele Nutzer ihre Mails niemals durchsuchen, werden sie diesen Nachteil auch nicht spüren.
- Der Provider kann für verschlüsselte Mails keinen Spam- und Virenschutz bereitstellen. Dies sollte in der Praxis kein Problem darstellen, denn bislang werden Spam und Viren noch unverschlüsselt versendet.

Diese Nachteile sind der Preis, den man für die durch die Verschlüsselung gewonnene Privatsphäre

bezahlt.

Wer heute verschlüsselt, ist noch ein Exot oder ein „Techie“. Verschlüsselung darf allerdings nicht mehr die Ausnahme sein. Es muss zum Standard werden und es muss einfach benutzbar werden. Dazu bedarf es technologischer Weiterentwicklungen, die Verschlüsselung möglichst einfach nutzbar machen. Die Welt wird hier in zwei, drei oder fünf Jahren sicherlich anders aussehen. Jedoch können wir Benutzer heute schon damit beginnen, auch wenn es noch eine kleine Herausforderung ist, dies umzusetzen. Diese lässt sich jedoch meistern. Meine Ausführungen sollen dabei helfen.

## 1.6 Zusammenfassung

In diesem Kapitel habe ich die Gefahren und Risiken für unsere Kommunikation im Allgemeinen und für die Email-Kommunikation im Besonderen aufgezeigt. Wenn wir kommunizieren, erzeugen wir dauernd Datenspuren im Netz. Damit steht die Sicherheit der Kommunikation und der Schutz unserer Privatsphäre auf dem Spiel. Interesse an unseren Daten haben große Internet-Konzerne (Google, Facebook etc.), Cyberkriminelle und die Geheimdienste. NSA und GCHQ haben sich dabei in besonderer Weise hervorgetan.

Um unsere Kommunikation abzusichern und die Privatsphäre möglichst gut zu schützen, müssen wir ein paar Unbequemlichkeiten in Kauf nehmen. Die Email-Kommunikation steht in diesem Dokument im Fokus. Um diese zu sichern, müssen wir ...

- uns einen möglichst sicheren Email-Provider suchen und ggf. den Email-Provider wechseln,
- im Email-Client die richtigen Konfigurationseinstellungen vornehmen,
- unser Nutzungsverhalten auf den Prüfstand stellen und ggf. verbessern, damit unser System nicht infiziert und kompromittiert wird
- und schließlich unsere Mails verschlüsseln, sodass nur der Absender und der Empfänger die Mail lesen können.

Ich habe die Problematik der Email-Scans durch die Provider dargestellt. Durch die Scans kann der Provider Spam-Mails und virenverseuchte Mails herausfiltern. Er kann aber auch nach „verdächtigen“ Inhalten suchen und die betreffenden Personen bei den Behörden anzeigen. Selbst wenn die Scans ihre guten Seiten haben mögen, so sind sie doch sehr fragwürdig, da sie die Privatsphäre der Mail-Nutzer massiv verletzen. Nur die Verschlüsselung der Mails kann dies verhindern.

Schließlich habe ich noch die Nachteile der Mail-Verschlüsselung aufgeführt. Die größte Einschränkung ist wohl, dass man mit dem Browser nicht auf verschlüsselte Mails zugreifen kann. Webmail und Mail-Verschlüsselung sind technisch nicht vereinbar.

## 1.7 Links zu diesem Kapitel

- „[Im NSA-Skandal ist ein langer Atem gefragt!](http://www.heise.de/newsticker/meldung/Kommentar-Im-NSA-Skandal-ist-ein-langer-Atem-gefragt-2214717.html)“, 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](http://www.heise.de/newsticker/meldung/Kommentar-zum-NSA-Skandal-Es-ist-Zeit-die-)“, 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>
- Links auf Heise Online zum Thema: Scannen von Mails durch Provider:
  - <http://www.heise.de/newsticker/meldung/Kinderpornographie-Google-bringt-Polizei-auf-die-Spur-eines-Nutzers-2282356.html>
  - <http://www.heise.de/newsticker/meldung/Kinderpornographie-Google-verteidigt-E-Mail-Scan-2284919.html>
  - <http://www.heise.de/newsticker/meldung/Auch-Microsoft-durchsucht-Mail-Konten-nach-Kinderpornografie-2287862.html>
  - <http://www.heise.de/newsticker/meldung/Deutsche-E-Mail-Anbieter-durchsuchen-Mails-nicht-nach-Kinderpornos-2288305.html>
  - <http://www.heise.de/newsticker/meldung/Datenschutzbeauftragte-Scannen-von-E-Mails-ist-Grundrechtseingriff-2289059.html>

## 2 Grundlegende Konzepte

Einige Begriffe, die ich in den folgenden Ausführungen verwende, sind dem IT-Experten geläufig. Dieses Dokument richtet sich jedoch insbesondere auch an den IT-Laien. Deshalb will ich die Begriffe *Email-Daten und -Metadaten*, *Client und Server*, *Protokoll* sowie *Verschlüsselung* erläutern. Ich erlaube mir dabei einige Vereinfachungen, um die Konzepte besser herauszuarbeiten. Die Experten unter meinen Lesern mögen mir das verzeihen. Sie können dieses Kapitel auch selektiv lesen oder überspringen und mit Kapitel 3 fortfahren.

### 2.1 Die Daten und Metadaten allgemein

Metadaten sind Daten über Daten.

Ein Beispiel sei dieser Satz:

„Gestern fuhr Mäxchen mit dem ICE von München nach Berlin.“

Die Aussage dieses Satzes ist nun das Datum, die eigentliche Information. Was kann man sich nun unter den Metadaten vorstellen?

Die Metadaten beschreiben z.B.

- wer den Satz gesagt (oder geschrieben) hat (Autor),
- wo der Satz gesagt wurde (Ort),
- wann er gesagt wurde (Zeitpunkt),
- mit welcher Lautstärke er ausgesprochen wurde,
- wer dem Sprecher zugehört hat,
- etc.

In der Informationstechnologie werden Daten verarbeitet, gespeichert und über Netzwerke übertragen. Dabei fallen immer auch viele Metadaten über diese Daten an.

So verhält es sich auch bei einer besonderen Art von Daten, mit denen wir uns in diesem Dokument beschäftigen: den Emails.

### 2.2 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 idealerweise 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.

## 2.3 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 (jedoch nicht die Metadaten) 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 in den Kapiteln 3 und 4 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 in den Kapiteln 5 bis 11 dieses Dokuments. Die Verantwortung dafür liegt ausschließlich bei den beiden Kommunikationspartnern.

## 2.4 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.

## 2.4.1 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, das genau auf die Erfordernisse des betreffenden Dienstes zugeschnitten ist (siehe Kap. 2.5).

## 2.4.2 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 Webbrower 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-Server des Absenders schickt die Mail weiter an den Mail-Server des Providers des Empfängers. 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 die Mail-Weiterleitung als Dienst an (Server) und Ersterer nimmt diesen Dienst in Anspruch (Client).

## 2.4.3 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. (Der Wechsel

zwischen Request und Response ist nur das Grundmuster der Kommunikation. Tatsächlich wird dieses Grundmuster in der Kommunikation häufig variiert oder durchbrochen. Darauf will ich an dieser Stelle aber nicht näher eingehen. Für das Verständnis dieses Textes ist die Kenntnis des Grundmusters der Kommunikation ausreichend.)

**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.

## 2.5 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 verstndigen knnen, sind fr jeden Dienst eine Reihe verschiedener Arten von Nachrichten (Request-Nachrichten und Response-Nachrichten) definiert. Wrde ein Client einem Server eine Nachricht schicken, die dieser nicht versteht, kann der Server auch keine Antwort senden. Er wrde im besten Fall mit einer Fehler-Nachricht antworten. Selbst dies ist mglicherweise nutzlos, wenn der Client die Fehler-Nachricht nicht versteht. Es geht bei der Kommunikation zwischen Client und Server also nicht ohne Vereinbarungen, an die sich beide Seiten halten mssen. Eine solche Nachrichten-Vereinbarung fr bestimmte Anwendungen/Dienste nennt man ein Protokoll.

**Ein Protokoll definiert also die Nachrichten (und Nachrichten-Formate) fr einen bestimmten Dienst, die vom Client zum Server und vom Server zum Client bertragen werden drfen.** Nur so knnen sich beide Seiten verstndigen 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 mssen das jeweilige Protokoll kennen, um diese (die Server und Clients) korrekt zu implementieren. Fr die normalen Anwender spielt die genaue Kenntnis der Protokolle keine Rolle.

Fr die vielen unterschiedlichen Dienste gibt es nun jeweils ein Protokoll, das festlegt, wie sich die Clients und Server des betreffenden Dienstes verstndigen. Um dies ein wenig konkreter zu machen, habe ich beispielhaft einige gngige Protokolle aufgefhrt, die der Internetnutzer (vielleicht ohne es zu wissen) tglich nutzt.

- **HTTP (Hypertext Transfer Protocol)** ist das Protokoll zur bertragung von verlinkten Webseiten (Hypertext). Der HTTP-Client (auch Web-Client) ist der Webbrower (*Chrome*, *Firefox* etc.), den wir tglich zum Surfen benutzen. Der HTTP-Server (auch Web-Server) stellt die Hypertext-Seiten im Web zur Verfgung. Er wird in der URL-Zeile des Browsers angezeigt.
- **FTP (File Transfer Protocol)** ist das Protokoll zur Dateibertragung. 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 bertragen und vieles mehr. Z.B. kann man auch Dateien und Ordner auf dem fernen FTP-Server anlegen, lschen 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 Webbrower 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*, *Apple Mail* 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. Ein Mail-Client wie *Thunderbird* 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. Auf dem Mail-Server wird die Mail normalerweise nach erfolgreicher Übertragung gelöscht; sie ist dann nur noch auf dem Client verfügbar. 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. Solange man sie nicht explizit löscht, bleiben die Mails auf dem IMAP-Server gespeichert. So ist auch der Zugriff mit mehreren Clients auf verschiedenen Geräten möglich. Mit IMAP 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.

Ein Mail-Server muss also 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 (**S**imple **N**etwork **M**anagement **P**rotocol) ein Protokoll zur Verwaltung von Computer-Netzwerken.

## 2.6 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:

## 2.6.1 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 Chiffrat) in die Datei *gedicht-chiffriert.bin* geschrieben. Das Chiffrat 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@??m@?????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 sein kann. 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.

## 2.6.2 Transport-Verschlüsselung vs. Daten-Verschlüsselung

Sind Daten im Internet auf Reisen, 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 sind dabei nur unzugänglich, solange sie unterwegs sind. 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 verschlüsselten Kanal zu übertragen,
- unverschlüsselte Daten über einen verschlüsselten Kanal zu übertragen,
- verschlüsselte Daten über einen unverschlüsselten Kanal zu übertragen oder
- unverschlüsselte Daten über einen unverschlüsselten Kanal zu übertragen.

Das Beispiel im vorigen Kapitel 2.6.1 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.

### 2.6.3 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 hinein fließen und nachdem sie wieder herausfließen, sind die Daten aber sehr wohl einsehbar, wenn die Daten selbst nicht verschlüsselt sind.

Transport-Verschlüsselung kommt heute häufig zum Einsatz, zum Beispiel im Web, bei der Kommunikation zwischen Webbrower 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. 2.4.3) 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 Webbrower und Web-Server wird durch das HTTP-Protokoll (siehe Kap. 2.5) definiert (**Hypertext Transfer Protokoll**). **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 Brower durch das Symbol eines Schlosses an, dass er die Daten mit dem Web-Server über einen verschlüsselten Kanal austauscht. Heute läuft auch jede Google-Suche über einen verschlüsselten Übertragungskanal. Google hat die Web-Suche wie auch die anderen Dienste auf durchgängige Verwendung von Transportverschlüsselung umgestellt.

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. 3.1). 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 3.

## 2.6.4 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 2.6.1 musste sich der Benutzer das Passwort (dieses diente als 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. 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 Ende-zu-Ende-Verschlüsselung bekommt der Provider nur verschlüsselte Daten, zur Speicherung oder zur Weiterleitung anvertraut. Der Provider kennt nur die Metadaten, über die Daten selbst hat er auf Grund ihres verschlüsselten Zustandes keine „Wissen“. Deshalb spricht man in diesem Fall (nur in Bezug auf die Daten) auch von **Zero Knowledge**. Dies ist ein Merkmal eines sicheren Dienstes. Hat der Provider keine Kenntnis von den Daten, so kann er sie nicht lesen oder verarbeiten. Auch wenn die Daten gestohlen würden oder kraft eines Gerichtsbeschluss herausgegeben werden müssten, so wären die Daten immer noch sicher, da auch der Dieb oder die staatliche Behörde mit den verschlüsselten nicht anfangen können, solange sie nicht über den Schlüssel dazu verfügen.

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 5 erläutere ich dieses Thema detailliert und versuche die Einstieg in den Versand und Empfang verschlüsselter Mails zu erleichtern.

## 2.7 Zusammenfassung

Bevor wir zu den praktischen Fragen sicherer Email kommen, erläutert dieses Kapitel noch einige grundlegende Konzepte, die zur Lektüre der nachfolgenden Kapitel verstanden sein will.

Hier ging es um die Unterscheidung zwischen **Daten und Metadaten** einer Mail. Es ging außerdem um die **Zuständigkeiten**. Wer ist für die sichere Mail-Übertragung verantwortlich (Absender, Absender-Provider, Empfänger-Provider und Empfänger) und wer muss sich um die Mail-Verschlüsselung kümmern (Absender und Empfänger)?

Dieses Kapitel liefert auch einige grundlegende Konzepte der Informationstechnologie, die nicht nur für die Email-Kommunikation wichtig sind. Dabei ging es um die **Kommunikationsrollen Client und Server** und um die (Anwendungs-)**Protokolle**, die es erst ermöglichen, dass ein Client und ein Server sinnvoll miteinander Daten austauschen können. Einige gängige Protokolle (HTTP, FTP, SMTP, POP, IMAP) sollten veranschaulichen, was überhaupt die Aufgabe eines Protokolls ist.

Schließlich habe ich noch gezeigt, was man sich unter **Verschlüsselung** vorzustellen hat, und dabei auch die **Transport-Verschlüsselung** von der **Daten-Verschlüsselung** abgehoben. Beide Arten der Verschlüsselung haben für den sicheren Email-Verkehr – aber nicht nur für diesen – eine große Bedeutung und werden uns in diesem Dokument immer wieder begegnen.

## 2.8 Wer mehr wissen will ...

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 technisch tiefergehend 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.

## 2.9 **Links zu diesem Kapitel**

- c't-Sonderheft mit dem Titel „Sichere E-Mail – NSA aussperren – Privates schützen“:  
<http://shop.heise.de/katalog/ct-wissen-sichere-e-mail>

## 3 (Möglichst) Sicherer Versand unverschlüsselter Mails

Um Emails wasserdicht abgesichert zu versenden (dies wäre sicherer als ein versiegelter Brief), müssen wir sie verschlüsseln (siehe Kap. 5 Bis 10). 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 hohe Sicherheit – und damit die Minimierung des Risikos der Kompromittierung der Mails – kann man aber auch erreichen, wenn man unverschlüsselte Mails über **sichere, d.h. verschlüsselte 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. 3.1). Für diese Sicherheitsstufe ist die **Wahl des richtigen Mail-Providers** von zentraler Bedeutung (siehe Kap. 3.3). Jedoch auch der Benutzer muss die **richtigen Einstellungen im Email-Client** vornehmen.

Die Transportwege können verschlüsselt werden, die Mail-Inhalte bleiben unverschlüsselt. In diesem Fall ist eine Mail nach wie vor nicht mit einem Brief, sondern mit einer Postkarte zu vergleichen, die idealerweise jedoch nur von vertrauenswürdigen Kurieren transportiert wird.

Ich (Absender) schreibe eine 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.

### 3.1 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 dies kurz mit einem Bild erläutern: Ein *verschlüsselter Übertragungskanal* kann mit einer undurchsichtigen Röhre (oder mit einer nicht abhörbaren Leitung) verglichen werden (siehe auch Kap. 2.6.3). 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 (**S**ecure **S**ocket **L**ayer) bzw. durch das neuere TLS (**T**ransport **L**ayer **S**ecurity) 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. 3.2 und 14.

Bei Emails sichert ein verschlüsselter Kanal immer nur eine Teilstrecke des gesamten Übertragungsweges vom Absender zum Empfänger. (Der gesamte Übertragungsweg einer Mail besteht aus drei Teilstrecken, s. u.) Mit verschlüsselten Kanälen kann keine Ende-zu-Ende-Verschlüsselung realisiert werden, bei der der Inhalt der Mail auf dem gesamten Übertragungsweg vom Absender bis zum Empfänger verschlüsselt ist (mehr dazu in Kap. 5).

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 sind in Abbildung 1 dargestellt und werden in den folgenden Abschnitten beschrieben.

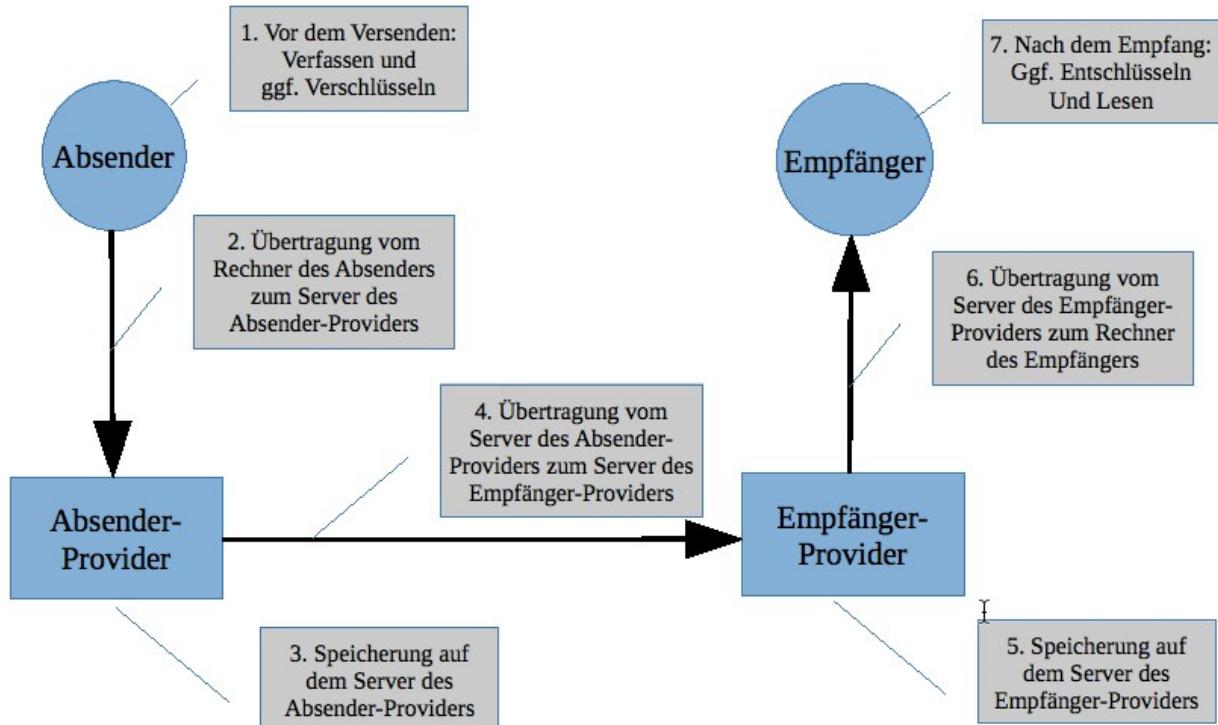


Abbildung 1: Die Mail auf ihrem Transportweg

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 5), 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 meinen Zugangsdaten (Benutzername und 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 nochmals 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 per SMTP (Simple Mail Transfer Protocol) zwischen den Providern muss über einen verschlüsselten Übertragungskanal erfolgen. Dies 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 GMX, 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.

## 3.2 Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?

In den vorangehenden Kapiteln habe einfach von verschlüsselten Übertragungskanälen gesprochen und dabei impliziert, dass diese sicher sind und nicht aufgebrochen oder gar umgeleitet werden können. Allerdings bieten verschlüsselte Übertragungskanäle 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. 14.1). Das veraltete SSL-Protokoll sollte gar nicht mehr verwendet werden. Die Unterstützung für SSL ist auch die Ursache für die Anfälligkeit für sog. Poodle-Angriffe (siehe Kap. 14.1.1).
- Unterstützung von PFS: Mit diesem Verfahren lässt sich die nachträgliche Entschlüsselung von aufgezeichneter, verschlüsselter Kommunikation verhindern. (siehe Kap. 14.2)
- 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. 14.3)
- Unterstützung von DANE: Dieses noch recht neue Verfahren erschwert die Korrumperung von Zertifikaten, auf denen die Transport-Verschlüsselung basiert. (siehe Kap. 14.4)
- Abgesichert gegen Heartbleed-Attacken (siehe Kap. 14.5). Dabei handelt es sich um Angriffe, die eine Sicherheitslücke in der OpenSSL-Software-Bibliothek ausnutzen. Die Lücke ist mittlerweile geschlossen. Die Mail-Provider sollten dagegen gefeit sein.

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 14 näher erläutert.

An dieser Stelle präsentiere ich nur die Testergebnisse. Die Beschreibung der Tests und das Zustandekommen dieser Ergebnisse sind in Kapitel 14.6 ausführlich beschrieben.

Email-Domain	Provider	Heartbleed-Verwundbarkeit	Poodle-Verwundbarkeit	PFS-Unterstützung	DANE-Unterstützung (tlsa.info)	Score (starttls.info)
gmail.com	Google	nein	ja	ja	nein	90,6 %
outlook.com	Microsoft	nein	ja	ja	nein	90,6 %
icloud.com	Apple	nein	ja	ja	nein	83,2 %
yahoo.com	Yahoo	nein	ja	ja	nein	90,6 %
web.de	WEB.DE	nein	ja	ja	nein	90,6 %
gmx.net	GMX	nein	ja	ja	nein	90,6 %
freenet.de	Freenet	nein	ja	ja	nein	Error: Could not connect
telekom.de	Telekom	nein	ja	ja	nein	48,8 %
mykolab.com	MyKolab	nein	nein	ja	nein	92,0 %
posteo.de	Posteo	nein	ja	ja	ja	Error: Connection rejected

jpberlin.de	JPBerlin	nein	nein	ja	teilweise	71,8 %
mailbox.org	mailbox.org	nein	nein	ja	ja	83,4 %
secure.mailbox.org	mailbox.org	nein	nein	ja	ja	94,8 %
mail.de	mail.de	nein	ja	ja	ja	90,6 %

Die Probleme mit *Heartbleed* scheinen mittlerweile bei den getesteten Providern behoben zu sein. Die PFS-Unterstützung ist bei allen gegeben. Auf *Poodle* (verursacht durch die unnötige SSL-Unterstützung) haben in den zwei Wochen seit Bekanntwerden der Lücke nur einige Pioniere reagiert und SSLv3 schon abgeschaltet. Auch die DANE-Unterstützung ist noch die Sache von einigen Pionieren und es sind weitgehend dieselben, die auch schnell auf *Poodle* reagiert haben. Ich nehme an, dass die „roten Flecken“ in der Poodle-Spalte bei *Posteo* und *mail.de* im Laufe des November 2014 verschwinden werden. Die großen Provider zeigen sich durch die Bank tendenziell etwas träger in ihrer Bereitschaft, auf Sicherheitslücken zu reagieren und technische Innovationen aufzugreifen.

Aus dieser Übersicht kristallisieren sich die Pioniere in Sachen Email-Sicherheit klar heraus: *MyKolab*, *Posteo*, *JPBerlin*, *mailbox.org* und *mail.de*.

Ein weiterer Test in einigen Monaten wird weitere Erkenntnisse bringen, wie sich die Provider weiterentwickeln.

### 3.3 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. 2.8.

#### 3.3.1 Auswahlkriterien

- Der Mail-Provider sollte zur Transport-Verschlüsselung die neueste TLS-Version 1.2 unterstützen (Ältere Versionen werden aus Gründen der Abwärtskompatibilität meist zusätzlich unterstützt: TLSv1, TLSv1.1). Das veraltete Protokoll SSL sollte nicht mehr unterstützt werden; es hat zu großen inzwischen bekannte Sicherheitslücken, die recht einfach ausgenutzt werden können.
- Der Mail-Provider sollte PFS unterstützen (siehe Kap. 3.2).
- Der Mail-Provider sollte HSTS auf seiner Website verwenden (siehe Kap. 3.2).
- Der Mail-Provider sollte kein Interesse an den Inhalten meiner Mail haben und nicht mit ihrer Auswertung Geld verdienen.
- 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.
- Der Mail-Provider sollte idealerweise ein deutscher Anbieter 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. 3.2).
8. Der Mail-Provider sollte einen guten Score auf <https://starttls.info> aufweisen.
9. Bestimmte Sicherheitsfeatures, die andere Anbieter nicht haben
10. Spamschutz
11. Virenschutz
12. Weitere Services wie Groupware-Dienste (zentrale Verwaltung von Kontakten, Kalender, Aufgabenlisten) (siehe Kap. 13.3.1) und Cloud-Speicher (siehe Kap. 13.3.2). Diese Dienste werden heute von sehr vielen Email-Providern angeboten. Sie müssen ebenfalls gut gesichert sein.
13. Preis der Leistungen
14. Nachhaltigkeit, Nutzung von Ökostrom
15. Professioneller Webauftritt

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, seit Juli 2014 gehört auch *mail.de* dazu. Vermutlich wird sie im Laufe der folgenden Monate auch von weiteren Providern implementiert.

### 3.3.2 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 des 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*, *mailbox.org* und *mail.de* 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:

- Sicherer Mail-Alias, bei dem die Mail zwischen den Providern garantiert verschlüsselt übertragen wird (Beschreibung in Kapitel 7 und unter <https://mailbox.org-mails-definitiv-sicher-versenden/>)
- Verschlüsseltes Postfach, bei dem unverschlüsselt eingehende Mails sofort nach dem Eingang vom Provider verschlüsselt werden (Beschreibung in Kapitel 8.5 und unter <https://mailbox.org/ihr-e-mail-postfach/#Datenschutz>)

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

### 3.4 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* umziehen.

Ein anderes (unrealistisches) Szenario: Ich bleibe bei Google und verschlüssle 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 5 und den darauf folgenden Kapiteln.

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 lag, habe ich ebenfalls zu [mailbox.org](http://mailbox.org), dem Provider meines Vertrauens transferiert, ebenso das Adressbuch und die Aufgabenlisten (siehe Kap. 13.3.1).

## 3.5 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 (Oktober 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.

### 3.5.1 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. 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 (**S**imple **M**ail **T**ransfer **P**rotocol): 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-Empfang mit IMAP (**I**nternet **M**essage **A**ccess **P**rotocol): In den Konten-Einstellungen muss beim Postausgangsserver (IMAP) als Verbindungssicherheit entweder SSL/TLS oder besser STARTTLS eingetragen sein. (Abb. 3)
3. Prüfung für den Mail-Empfang mit POP3 (**P**ost **O**ffice **P**rotocol, **V**ersion **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 für POP sind analog zu den IMAP-Einstellungen vorzunehmen und werden nicht in einer eigenen Abbildung gezeigt.)

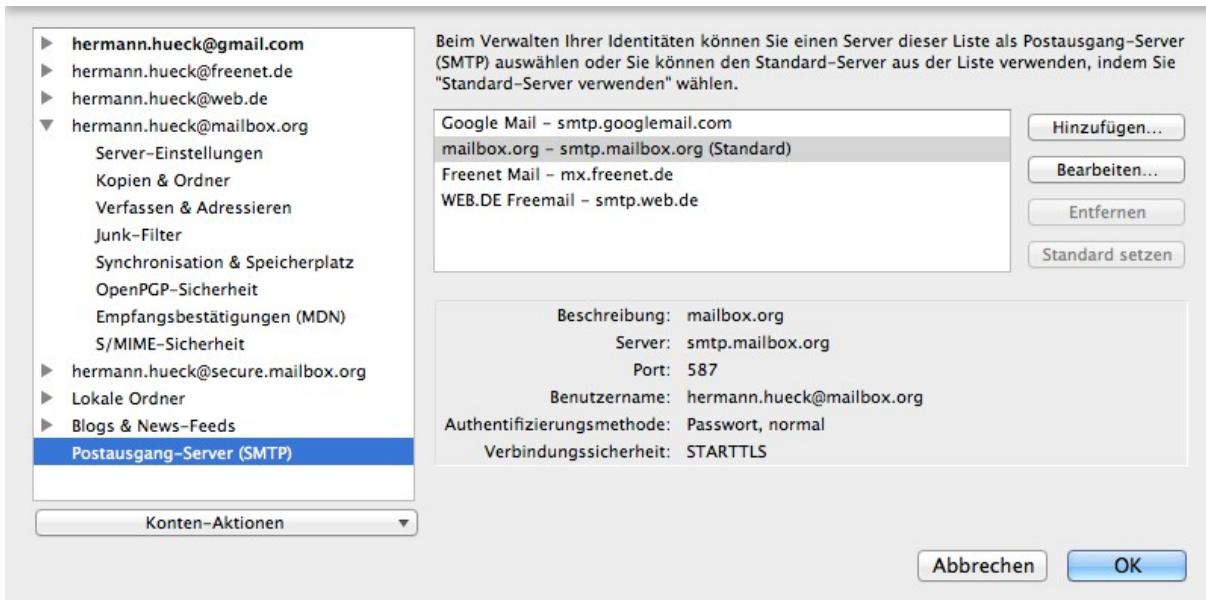


Abbildung 2: Thunderbird-Konfiguration für den Postausgang-Server (SMTP)

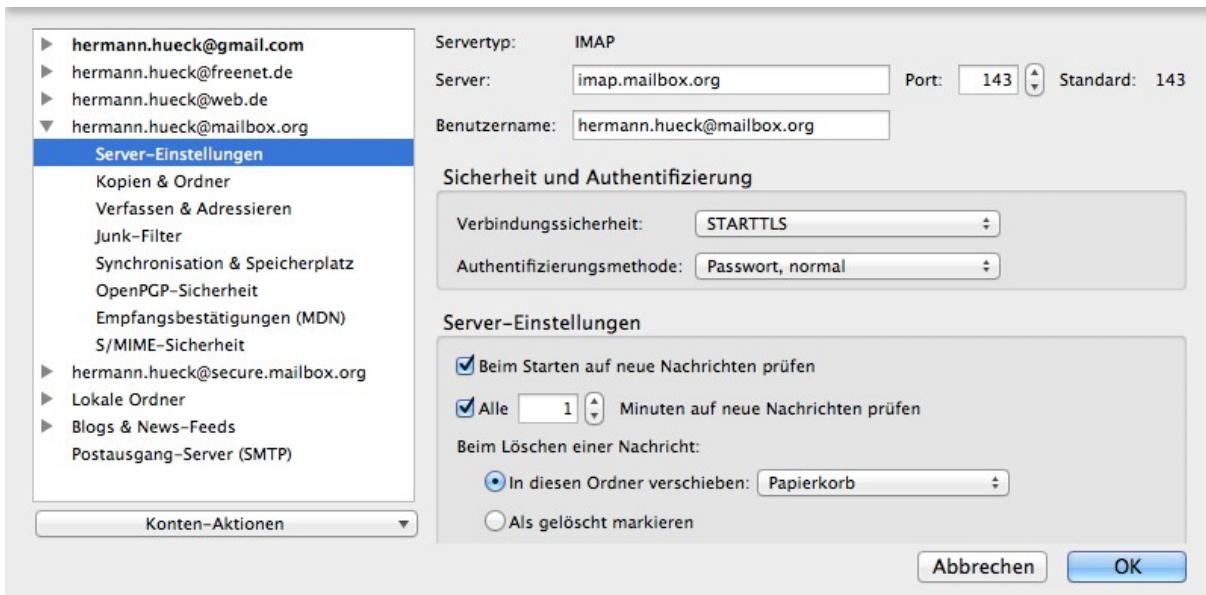


Abbildung 3: Thunderbird-Konfiguration für den Posteingang-Server (IMAP)

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 wäre für jeden einsehbar. Diese Einstellung funktioniert allerdings in den meisten Fällen nicht mehr, da die unverschlüsselte Übertragung vom Mail-Provider 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](http://www.thunderbird-mail.de/wiki/E-Mail-Konto_einrichten)

### **3.6 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. Dies ist heute eher unüblich geworden. Heute wird zum Zugriff auf Mails fast nur noch IMAP verwendet.

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.

Das Risiko, die Mails zu verlieren, ist bei IMAP ebenfalls viel geringer. Selbst wenn der Rechner durch einen Festplattencrash unbenutzbar wird, sind die Mails, die auf dem Server des Providers lagern, sicher verwahrt. Man kann jederzeit auch mit einem anderen Rechner darauf zugreifen.

Auf Tablets und Smartphones steht der Mail-Client *Thunderbird* nicht zur Verfügung. Es gibt jedoch diverse Mail-Apps für Android und iOS. Die Auswahl ist groß. Das IMAP-Protokoll wird von fast allen unterstützt (siehe c't-Sonderheft „Sichere Email“, Seite 50).

Hat man die Absicht, seine Mails später auch noch mit PGP zu verschlüsseln, wird die Auswahl der Mail-Apps viel kleiner.

Unter iOS steht nur *iPGMail* zur Verfügung (siehe Kap. 8.4).

PGP-Unterstützung gibt es unter Android gegenwärtig (2014) nur für *K-9 Mail*, *K-@ Mail*, *Kaiten* oder *Squeaky Mail* in Frage (siehe Kap. 10.1.1). Legt man außerdem Wert auf die Unterstützung des modernen Verschlüsselungsformates PGP/MIME, so bleibt *Squeaky Mail* als einziger Kandidat unter Android übrig (dies ist der Stand der Dinge im Oktober 2014).

Die Mail-App ist grundsätzlich mit denselben Konfigurationseinstellungen einzurichten wie *Thunderbird* (siehe Kap. 3.5.1). 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.

### **3.7 Sicherer Mail-Alias bei [mailbox.org](http://mailbox.org)**

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

[max.mayer@secure.mailbox.org](mailto: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. 3.1, 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 bei der Übertragung von der

NSA oder von anderen Internet-Bösewichtern abgefangen und gelesen werden.

Schickst du eine Mail an mich, ist die sichere Mail-Adresse [hermann.hueck@secure.mailbox.org](mailto: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](mailto: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 ist die Übertragung unverschlüsselt.

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

### 3.8 Zusammenfassung

Wir haben uns in diesem Kapitel den Übertragungsweg einer Mail vom Absender zum Empfänger angesehen. Auf diesem Weg wird die Mail über drei Teilstrecken transportiert (Absender → Absender-Provider, Absender-Provider → Empfänger-Provider, Empfänger-Provider → Empfänger). Die Übertragung auf den Teilstrecken muss durch Transport-Verschlüsselung gesichert sein, damit die Mails auf keiner der Teilstrecken kompromittiert (abgefangen, gelesen oder sogar verfälscht) werden kann.

Wir haben gesehen, dass es Transport-Verschlüsselung unterschiedlicher Qualität gibt und auch einige Qualitätsmerkmale dafür aufgestellt. Zuständig für eine qualitativ hochwertige Transport-Verschlüsselung sind in erster Linie die Mail-Provider. Die Qualität der Transport-Verschlüsselung kann ein wichtiges Kriterium für die Auswahl des richtigen Email-Providers sein. Doch auch andere Kriterien haben wir uns angesehen, um unter den Email-Providern den geeigneten auszuwählen.

Doch auch wir Mail-Nutzer müssen die richtigen Einstellungen in unserem Mail-Client vornehmen, damit (bei den Protokollen SMTP, IMAP und POP) die optimale Verschlüsselung bei der Kommunikation mit dem Provider genutzt wird. Dies wurde am Beispiel des Mail-Client *Thunderbird* gezeigt, gilt jedoch grundsätzlich für alle Mail-Clients in gleicher Weise.

Um auch auf dem Smartphone Zugriff auf die Mails zu haben, ist der Zugriff auf die Mails auf dem PC und auf dem Smartphone mit IMAP einzurichten. Das alte POP ist nur dann sinnvoll, wenn man den Mailzugriff nur auf einem einzigen Gerät benötigt. Man muss sich auf dem Smartphone eine Mail-App installieren und die Einstellungen für SMTP und IMAP analog zu den Thunderbird-Einstellungen vornehmen.

### 3.9 Links zu diesem Kapitel

- 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 Email-Provider: <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](http://www.thunderbird-mail.de/wiki/E-Mail-Konto_einrichten)

## 4 Tipps für die sichere Mail-Nutzung

In diesem Kapitel geht es um der sichere Nutzung des Mail-Client im täglichen Gebrauch. Einerseits sind sinnvolle Einstellungen vorzunehmen, andererseits soll das Bewusstsein des Nutzers geschärft werden, sodass er nicht ungewollt Informationen von sich preisgibt oder unbedacht seinen Rechner mit einem Schadprogramm infiziert.

### 4.1 Rechnersicherheit

Gehen wir davon aus, dass wir einen tauglichen Mail-Provider gewählt haben und dass dieser einen guten Job macht. Die in Kap. 3.5 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 3.1) 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 die Viren-Signaturen aktuell gehalten werden.

Den Rechner aktuell zu halten, dient nicht nur dazu, die Kompromittierung der Mails zu verhindern, sondern es ist eine allgemeine Sicherheitsmaßnahme. Die Rechnersicherheit ist für verschiedene Betriebssysteme in Kap. 12.2 beschrieben.

### 4.2 Thunderbird-Updates

Jedes Computerprogramm enthält Fehler und Sicherheitslücken. Bei Programmen, die in erster Linie zur Kommunikation über das Internet eingesetzt werden, sind diese besonders kritisch. Werden neue Mängel entdeckt, dann werden sie vom Hersteller des Programms hoffentlich schnell behoben. Updates bringen die neuen fehlerbereinigten Programmversionen auf meinen Rechner.

Dies gilt natürlich auch für die Email-Clients, in unserem Fall für *Thunderbird*. Dieser Email-Client 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>

Nutzt man einen anderen Email-Client (*Outlook*, *Apple Mail*, *Pegasus Mail* etc.), so muss dieser selbstverständlich auch aktuell gehalten werden.

### 4.3 Gefahren in Mail-Anhängen abwenden

Mit einem virenfreien Rechner und einem aktuellen *Thunderbird* sind wir allerdings noch nicht ganz auf der sicheren Seite. Es gibt weitere Gefahrenstellen, derer wir uns bewusst sein sollten.

Mail-Anhänge können Viren oder andere Schadprogramme enthalten. Ob ein Anhang infiziert ist, ist in der Regel nicht zu erkennen.

Doch dabei hilft ein aktueller **VirensScanner**. Dieser muss natürlich so eingestellt sein, dass er die

Emails überprüft. Wenn der Benutzer einen infizierten Mail-Anhang zu öffnen oder zu speichern versucht, dann sollte der VirensScanner diese Aktion verhindern und den Benutzer warnen.

Mit einem aktuellen VirensScanner lassen sich so die meisten Viren in Mail-Anhängen unschädlich machen. Eine Garantie für Virenfreiheit ist dies allerdings nicht. Einen ganz neuen Virus erkennt der VirensScanner möglicherweise nicht oder erst am nächsten Tag. Da kann es jedoch schon zu spät sein.

Dem verbleibenden Restrisiko kann man nur ein gesundes **Misstrauen gegenüber Mail-Anhängen** entgegensetzen. Idealerweise öffnet man nur Anhänge, die man erwartet.

Diese Verhaltensregel ist allerdings nicht sehr praxisnah. Man kann die Regel etwas abmildern: man öffnet nur Anhänge von bekannten Absendern. Doch bereits hier begibt man sich möglicherweise schon aufs Glatteis, denn die Absenderadresse einer Mail lässt sich mit ein wenig technischem Know-how recht einfach fälschen.

Eine gut gemachte, infizierte Mail versucht natürlich vertrauenswürdig auszusehen. Sie fälscht typischerweise den Absender, kommt dann beispielsweise von *telekom.de* und enthält im Anhang eine PDF-Rechnung. Man öffnet die Rechnung, weil man nachsehen will, wie viel man zu bezahlen hat, und genau damit ist man dem Betrug auf den Leim gegangen und hat seinen Rechner infiziert, falls der VirensScanner nicht vorher warnt.

Also sollte man nicht gleich auf den Anhang klicken, sondern zuerst überlegen, ob die Rechnung überhaupt plausibel ist:

- Eine Rechnung von der Telekom? Ich habe meinen Anschluss doch bei Vodafone!
- Die Rechnung jetzt am 20. des Monats? Die kommt doch sonst immer ungefähr am 10.!
- Wieso kommt die Rechnung denn jetzt per Mail? Normalerweise muss ich mich doch auf der Website des Providers anmelden und die Rechnung dann herunterladen!

Mehr Information zu gefährlichen Email-Anhängen findet man auf den Seiten des Heise-Verlags unter <http://www.heise.de/security/dienste/Dateianhaenge-472901.html>.

Unter <http://www.heise.de/security/dienste/Mails-mit-Anhaengen-777837.html> kann man sich auch Test-Mails mit harmlosen schädlichen Anhängen zusenden lassen. Dabei lässt sich gut ausprobieren, wie schädliche Anhänge funktionieren, ohne dass wirklich ein Schaden auf dem Rechner angerichtet wird.

#### 4.4 Gefahren beim Empfang von HTML-Mails abwenden

Text-Mails sind ungefährlich. Text-Mails stammen allerdings aus der „Steinzeit des Internet“. Sie unterstützen keine gestalterischen Elemente wie unterschiedliche Schriftarten, Kursivschrift oder eingebettete Bilder.

Mails sind heute meist kein reiner Text, sondern sie sind im HTML-Format verfasst. Gesundes **Misstrauen bei HTML-Mails** ist – wie bei Mails mit Anhängen – angebracht.

HTML-Mails sind multimedial. Wie Webseiten erlauben sie die Strukturierung und Formatierung des Textes (Überschriften, Kursiv- oder Fettdruck, Überschriften und Absätze, farbige Schriften oder Schrifthintergründe) aber auch eingebettete Links, Bilder, Sounds oder Videos und sogar kleine eingebettete Programme.

Doch genau diese Eigenschaften machen die HTML-Mails potentiell gefährlich.

- Die eingebetteten JavaScript-Programme können Unfug treiben.
- Ebenso können eingebettete Java-Applets Unsinn anstellen.
- Links können den Benutzer dazu verlocken, gefährliche Webseiten im Browser zu öffnen.
- Bilder (aber auch Audio- und Video-Dateien) können (durch sog. User-Tracking) den Benutzer ausspionieren.

Mehr Information zu den Gefahren von HTML-Mails findet man auf den Seiten des Heise-Verlags unter <http://www.heise.de/security/dienste/HTML-E-Mails-472898.html>.

Unter <http://www.heise.de/security/dienste/HTML-Mails-773971.html> kann man sich auch Test-Mails mit harmlosen HTML-Mails zusenden lassen, die jedoch das Gefahrenpotential, das diese Mails bergen, sehr gut veranschaulichen.

#### 4.4.1 Die Ausführung von JavaScript muss deaktiviert sein.

JavaScript ist eine Programmiersprache, die meist in erster Linie in Webbrowsern zum Einsatz kommt. JavaScript-Programme werden in HTML-Seiten eingebettet und sind dann für das dynamische Verhalten der Seite verantwortlich. Im Browser JavaScript abzuschalten, ist in der Regel nicht zweckmäßig, da man dann die meisten Webseiten nicht mehr sinnvoll nutzen kann.

JavaScript-Programme kann man auch in HTML-Mails einbetten. In diesem Fall ist das dynamische Verhalten des Mail-Inhalts allermeist unerwünscht. Es kann – wie schon gesagt – recht gefährlich sein, da man nicht kontrollieren kann, welche erwünschten oder unerwünschten Aktionen das JavaScript-Programm ausführt. Deshalb sollte man JavaScript im Email-Client abschalten (siehe auch Glossar, Kap 15).

Die Ausführung von **JavaScript** ist in *Thunderbird* per Voreinstellung deaktiviert.

Ob *Thunderbird* wirklich kein JavaScript ausführt, lässt sich leicht auf der folgenden Webseite von Heise prüfen. Auf <http://www.heise.de/security/dienste/emailcheck/html-mails/javascript/> gibt man die eigene Email-Adresse ein und fordert eine Test-Mail an. Kurz darauf erhält man eine automatisch erstellte Mail mit einem Anforderungslink. Erst beim Klick auf diesen Link erhält man die eigentliche Test-Mail, die harmlosen JavaScript-Code enthält, der nur ein Popup-Fenster öffnet. Erscheint beim Öffnen der zweiten Mail das Popup-Fenster, dann ist JavaScript in *Thunderbird* aktiviert und man muss es abschalten. Erscheint das Popup-Fenster nicht, dann führt *Thunderbird* den JavaScript-Code nicht aus und man darf beruhigt sein.

Sollte JavaScript wider Erwarten doch aktiviert sein, so muss man die recht versteckte erweiterte Konfiguration von *Thunderbird* öffnen und dort die Deaktivierung vornehmen:

- Erweiterte Konfiguration öffnen unter *Thunderbird* → *Einstellungen* → *Erweitert* → *Konfiguration bearbeiten...*
- Fenster mit der Warnung „*Ich werde vorsichtig sein, versprochen!*“ bestätigen. Jetzt geht's nämlich in die Eingeweide von *Thunderbird*. Eine gewaltige Liste von Einstellungen tut sich auf.
- Um die Liste auf die Einträge einzuschränken, die JavaScript betreffen, gibt man im Suchfenster „*javascript*“ ein.
- Die beiden folgenden Konfigurationsvariablen sollten auf „*false*“ eingestellt sein.
  - *javascript.enabled=false*

- javascript.allow.mailnews=false
- Fehlt einer der Werte, so ist dies unproblematisch, da die Voreinstellung „false“ auch dann gilt, wenn der betreffende Wert in der Liste nicht enthalten ist.

Einstellungsname	Status	Typ	Wert
javascript.allow.mailnews	vom Benutzer festgelegt	boolean	false
javascript.enabled	vom Benutzer festgelegt	boolean	false
javascript.options.asmjs	Standard	boolean	true
javascript.options.baselinejit	Standard	boolean	true
javascript.options.discardSystemSource	Standard	boolean	false
javascript.options.gc_on_memory_pressure	Standard	boolean	true
javascript.options.ion	Standard	boolean	true
javascript.options.ion.parallel_compilation	Standard	boolean	true
javascript.options.mem.gc_allocation_threshold_mb	Standard	integer	30

Abbildung 4: Erweiterte Einstellungen von Thunderbird: Konfigurationsvariablen für JavaScript

- Ist dies nicht der Fall (Variable ist auf „true“ eingestellt) kann der betreffende Wert durch einen Doppelklick auf die betreffende Zeile umgestellt werden.
- Das Fenster mit der erweiterten Konfiguration kann wieder geschlossen werden.

#### 4.4.2 Die Ausführung von Java-Applets muss deaktiviert sein.

Java-Applets sind in HTML-Seiten eingebettete Java-Programme. Sie können ebenfalls innerhalb einer HTML-Seite oder einer HTML-Mail aktiv werden und sind damit eine potentielle Gefahrenquelle. Java-Applets sind längst nicht so weit verbreitet wie JavaScript. Dennoch sollte auch ihre Ausführung im Mail-Client deaktiviert sein (siehe auch Glossar, Kap 15).

Ob *Thunderbird* tatsächlich Java-Applets ausführen würde, lässt sich analog zum JavaScript-Test im vorigen Kapitel auch auf der Webseite des Heise-Verlags prüfen unter:

<http://www.heise.de/security/dienste/emailcheck/html-mails/java/>

Ist Java auf dem PC installiert (dies ist auf den meisten PCs der Fall), dann findet sich in *Thunderbird* auch das Java-Plugin. Unter dem Menüpunkt *Extras → Add-ons → Plugins* werden alle in *Thunderbird* installierten Plugins aufgelistet. Hier lässt sich das Java-Plugin deaktivieren. Deaktivierte Plugins sind entsprechend gekennzeichnet und erscheinen grau unterlegt.



Abbildung 5: Thunderbird-Plugins: Das Java-Plugin (grau unterlegt) ist deaktiviert.

### 4.4.3 Vorsicht bei Links

HTML-Mails können ebenso wie HTML-Seiten Links enthalten. Beim Klick auf den Link öffnet der Browser die Webseite, auf die der Link verweist.

Unter dem Gesichtspunkt der Sicherheit, kann man jeden Link als Verführung des Benutzers betrachten, darauf zu klicken. Der Link selbst ist in der Regel ungefährlich. Er kann jedoch zu einer infizierten Webseite führen. Sobald sie geöffnet wird, wird das in der Webseite versteckte Schadprogramm heruntergeladen und möglicherweise auch gleich ausgeführt, falls der VirensScanner nicht vorher Alarm schlägt. Doch kann man sich auf den VirensScanner nicht immer zu 100 Prozent verlassen, da er die allerneuesten Viren möglicherweise nicht findet.

Ähnlich wie beim Öffnen von Mail-Anhängen muss der Nutzer beim Klicken auf Links in HTML-Mails besonders misstrauisch sein und prüfen, wohin der Link wirklich führt.

Die URL, die dem Linktext hinterlegt ist, wird sichtbar, wenn man mit der Maus auf den Linktext zeigt, ohne darauf zu klicken. Der komplette URL erscheint dann in der linken unteren Ecke des *Thunderbird*-Fensters.

Dort könnte dann <http://www.microsoft.com.boese-domain.net/dies/und/das/und/sonst/noch/was/> stehen. Sieht man nicht genau hin, so glaubt man, der Link würde auf eine Webseite von Microsoft führen. Tatsächlich führt der Link auf eine Seite von *boese-domain.net*.

Auch diese Falle lässt sich gefahrlos mit einer Test-Mail veranschaulichen, die man sich von Heise unter folgender URL anfordern kann:

[http://www.heise.de/security/dienste/emailcheck/html-mails/link\\_faelschung/](http://www.heise.de/security/dienste/emailcheck/html-mails/link_faelschung/). Man erhält dann eine Test-Mail, die mehrere Beispiele für solche irreführenden Hyperlinks enthält.

### 4.4.4 User-Tracking verhindern

HTML-Mails können externe eingebettete Objekte bzw. Inhalte enthalten, die erst bei der Anzeige der Mail aus dem Web nachgeladen und dann angezeigt oder ausgeführt werden. Solche externen Objekte sind häufig Bild-Dateien (im JPEG-, PNG- oder GIF-Format). Es können jedoch auch Audio- oder Video-Dateien sein oder die in den vorigen Kapiteln bereits erwähnten JavaScript-Programme. Audio- und Video-Dateien werden allerdings in HTML-Mails wesentlich seltener eingesetzt als Bilder. Bilder in HTML-Mails zu verschicken, ist gang und gäbe. Ich beschränke mich hier auf die Bilder. Manchmal sind sie nur ein Pixel klein und damit ganz unauffällig.

*Extern* nennt man diese Objekte, weil sie nicht Teil der HTML-Mail sind. In der Mail befinden sich nur die URLs dieser Objekte. Beim Empfang einer HTML-Mail befinden sich diese Objekte

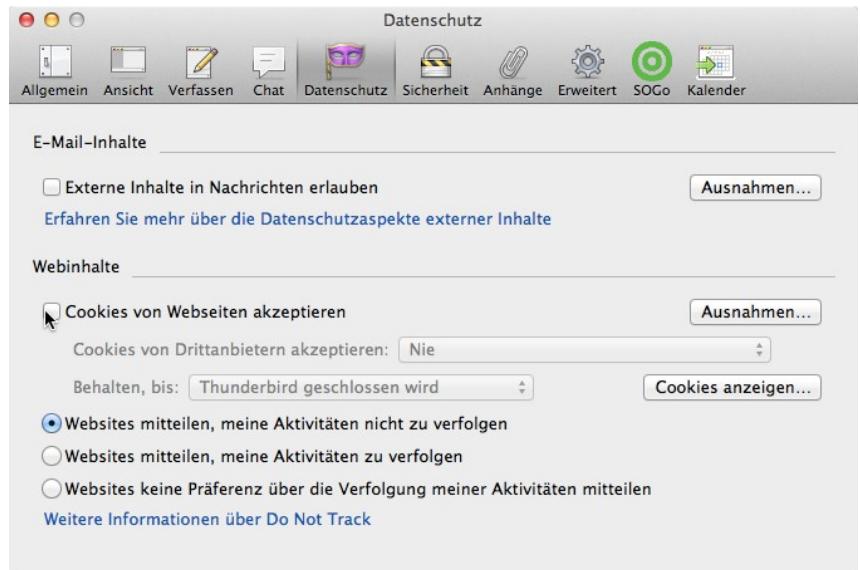


Abbildung 6: Thunderbird-Einstellungen: Nachladen externer Inhalte deaktivieren

noch nicht auf dem Rechner. Sie werden erst aus dem Web nachgeladen, wenn die HTML-Mail angezeigt werden soll.

In *Thunderbird* lässt sich das **Nachladen externer Objekte bzw. Inhalte deaktivieren** oder aktivieren.

Die Deaktivierung ist die Voreinstellung.

Abbildung 7 und 8 zeigen eine Werbemail von *WEB.DE* vor und nach dem Laden der externen Objekte. Dies veranschaulicht, wie sehr manche HTML-Mails auf ihre externen Inhalte angewiesen sind.

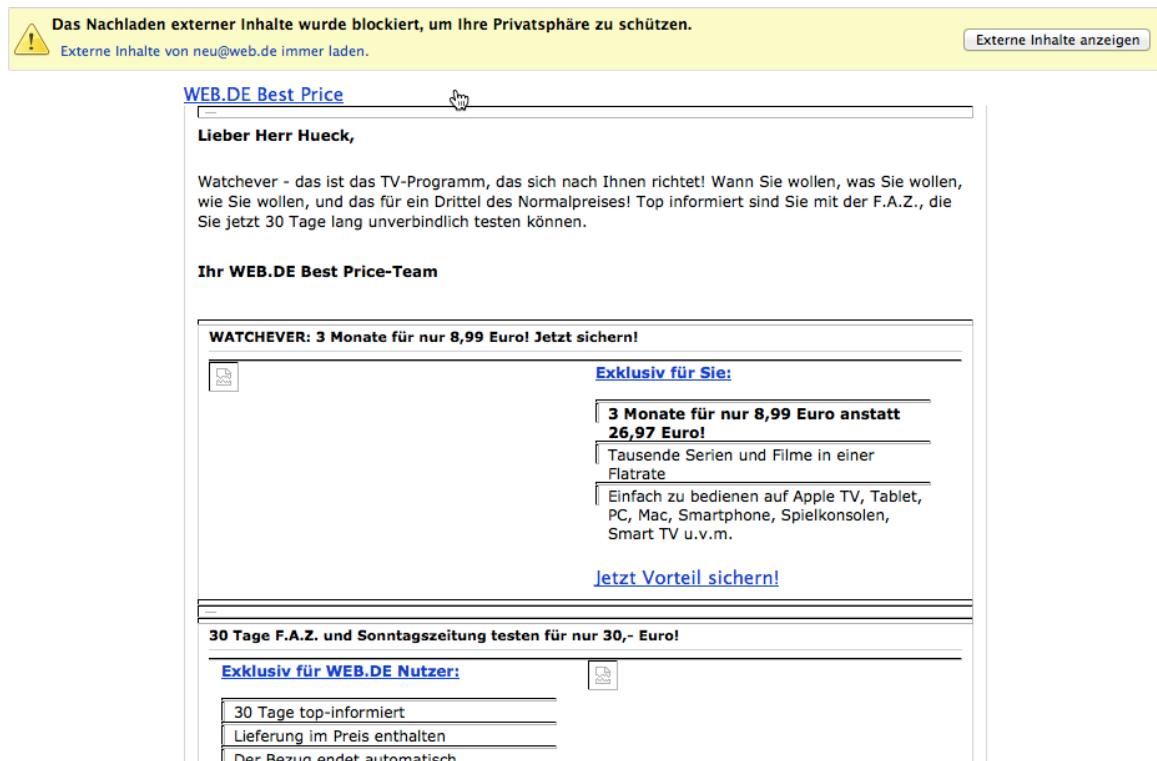


Abbildung 7: *Thunderbird*: Werbemail von *WEB.DE* vor dem Laden der externen Inhalte

In den Bildern lauert in der Regel keine Gefahr in Gestalt von Schadprogrammen, die auf dem Rechner ihr Unwesen treiben.

Die Gefahr ist eine andere. Die Bilder (und die anderen externen Objekte) können beim Nachladen Informationen über den Mail-Empfänger an den Versender liefern. Sie können den Mail-Empfänger quasi ausspionieren oder verfolgen (User-Tracking). Dies wird nicht nur von den vielen Versendern von Newslettern praktiziert, auch Spam-Versender sind sehr daran interessiert, etwas über die Empfänger der Spam-Mails zu erfahren.

Welche Informationen lassen sich durch User-Tracking über den Mail-Empfänger herausfinden? Der Mail-Versender kann auf diesem Wege erfahren,

- wann die Mail geöffnet wurde (Uhrzeit),
- wo die Mail gelesen wurde (die IP-Adresse des Empfängers lässt sich ermitteln und diese lässt häufig den Rückschluss auf seinen Aufenthaltsort zu),
- auf welchem Betriebssystem (Windows, OS X, iOS oder Android) die Mail gelesen wurde,

- mit welchem Mail-Client die Mail gelesen wurde (Thunderbird, Outlook oder ein anderes Mail-Programm oder bei Nutzung von Webmail mit welchem Browser)
- und bei welchem Internet-Provider der Mail-Empfänger aktuell seinen Internetzugang hat.
- Und schließlich wurde die Existenz der Email-Adresse verifiziert. Die Tatsache, dass eine Mail geöffnet und die externen Inhalte nachgeladen wurden, ist ein Nachweis dafür, dass die betreffende Mail-Adresse gültig ist und von jemandem benutzt wird. Dies kann für Spam-Versender interessant sein, die Millionen Emails an Adressen versenden, von denen sie gar nicht wissen, ob sie existieren. Durch das Laden der externen Inhalte ist die Mail-Adresse bestätigt.

Alles in Allem sind dies durchaus sensible Informationen. Durch das Deaktivieren des Ladens der externen Inhalte schützen wir unsere Privatsphäre. Möchte man eine bestimmte Mail (von einem Absender, dem man vertraut) „in ihrer ganzen Schönheit“ sehen, dann kann man die externen Inhalte in *Thunderbird* jederzeit mit einem einzigen Mausklick anfordern und laden (siehe Abb. 7).

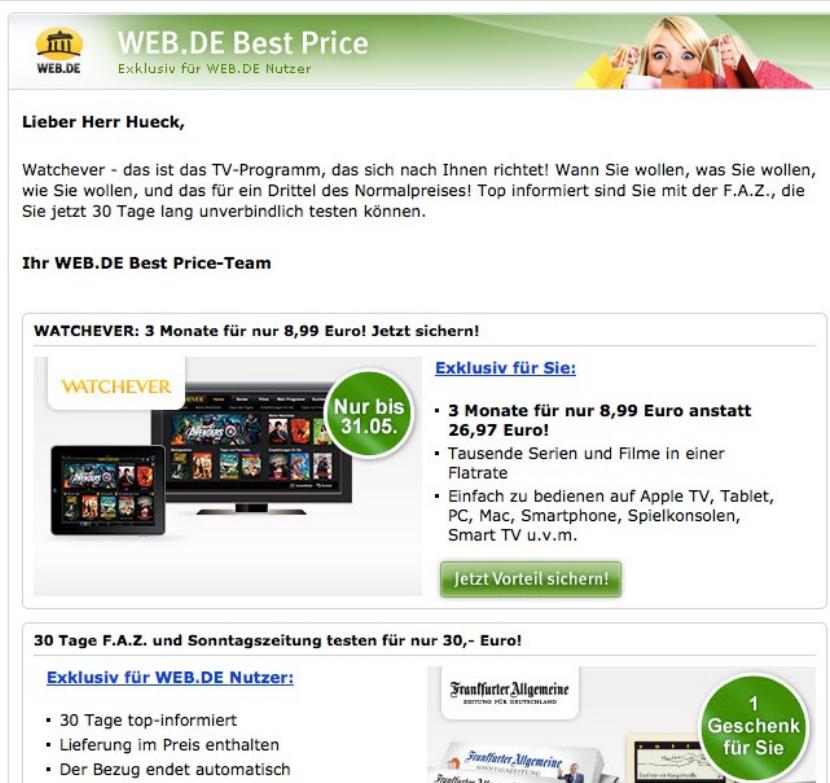


Abbildung 8: Thunderbird: Werbemail von WEB.DE nach dem Laden der externen Inhalte

Sog. **Tracking-Images** sind kleine, unsichtbare, in eine HTML-Mail eingebettete Bilder. Sie sind in der Regel weiß wie die Hintergrundfarbe und nur 1x1 Pixel groß und damit für den Mail-Empfänger quasi unsichtbar. Diese unsichtbaren Bilder haben den einzigen Zweck, den „User zu tracken“ und dabei in der Mail nicht aufzufallen.

Auch zu diesem Problemfeld kann man bei Heise unter der URL <http://www.heise.de/security/dienste/emailcheck/html-mails/webbug/> eine Test-Mail anfordern, die das Problem anschaulich demonstriert, ohne tatsächlich Schaden anzurichten.

Mehr Infos zu diesem Thema sind wieder im c't-Sonderheft „Sichere E-Mail“ im Artikel „Tracking aufspüren und abstellen“ zu finden (siehe Kap. 2.8).

## 4.5 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 Website 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.
- Das Laden der externen Inhalte und die Ausführung von JavaScript lässt sich im Webbrowser allerdings längst nicht so einfach verhindern wie in *Thunderbird*. Denn im Webbrowser will man das Nachladen der Bilder und die Ausführung von JavaScript-Programmen in aller Regel aktiviert lassen. Sonst könnte man nicht mehr komfortabel im Web surfen.

Eine gute **Grundregel: Webmail so wenig wie möglich nutzen**. Für Smartphone-Besitzer ist das auch kaum mehr erforderlich, da sie mit dem Smartphone fast immer online sind und so nahezu an jedem Ort auf ihre Mails zugreifen können, ohne den Browser zu benutzen.

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 die verschlüsselten Mails mit Webmail weder lesen noch versenden (siehe Kap. 8.2).

## 4.6 Zusammenfassung

Dieses Kapitel befasst sich mit dem richtigen Umgang mit den Mails. Es gibt Empfehlungen für die möglichst sichere Mail-Konfiguration und -Nutzung. Dazu muss man

- die Rechnersicherheit beachten (aktuelles Betriebssystem, aktuelle Programme, aktuellen Virenscanner),
- dafür sorgen, dass der Mail-Client *Thunderbird* immer aktuell ist,
- dass man Mail-Anhänge mit Bedacht und möglichst nur dann öffnet, wenn man den Absender kennt,
- und bei HTML-Mails die entsprechende Vorsicht walten lässt.

Während Text-Mails ungefährlich sind, lauern bei HTML-Mails einige Gefahren, die sich einerseits durch entsprechende *Thunderbird*-Einstellungen, andererseits durch vorsichtiges Verhalten minimieren lassen:

- Im Mail-Client muss JavaScript deaktiviert sein.
- Auch Java-Applets müssen deaktiviert sein.
- Beim Klicken auf in Mails enthaltene Links ist Vorsicht geboten. Zuerst genau prüfen, wohin der Link führt, bevor man darauf klickt!
- Indem man das automatische Laden externer Inhalte abstellt, verringert man die Gefahr, ausspioniert zu werden (User-Tracking) erheblich. Bei Bedarf kann man die externen Inhalte im Einzelfall dennoch nachladen.

Schließlich werden noch die Risiken von Webmail (Mailzugriff mit dem Webbroweser) erläutert.

Das vorige Kapitel 3 und dieses Kapitel 4 hatten die Sicherheitsmaßnahmen zum Thema, die sowohl für unverschlüsselte als auch für verschlüsselte Mails wichtig sind. Erst in den folgenden Kapiteln geht es um die Verschlüsselung der Mail.

Die Sicherheitsmaßnahmen, die in diesen Kapiteln beschrieben wurden, und das Sicherheitsbewusstsein, das für die richtige Nutzung der Email erforderlich ist, halte ich für wichtiger als die Verschlüsselung der Mail. Derjenige, für den Email-Sicherheit ein ernstes Anliegen ist, sollte sich zunächst um die in diesen beiden Kapiteln beschriebenen Maßnahmen kümmern. Und wer es dann ganz sicher haben will und auch den Einstieg in das komplexe Thema nicht scheut, dem empfehle ich, die Emails zu verschlüsseln und sich auch mit den nachfolgenden Kapiteln zu befassen.

## 4.7 Links zu diesem Kapitel

- Thunderbird-Updates:  
<https://support.mozilla.org/de/kb/thunderbird-update-einstellungen>
- Informationen zu schädlichen Email-Anhängen bei Heise:  
<http://www.heise.de/security/dienste/Dateianhaenge-472901.html>
- Test-Mails mit harmlosen, schädlichen Email-Anhängen bei Heise:  
<http://www.heise.de/security/dienste/Mails-mit-Anhaengen-777837.html>
- Informationen zu gefährlichen HTML-Mails bei Heise:  
<http://www.heise.de/security/dienste/HTML-E-Mails-472898.html>
- Test-Mails mit harmlosen, gefährlichen HTML-Mails bei Heise:  
<http://www.heise.de/security/dienste/HTML-Mails-773971.html>
- Test-Mail-Anforderung für den JavaScript-Test bei Heise:  
<http://www.heise.de/security/dienste/emailcheck/html-mails/javascript/>
- Test-Mail-Anforderung für den Java-Applet-Test bei Heise:  
<http://www.heise.de/security/dienste/emailcheck/html-mails/java/>
- Test-Mail-Anforderung für den Test falscher Links bei Heise:  
[http://www.heise.de/security/dienste/emailcheck/html-mails/link\\_faelschung/](http://www.heise.de/security/dienste/emailcheck/html-mails/link_faelschung/)
- Test-Mail-Anforderung für den Test eingebetteter Bilder bei Heise:  
<http://www.heise.de/security/dienste/emailcheck/html-mails/webbug/>

## 5 Verschlüsselte und signierte Mails mit PGP – Die „graue“ Theorie

Wenn die Mail nicht verschlüsselt ist, 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 die Mails nicht mehr lesen. Nicht nur die Provider – auch andere haben keinen Zugriff mehr auf die Mails (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, CC, Betreff, 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 ausschließlich in unserer 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 können nur Absender und 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.)

Dieses und die folgenden Kapitel versuchen, das schwierige Thema Email-Verschlüsselung mit **PGP (Pretty Good Privacy)** so einfach wie möglich zu erläutern, sodass auch der IT-Laie sie verstehen und umsetzen kann. Schließlich möchte ich dich als Mail-Partner gewinnen, mit dem ich verschlüsselte Mails austauschen kann.

Ich erläutere in diesem Kapitel die wichtigsten Konzepte der Verschlüsselung mit PGP und gebe in den darauf folgenden Kapiteln die praktischen Anleitungen dazu. Ich liefere auch viele Web-Links mit, die es den Lesern ermöglichen, weitergehende Informationen an den betreffenden Web-Adressen abzurufen. Technische Details beschreibe ich nur in dem Umfang, wie sie zum Verständnis unbedingt erforderlich sind.

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 und besonders die Schlüssel-Verwaltung zu verstehen und einzurichten, ist eine harte Nuss. Doch es lohnt sich, sie zu knacken.** Ich liefere euch hiermit den Nussknacker dazu.

### 5.1 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 kommt 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.

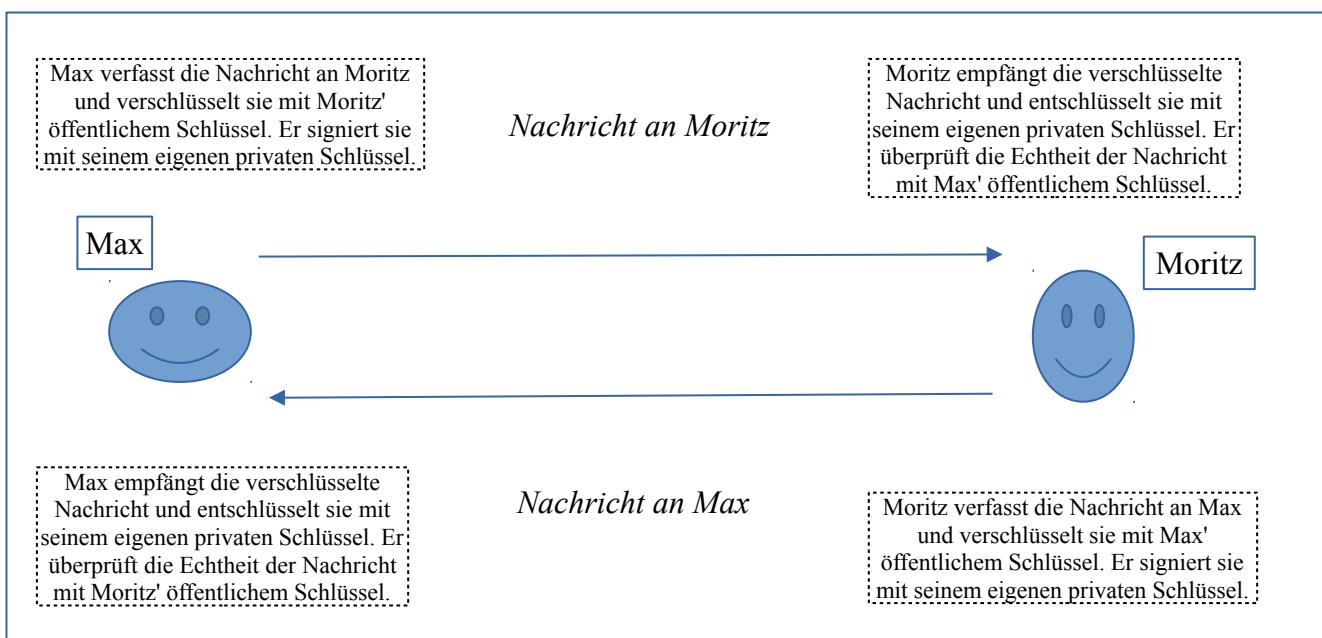


Abbildung 9: Asymmetrisch verschlüsselte Kommunikation

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. 7.1.9).

## 5.2 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](http://de.wikipedia.org/wiki/Pretty_Good_Privacy)

Beide Verfahren sind asymmetrische Verschlüsselungsverfahren (siehe Kap. 5.3). 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; danach gilt er 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 installierten 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.

Tatsächlich kommuniziere ich dann (ohne dass der Browser es bemerkt) nicht mit dem echten Server von Google, sondern mit dem falschen Server der NSA, der sich als Google-Server ausgibt.

Zweites Szenario: Hacker brechen bei einer CA ein und können ihre eigenen Zertifikate im Namen der gehackten CA ausstellen. Sie können damit ebenfalls einen falschen Google-Server mit vermeintlich echten Zertifikaten betreiben. Dies ist im Jahr 2012 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 Website mit einem von Diginotar beglaubigten Zertifikat per HTTPS besuche. Ein Warnhinweis im Browser teilt mir mit, dass das Zertifikat der Website, die ich besuchen will, nicht überprüft werden kann. Nun kann ich die Warnung beachten und auf den Besuch der fraglichen Website verzichten oder die Warnung ignorieren und die Site auf eigenes Risiko 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 korrumptierbar.

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

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

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. 7.1.9).

### 5.3 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!)

Wie oben beschrieben (siehe Kap. 5.1) basiert asymmetrische Verschlüsselung 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 veröffentlicht seinen öffentlichen Schlüssel, d.h. er gibt ihn allen anderen Benutzern, mit denen er kommuniziert, oder er macht ihn sehr leicht zugänglich. Seinen privaten Schlüssel sichert er so gut es geht, am besten passwortgeschützt 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 entschlüsseln und 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, verschlüssle 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. (Dies 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. (Dies bedeutet Signaturprüfung.)

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. 7.1.9.

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 beglaubigtem 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.

Nehmen wir an, 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 Moritz Witwe Bolte persönlich, weil er sie kennt, oder er hat sich ihren Ausweis zeigen lassen.) Also kann er jetzt Witwe Bolte vertrauen, da er Moritz (Witwe Boltes Beglaubiger) 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 (PC oder mobiles Gerät) 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 noch detaillierter ist dies alles beschrieben im c't-Sonderheft „Sichere Email“ (siehe Kap. 2.8) auf Seite 82 im Artikel „Verschlüsseln und Signieren mit PGP“. Meine Ausführungen reduziere ich auf das, was man braucht, um möglichst schnell zum Ziel zu kommen und verschlüsselte und signierte Mails versenden und empfangen zu können. Bei der Fokussierung auf die Praxis der Verschlüsselung kann die technische Detailtiefe auch zuweilen auf der Strecke bleiben.

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

## 5.4 Einschränkungen bei der Nutzung verschlüsselter Mails

Bevor wir mit der Verschlüsselung loslegen, möchte ich nochmals auf die (bereits in Kap. 1.5 aufgeführten) Nachteile hinweisen, die die Mail-Verschlüsselung mit sich bringt.

- Der Zugriff auf die Mails mit dem Browser (Webmail) ist nicht mehr möglich.
- Die Volltextsuche durch die Mail-Inhalte ist nicht möglich.
- Der Provider kann für verschlüsselte Mails keinen Spam- und Virenschutz bereitstellen.

Diesen Preis bezahlt man für die durch die Verschlüsselung gewonnene Privatsphäre.

## 5.5 Zusammenfassung

In diesem Kapitel habe ich die grundlegenden Konzepte der Email-Verschlüsselung erläutert.

Email-Verschlüsselung beruht immer auf einem **asymmetrischen Verschlüsselungsverfahren**. Zwei solcher Verfahren gibt es, **S/MIME** und **PGP**. Diese unterscheiden sich im Kern durch die Art der Schlüsselbeglaubigung. Bei S/MIME werden die Schlüssel durch sog. CAs (Certificate Authorities) beglaubigt. CAs können allerdings kompromittiert werden und damit werden auch die von einer kompromittierten CA beglaubigten Schlüssel unglaublich. Bei PGP beglaubigen die kommunizierenden Benutzer ihre Schlüssel gegenseitig. Dieses Dokument beschreibt Email-Verschlüsselung auf Basis von PGP.

Bei PGP erstellt jeder Benutzer ein Schlüsselpaar aus **Private Key** und **Public Key**. Der private Schlüssel muss gut geschützt werden und darf niemals in andere Hände geraten. Der öffentliche Schlüssel wird an andere Benutzer verteilt oder leicht zugänglich bereit gestellt.

Mit dem öffentlichen Schlüssel eines anderen Benutzers kann man ...

- eine Mail an diesen Benutzer verschlüsseln, sodass nur dieser sie entschlüsseln kann,

- eine Mail von diesem Benutzer verifizieren, d.h. nachweisen, dass sie genau von diesem Benutzer stammt (Authentizität der Nachricht), und dass kein Jota der Nachricht verändert wurde (Unverfälschtheit der Nachricht).

Mit dem eigenen privaten Schlüssel kann man ...

- eine empfangene, verschlüsselte Mail von einem anderen Benutzer, die dieser mit dem eigenen öffentlichen Schlüssel verschlüsselt hat, entschlüsseln,
- die Mail an einen anderen Benutzer signieren
- und den öffentlichen Schlüssel eines anderen Benutzers signieren/beglaubigen, nachdem man geprüft hat, dass der Schlüssel und die Email-Adresse(n) wirklich diesem Benutzer gehören.

Durch wechselseitiges Signieren/Beglaubigen von öffentlichen Schlüsseln entsteht zwischen zwei Benutzern ein **direktes Vertrauensverhältnis**. Vertraut ein Benutzer einem anderen Benutzer, dem sein direkter Vertrauter vertraut, so handelt es sich um ein **transitives Vertrauensverhältnis**. Durch diese direkten und transitiven Vertrauensverhältnisse entsteht ein Vertrauensnetz, ein sog. **Web of Trust (WoT)**.

## 5.6 Endlich loslegen

Im übernächsten Kapitel 7 beschreibe ich ausführlich die Einrichtung und Benutzung von PGP auf dem PC. Ich beschreibe nicht nur, was zu tun ist, sondern auch warum das so zu machen ist. Der Benutzer soll wissen, was er tut. Manchen mag das zu ausführlich sein.

Kapitel 6 enthält deshalb eine PGP-Schnellanleitung, die eher wie ein Kochrezept aufgebaut ist. Die Ungeduldigen können mit Kapitel 6 weiterlesen. Diejenigen, die sich der Materie über die ausführlichen Darstellungen nähern möchten, können Kapitel 6 überspringen und die Lektüre mit Kapitel 7 fortsetzen. Ich persönlich halte die in Kapitel 7 beschriebenen, ausführlichen Weg für den besseren. Gerade für den Einsteiger, der mit PGP noch nicht vertraut ist, sind ausführliche Erläuterungen und Screenshots eher hilfreich und unterstützen das Verständnis. Doch jeder soll selbst entscheiden, auf welchem Weg er schneller zum Ziel kommt.

Ebenso wie Kapitel 6 und 7 bieten auch die Kapitel 9 und 10 alternative Möglichkeiten des Einstiegs in Mail-Verschlüsselung mit PGP auf dem Android-Gerät.

Hier noch einmal der Überblick über die folgenden Kapitel für die praktische Nutzung von PGP:

- Kapitel 6: PGP auf dem PC – Schnelleinstieg für Ungeduldige
- Kapitel 7: PGP auf dem PC – Ausführlicher Einstieg für Wissbegierige
- Kapitel 8: PGP – Was noch wichtig oder interessant ist
- Kapitel 9: PGP auf dem Android-Gerät – Schnelleinstieg für Ungeduldige
- Kapitel 10: PGP auf dem Android-Gerät – Ausführlicher Einstieg für Wissbegierige
- Kapitel 11: PGP/MIME – Und es funktioniert doch.

## 5.7 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](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> .

## 6 PGP auf dem PC – Schnelleinstieg für Ungeduldige

Dieses Kapitel liefert die Schritt-für-Schritt-Anleitungen für PGP auf dem PC (mit Windows, Mac OS X oder Linux). Es verzichtet auf Screenshots, auf ausführliche Erläuterungen und auf die Darstellung verschiedener Optionen, sondern zeigt nur den von mir favorisierten Weg. Hier werden die Schritte aufgeführt, die durchzuführen sind, um PGP in *Thunderbird* auf dem PC einzurichten und zu nutzen. Dabei wird vorausgesetzt, dass *Thunderbird* auf dem PC zum Mailversand und -empfang verwendet wird. Die einzelnen Schritte haben Verweise ins Kapitel 7, sodass die ausführlichen Erläuterungen bei Bedarf schnell auffindbar sind.

### 6.1 PGP-Schlüsselverwaltung mit *Thunderbird/Enigmail* auf dem PC

Als erstes benötigen wir einen Schlüsselbund. Darin muss sich mindestens ein eigenes Schlüsselpaar bestehend aus Private Key und Public Key befinden. Außerdem kann der Schlüsselbund beliebig viele öffentliche Schlüssel von Kommunikationspartnern enthalten. Das eigene Schlüsselpaar muss erzeugt werden. Die öffentlichen Schlüssel der Kommunikationspartner müssen vom Schlüssel-Server importiert, dann beglaubigt und schließlich wieder auf den Schlüssel-Server exportiert werden.

- **Native Schlüsselbund-Verwaltung installieren** (siehe Kap. 7.1.1)
  - Auf dem Windows-PC: **Gpg4win** installieren
  - Auf dem Mac: **GPG Suite** installieren
  - Auf dem Linux-PC muss keine Zusatzsoftware installiert werden. **GnuPG** ist in der Regel auf jedem Linux-System vorinstalliert und kann direkt von der Kommandozeile aus genutzt werden.
- Mit der Schlüsselbund-Verwaltung (oder mit dem Kommando `gpg --gen-key`) ein **neues Schlüsselpaar erzeugen** (siehe Kap. 7.1.2). Dabei sind verschiedene Angaben zu machen:
  - Primäre Benutzer-ID: Vorname und Nachname und die primäre Email-Adresse (diejenige, die man in erster Linie zur Kommunikation verwendet). Optional kann auch ein Kommentar angegeben werden.
  - Verschlüsselungs-Algorithmus: RSA
  - Schlüssellänge: 4096 Bit. Ein langer Schlüssel mit 4096 Bit ist sicherer als ein kurzer mit 2048 Bit. Ein Schlüssel mit einer Länge von 1028 Bit ist als unsicher zu betrachten und sollte deshalb nicht erzeugt werden.
  - Gültigkeitsdauer des Schlüssels: unbegrenzt (Ablaufdatum leer lassen)
  - Das Widerrufszertifikat noch nicht erzeugen (wird später erzeugt)
  - Die Passphrase sollte mindestens 12 Zeichen lang sein und aus Großbuchstaben, Kleinbuchstaben, Ziffern und Sonderzeichen bestehen.
- **Weitere Benutzer-IDs zum erzeugten Schlüsselpaar hinzufügen** (siehe Kap. 7.1.4): Will man den generierten Schlüssel mit weiteren Email-Adressen nutzen, dann kann für jede Email-Adresse eine weitere Benutzer-ID hinzugefügt werden. Vorname und Nachname dürfen dieselben sein wie in der primären Benutzer-ID, die Email-Adresse muss sich unterscheiden.

- **Schlüssel-Server** in der nativen Schlüsselbund-Verwaltung **konfigurieren**: Folgende Schlüssel-Server sind in meiner Konfiguration in der angegebenen Reihenfolge eingetragen:
  - a.keyserver.pki.scientia.net
  - p80.pool.sks-keyservers.net
  - pgp.mit.edu
  - subkeys.pgp.net
- Die native Schlüsselbund-Verwaltung kann jetzt geschlossen werden, da die weiteren Schritte mit der *Enigmail*-Schlüsselbund-Verwaltung innerhalb von *Thunderbird* durchgeführt werden können.
- **Enigmail installieren** (siehe Kap. 7.1.1): In *Thunderbird* unter Menüpunkt *Extras* → *Add-ons* den Add-ons-Manager starten und dort den Suchbegriff *Enigmail* eingeben und danach suchen. Bei *Enigmail* auf *Installieren* klicken und *Thunderbird* neu starten. Den Add-ons-Manager schließen.
- Nach der Installation von *Enigmail* steht in *Thunderbird* der neue Menüpunkt *Enigmail* zur Verfügung. Unter *Enigmail* → *Schlüssel verwalten...* ist unter Anderem auch die *Enigmail*-Schlüsselbund-Verwaltung erreichbar.
- **Schlüssel-Server in *Enigmail* konfigurieren** (siehe Kap. 7.1.7): In *Thunderbird* unter *Enigmail* → *Einstellungen...* → *Schlüssel-Server* die Schlüssel-Server eintragen, die man verwenden will. Es sind dieselben Schlüssel-Server, die man zuvor in der nativen Schlüsselbund-Verwaltung konfiguriert hat (s.o.). Bei mir sind dies folgende:
  - a.keyserver.pki.scientia.net
  - p80.pool.sks-keyservers.net
  - pgp.mit.edu
  - subkeys.pgp.net
- **Eigenen öffentlichen Schlüssel auf Schlüssel-Server hochladen** (siehe Kap. 7.1.8): In der *Enigmail*-Schlüsselbund-Verwaltung den eigenen Schlüssel markieren und auf den Schlüssel-Server hochladen. Es genügt das Hochladen auf einen einzigen Schlüssel-Server, da die Schlüssel-Server ihre Daten wechselseitig abgleichen (SKS = Synchronizing Key Server). Nach ca. 24 Stunden befindet sich der eigene hochgeladene Schlüssel auf allen Schlüssel-Servern des Internets. (Vorsicht: Keine Testschlüssel hochladen! Hochgeladene Schlüssel können nie wieder vom Schlüssel-Server gelöscht werden.)
- **Den öffentlichen Schlüssel eines Kommunikationspartners vom Schlüssel-Server importieren** (siehe Kap. 7.1.8): In der *Enigmail*-Schlüsselbund-Verwaltung den Menüpunkt *Schlüssel-Server* → *Schlüssel suchen...* aufrufen. In das Suchfeld die Email-Adresse eines Kommunikationspartners oder einen Teil davon eingeben. Die Treffer der Suche werden aufgelistet. Aus der Trefferliste können einzelne Schlüssel ausgewählt und in die Schlüsselbund-Verwaltung importiert werden.
- **Den öffentlichen Schlüssel des Kommunikationspartners prüfen**, bevor man ihn beglaubigt: Einen Schlüssel mit einer bestimmten Email-Adresse kann jeder erstellen. Bevor man den importierten Schlüssel des Kommunikationspartners benutzt, muss man sicher sein, dass es sich tatsächlich um den Schlüssel des vermuteten Kommunikationspartners ist. Dies

muss zunächst geprüft werden (genaue Beschreibung in Kapitel 7.1.9.1).

- **Den öffentlichen Schlüssel des Kommunikationspartners** (nach erfolgreicher Prüfung) **beglaubigen** (siehe Kap. 7.1.9). Technisch gesehen heißt „beglaubigen“, den fremden Schlüssel mit dem eigenen Schlüssel unterschreiben/signieren. Man markiert den zu beglaubigenden Schlüssel und wählt im Kontextmenü den Punkt „*Unterschreiben...*“. Dies genügt noch nicht. Man muss auch noch das Vertrauensverhältnis zum Schlüssel definieren. Im Kontextmenü unter „*Besitzervertrauen festlegen...*“ kann man zwischen fünf Vertrauensstufen auswählen (siehe Kap. 7.1.5 und 7.1.9). Meist ist „Volles Vertrauen“ für den zuvor signierten Schlüssel festzulegen. (Die Vertrauensstufe „*Absolutes Vertrauen*“ sollte nur für eigene Schlüssel verwendet werden.)
- **Den signierten, öffentlichen Schlüssel des Kommunikationspartners wieder auf den Schlüssel-Server hochladen** (siehe Kap. 7.1.9): Im Kontextmenü des betreffenden Schlüssels wählt man den Punkt „*Auf Schlüssel-Server hochladen...*“. Der Schlüssel des Kommunikationspartners wird dadurch auf den Key-Servern aktualisiert und enthält auch meine Beglaubigung. Damit gibt man allen anderen PGP-Benutzern bekannt, dass man diesem Schlüssel vertraut. Diejenigen, die mir trauen, können dann evtl. auch meinem Kommunikationspartner trauen, da sie annehmen, dass ich die Signierung seines Schlüssel erst nach gewissenhafter Prüfung vorgenommen habe.

#### Links:

- Gpg4win, Download unter <http://www.gpg4win.de/> oder unter <http://www.heise.de/download/gpg4win-e2d914fd06f82399ae36052e8362b95c-1398246471-2619466.html>
- GPG Suite 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>
- Detaillierte Beschreibung der Verschlüsselung und Schlüsselverwaltung mit *Thunderbird/Enigmail*: [http://www.thunderbird-mail.de/wiki/Enigmail\\_OpenPGP](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 Suite*: <http://support.gpgtools.org/kb>

## 6.2 Schlüssel und Widerrufszertifikat sichern

1. Das eigene Schlüsselpaar (privater und öffentlicher Schlüssel) muss gesichert werden, denn bei

einem Festplattencrash muss das Schlüsselpaar wieder hergestellt werden können.

2. Das Widerrufszertifikat muss gesichert werden. Mit diesem kann man den eigenen Schlüssel auf den Schlüssel-Servern widerrufen, d.h. für ungültig erklären. Das geht auch dann, wenn der private Schlüssel abhanden gekommen sein sollte.

3. Die gesammelten öffentlichen Schlüssel können gesichert werden. Dies ist nicht notwendig, jedoch zweckmäßig. Würden diese verloren gehen, so könnte man sie sich jeder Zeit wieder von einem Schlüssel-Server holen. Die Sicherung ist dennoch sinnvoll, z.B. um sie auf einem anderen Gerät in einem Rutsch importieren zu können (siehe Kap. 6.5).

Das zweckmäßigste Sicherungsmedium ist ein USB-Stick (oder eine Speicherkarte), auf dem sich am besten keine anderen Daten befinden. Der Stick wird also ausschließlich für die Sicherung der Schlüsseldateien verwendet.

Dazu verfährt man am besten so:

- Leeren USB-Stick an den Rechner anschließen
- In der *Enigmail*-Schlüsselbund-Verwaltung den zuvor generierten Schlüssel markieren und dann „*In Datei exportieren...*“. Dabei ist darauf zu achten, dass *der private und der öffentliche Schlüssel* exportiert wird. Die exportierte Datei direkt auf dem angeschlossenen USB-Stick speichern. (siehe Kap. 7.1.6)
- In der *Enigmail*-Schlüsselbund-Verwaltung den zuvor generierten Schlüssel markieren und dann „*Widerrufszertifikat erzeugen & speichern...*“. Die erzeugte Datei mit dem Widerrufszertifikat auf dem angeschlossenen USB-Stick speichern (siehe Kap. 7.1.3).
- In der *Enigmail*-Schlüsselbund-Verwaltung alle Schlüssel markieren und dann „*In Datei exportieren...*“. Dabei ist darauf zu achten, dass *nur die öffentlichen Schlüssel* exportiert wird. Die exportierte Datei direkt auf dem angeschlossenen USB-Stick speichern (siehe Kap. 7.1.6).
- Nach dem erfolgreichen Export der drei Dateien den USB-Stick abmelden und vom Rechner abziehen.
- Den USB-Stick gut aufbewahren. Bei einem Festplattencrash des Rechners kann der private Schlüssel nur vom USB-Stick wieder hergestellt werden (siehe Kap. 7.1.6.1).
- Um die Sicherheit weiter zu erhöhen, kann man die auf dem USB-Stick gespeicherten Dateien zusätzlich auf eine CD brennen.
- Es sollten keine exportierten Schlüsseldateien auf der Festplatte des Rechners bleiben. Ein Schadprogramm, das den Rechner befällt, könnte die Dateien stehlen. Exportierte Schlüsseldateien auf der Festplatte des Rechners sind zu löschen.

### 6.3 Thunderbird für PGP-Nutzung konfigurieren

Nun habe ich einen Schlüsselbund, der mindestens ein eigenes Schlüsselpaar (Private Key und Public Key) enthält. Außerdem kann der Schlüsselbund beliebig viele weitere öffentliche Schlüssel enthalten, einen von jedem Kommunikationspartner, der seinen öffentlichen Schlüssel auf einen Key-Server exportiert hat und den ich importiert und beglaubigt habe.

In den folgenden Schritten wird die *Thunderbird/Enigmail*-Konfiguration beschrieben, die zur Nutzung dieser Schlüssel beim Versand und beim Empfang signierter und verschlüsselter Mails erforderlich ist. Fast alle Einstellungen sind konten-spezifisch, d.h. sie sind für jedes *Thunderbird*-

Mailkonto vorzunehmen, bei dem PGP genutzt werden soll.

- **PGP-Unterstützung für das Konto aktivieren** (siehe Kap. 7.2.1)
  - Konto-Einstellungen des Mail-Kontos öffnen, das mit PGP genutzt werden soll: Im Kontextmenü des betreffenden Kontos den Punkt „*Einstellungen*“ auswählen.
  - Den Menüpunkt *OpenPGP-Sicherheit* des betreffenden Kontos wählen
  - Die Option „*OpenPGP-Unterstützung für diese Identität aktivieren*“ auswählen
- Den **für das Konto zu verwendenden eigenen Schlüssel angeben** (siehe Kap. 7.2.1). Dieser kann entweder über die im Schlüssel eingetragene Email-Adresse oder über die Schlüssel-ID angegeben werden.
- Die folgenden Optionen legen nur **Standard-Vorgaben für den Mail-Versand** fest. Beim Versand jeder Mail können davon abweichend andere Festlegungen getroffen werden (siehe Kap. 7.2.1). Außerdem kann man mit sog. Empfängerregeln (s.u.) für einzelne Empfänger abweichende Einstellungen für die folgenden Standard-Vorgaben treffen.
  - Die Option „*Nachrichten standardmäßig verschlüsseln*“ nicht auswählen. Diese Option zu wählen ist in der Regel nicht sinnvoll, da man normalerweise nur mit sehr wenigen Kommunikationspartnern verschlüsselt kommunizieren kann.
  - Die Option „*Nachrichten standardmäßig unterschreiben*“ auswählen
  - Die Option „*PGP/MIME standardmäßig verwenden*“ nicht auswählen. PGP/MIME ist zwar das modernere Format für verschlüsselte Mails. Es wird aber von einigen Mail-Clients noch nicht unterstützt. Deshalb rate ich zurzeit von seiner Verwendung ab.
  - Die Option „*Unverschlüsselte Nachrichten standardmäßig unterschreiben*“ auswählen
  - Die Option „*Verschlüsselte Nachrichten standardmäßig unterschreiben*“ auswählen
  - Die Option „*Nachrichten-Entwürfe verschlüsselt speichern*“ auswählen
- Über den Button „*Erweitert ...*“ erreicht man weitere, weniger wichtige Optionen (siehe Kap. 7.2.1):
  - Die Option „*Sende OpenPGP-Schlüssel-ID*“ ist sinnvollerweise zu setzen. Sie legt fest, ob in den Metadaten der Mail die Schlüssel-ID des eigenen Schlüssels mitgesendet wird.
  - Die Option „*Sende URL, um Schlüssel zu empfangen*“ nur dann setzen, wenn man die URL seines Schlüssels auf einem Key-Server kennt. Diese Option ist nicht wichtig.
  - Die Option „*Öffentlichen Schlüssel an Nachrichten anhängen*“ nicht setzen. Hat man seinen öffentlichen Schlüssel auf einen Key-Server hochgeladen, so ist es überflüssig, diesen zusätzlich bei jeder Mail als Anhang mitzusenden.
- Das **HTML-Format für den Mail-Versand abschalten** (siehe Kap. 7.2.3). Ist PGP/MIME als Verschlüsselungsformat deaktiviert (s.o.), dann muss man auf HTML-Mails verzichten und reine Text-Mails senden. In den Konto-Einstellungen unter „*Verfassen & Adressieren*“ ist die Option „*Nachrichten im HTML-Format verfassen*“ abzuwählen.
- **Die eigenen verschlüsselt versendeten Mails lesbar machen** (siehe Kap. 7.2.4). Dies ist keine konten-spezifische, sondern eine globale Option. Sie gilt für alle *Thunderbird*-Mailkonten und ist deshalb auch nicht in den Konten-Einstellungen zu finden. Unter

*Enigmail → Einstellungen... → Senden* ist die Option „*Zusätzlich mit eigenem Schlüssel verschlüsseln*“ auszuwählen. Somit wird eine gesendete Mail, die im eigenen *Gesendet-*Ordner verbleibt, mit dem eigenen öffentlichen Schlüssel verschlüsselt und kann mit dem privaten Schlüssel wieder entschlüsselt werden. Ohne diese Einstellung wären die versendeten Mails nicht lesbar.

- Der **Bestätigungsdialog vor dem Versenden** (siehe Kap. 7.2.1) ist unter *Enigmail → Einstellungen... → Senden* zu konfigurieren. Wählt man die Einstellung „*immer bestätigen*“, wird vor dem endgültigen Versand einer Mail angezeigt, ob sie verschlüsselt und/oder signiert versandt wird oder unverschlüsselt und unsigniert. Dieser Dialog muss vom Benutzer bestätigt werden. Der Mail-Versand kann hier noch abgebrochen werden. Bestätigt man mit „OK“, wird die Mail versandt.
- **Enigmail → Automatisch entschlüsseln/überprüfen:** Ist diese Option aktiviert, dann werden empfangene, verschlüsselte und signierte Mails beim Öffnen automatisch entschlüsselt und einer Signaturprüfung unterzogen, wenn der Schlüssel des Absenders im Schlüsselbund vorliegt. Liegt der Schlüssel des Absenders nicht vor, so bietet Thunderbird an, diesen auf dem Schlüssel-Server zu suchen und ggf. herunterzuladen und in den Schlüsselbund zu importieren.

### Links:

- *Enigmail*-Installationshilfe:  
[http://www.thunderbird-mail.de/wiki/Enigmail\\_OpenPGP#Enigmail\\_installieren](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](http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP)

## 6.4 PGP mit Thunderbird nutzen

Wir haben die Schlüssel im Schlüsselbund erzeugt, bzw. vom Schlüssel-Server importiert. Wir haben die geeigneten Einstellungen in *Thunderbird/Enigmail* vorgenommen. Nach all den Vorbereitungen ist der Versand und Empfang signierter und verschlüsselter Mails nahezu trivial und weicht kaum von der Mail-Nutzung ohne PGP ab.

### 6.4.1 Mailversand

Beim Mailversand kann man bei jeder Mail von den zuvor in Kapitel 6.3 gemachten Vorgaben abweichen. Sind die Vorgaben zweckmäßig getroffen, so ist das nur noch ganz selten erforderlich. Über das *Enigmail*-Menü kann man vor dem Versand mit dem entsprechenden Untermenüpunkt die geeigneten Festlegungen für jede einzelne Mail treffen (siehe Kap. 7.3.1):

- Nachricht unterschreiben
- Nachricht verschlüsseln
- PGP/MIME verwenden

## 6.4.2 Mailempfang

Beim Empfang ist in der Regel nichts weiter zu tun. Mit den in Kapitel 6.3 vorgenommenen Einstellungen werden empfangene, verschlüsselte Mails beim Öffnen automatisch entschlüsselt und damit lesbar gemacht. Außerdem werden signierte Mails einer Signaturprüfung unterzogen. Die PGP-Information wird in einem farbigen Balken oberhalb der Mail angezeigt.

Öffnet man eine signierte Mail, deren Absender-Schlüssel sich nicht im Schlüsselbund befindet, so kann Thunderbird zunächst keine Signaturprüfung vornehmen. Thunderbird bietet an, diesen Schlüssel auf dem Schlüssel-Server zu suchen und ggf. herunterzuladen und in den Schlüsselbund zu importieren. Nimmt man das Angebot an und importiert den neuen Schlüssel, kann Thunderbird die Signatur der Mail prüfen. Wie schon in Kapitel 6.1 beschrieben, sollte man danach den frisch importierten Schlüssel ...

- verifizieren (siehe Kap. 7.1.9.1)
- (nach erfolgreicher Verifizierung) beglaubigen und das Besitzervertrauen festlegen (siehe Kap. 7.1.9)
- und wieder auf den Schlüssel-Server exportieren, damit die gerade vollzogene Beglaubigung unter allen PGP-Nutzern publik wird.

## 6.4.3 Schlüsselbund-Pflege

Von Zeit zu Zeit sollte man die öffentlichen Schlüssel im Schlüsselbund aktualisieren. Sowohl der eigene öffentliche Schlüssel als auch die Schlüssel der Kommunikationspartner können seit der letzten Aktualisierung weitere Beglaubigungen anderer Nutzer erhalten haben. Den Nutzen dieser Beglaubigungen erhalte ich nur, wenn ich meinen Schlüsselbund in regelmäßigen Abständen mit dem Schlüssel-Server synchronisiere. Dies erledigt man in *Thunderbird* unter *Enigmail → Schlüssel verwalten... → Schlüssel-Server → Alle Schlüssel aktualisieren*.

## 6.4.4 Empfängerregeln

In Kap 6.3 wurden einige Standard-Vorgaben für den Mail-Versand getroffen. Mit den Empfängerregeln kann man für einzelne Empfänger abweichende Einstellungen vornehmen. Dies kann bei den Empfängern sinnvoll sein, deren öffentlichen Schlüssel man in den eigenen Schlüsselbund importiert und beglaubigt hat.

Unter *Enigmail → Empfängerregeln → Hinzufügen* lässt sich eine neue Empfängerregel erstellen. Man gibt eine Email-Adresse an, für die die Regel gelten soll. Man definiert, welcher Schlüssel verwendet werden soll (normalerweise der Schlüssel dieses Empfängers) und legt außerdem fest, ob die Mails an den betreffenden Empfänger unterschrieben werden sollen, ob sie verschlüsselt werden soll und ob das Format PGP/MIME dabei zu verwenden ist. Mit einem Klick auf „OK“ speichert man die neue Regel. (Siehe auch Kap. 7.3.3)

## 6.5 PGP auf einem weiteren PC einrichten

Die Einrichtung des zweiten PCs funktioniert grundsätzlich so wie die des ersten – wie in Kapitel 6.1 beschrieben. Allerdings ist hier kein neues Schlüsselpaar zu generieren, sondern das auf dem ersten PC generierte ist wieder zu verwenden. Dazu ist am einfachsten das gesicherte Schlüsselpaar auf dem USB-Stick auf dem zweiten PC in den Schlüsselbund zu importieren.

Die öffentlichen Schlüssel der Kommunikationspartner kann natürlich wieder vom Schlüssel-Server

herunterladen. Einfacher ist es natürlich, sie aus dem Schlüsselbund des ersten PC auf den USB-Stick zu exportieren und auf dem zweiten PC in einem Rutsch wieder zu importieren.

Die folgende Schritt-für-Schritt-Anleitung geht davon aus, dass das eigenen Schlüsselpaar in einer Datei und alle fremden öffentlichen Schlüssel in einer weiteren Datei auf den USB-Stick gesichert wurden (siehe Kap. 6.2).

- USB-Stick mit exportierten Schlüsseln an den Rechner anschließen
- **Eigenes Schlüsselpaar importieren:** In der *Enigmail*-Schlüsselbund-Verwaltung den Menüpunkt *Datei → Importieren... auswählen*, dann auf dem angeschlossenen USB-Stick die Datei mit dem exportierten eigenen Schlüsselpaar auswählen und importieren. (siehe Kap. 7.1.6)
- **Öffentliche Schlüssel der Kommunikationspartner importieren:** In der *Enigmail*-Schlüsselbund-Verwaltung den Menüpunkt *Datei → Importieren... auswählen*, dann auf dem angeschlossenen USB-Stick die Datei mit den exportierten öffentlichen Schlüsseln auswählen und importieren. (siehe Kap. 7.1.6)
- Nach dem erfolgreichen Import der Schlüssel den USB-Stick abmelden und vom Rechner abziehen.

## 6.6 Zusammenfassung

Die Zusammenfassung ist am Ende des ausführlichen Kapitels zur *Thunderbird/Enigmail*-Konfiguration zu finden (siehe Kap. 7.5).

## 6.7 Links zu diesem Kapitel

Die Link-Sammlung ist am Ende des ausführlichen Kapitels zur *Thunderbird/Enigmail*-Konfiguration zu finden (siehe Kap. 7.6).

## 7 PGP auf dem PC – Ausführlicher Einstieg für Wissbegierige

Dieses Kapitel liefert den ausführlichen Einstieg in PGP auf dem PC (mit Windows, Mac OS X oder Linux). Dabei wird vorausgesetzt, dass *Thunderbird* auf dem PC zum Mailversand und -empfang verwendet wird. Es versucht nicht nur die Frage „Wie muss ich vorgehen?“ zu beantworten, sondern auch „Warum ist das so?“ und „Welche anderen Optionen gibt es?“. Die gesamte Beschreibung ist einerseits detaillierter und möchte andererseits das Verständnis für das, was man tut, unterstützen.

### 7.1 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 fremden Schlüssel**. Fremde Schlüssel sind die Schlüssel meiner 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 im Folgenden synonym verwenden und bezeichne damit ein Software-Tool, das die Schlüssel verwaltet, d.h. sie darstellt und bearbeitet, löscht oder neue hinzufügt.

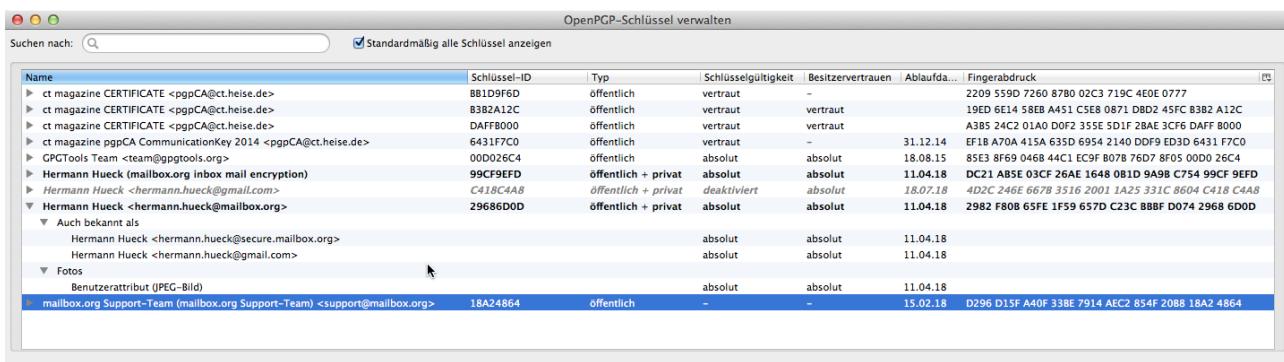


Abbildung 10: Thunderbird: Der Enigmail-Schlüsselbund mit einigen Schlüsseln

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

#### 7.1.1 Tools zur Schlüsselverwaltung

Wir benötigen folgende Tools auf unserem Rechner:

- Ein natives (d.h. ein plattform-spezifisches) Schlüsselverwaltungstool:
  - **Gpg4win** unter Windows; Download unter <http://www.gpg4win.de/> oder unter <http://www.heise.de/download/gpg4win-e2d914fd06f82399ae36052e8362b95c-1398246471-2619466.html>
  - **GPG Suite** von *GPGTools* unter Mac OS X. Download unter <https://gpgtools.org/> oder bei Heise unter <http://www.heise.de/download/gpg-suite-a8929973af027b9846bd8eeb330da2d4-1398337947-2678714.html>.  
Installiert man die *GPG Suite*, so kann man damit auch PGP-Verschlüsselung mit dem Apple Mail-Client, der auf jedem Mac vorinstalliert ist, realisieren. Dies wird jedoch im vorliegenden Dokument nicht weiter erläutert, da diese Lösung nur auf dem Mac funktioniert. Das beschriebene Vorgehen mit Thunderbird kann auf jedem PC – unabhängig vom verwendeten Betriebssystem (Windows, Mac OS X oder Linux) – eingesetzt werden.
  - **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 Suite* 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 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. Zweckmäßig ist es, wenn man das native, plattform-spezifische Schlüsselverwaltungstool (*Gpg4win* bzw. *GPG Suite*) schon vorher installiert hat, 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](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 *Enigmail*. Außerdem gibt es

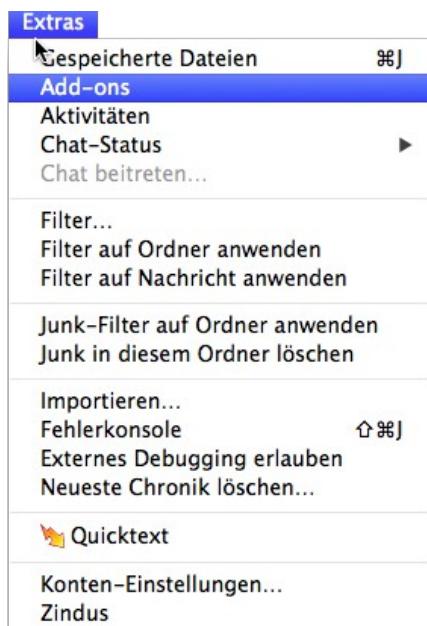


Abbildung 11: Thunderbird:  
Verwaltung der Add-ons öffnen

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 Suite* 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 detaillierte Anleitungen, Tutorials und FAQs im Web zu finden sind. Folgende URLs sind zu empfehlen:

- *Thunderbird* und *Enigmail OpenPGP*:  
[http://www.thunderbird-mail.de/wiki/Enigmail\\_OpenPGP](http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP)
- *Gpg4win*:  
<http://gpg4win.de/handbuecher/einsteiger.html>
- *GPG Suite*:  
<http://support.gpgtools.org/kb>

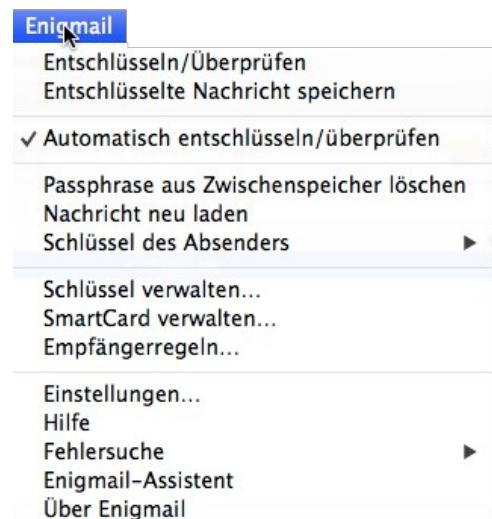


Abbildung 12: Thunderbird nach der Enigmail-Installation: Das neue Enigmail-Menü

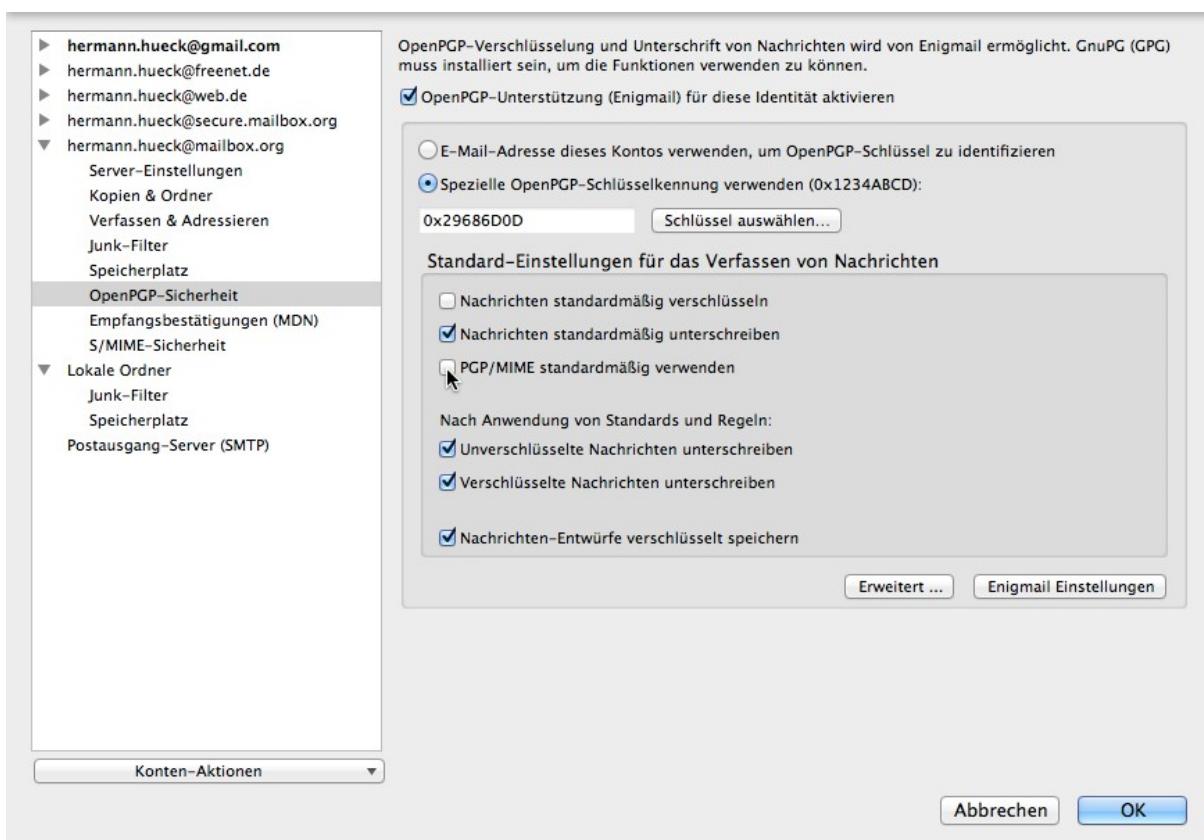


Abbildung 13: Thunderbird: Konten-spezifische Enigmail-Konfiguration

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 (unter Windows und Mac OS X) die Verschlüsselung auch ohne die Verwendung der Kommandozeile realisieren.

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

## 7.1.2 Erzeugung eines neuen Schlüsselpaars

Ein neues Schlüsselpaar kann mit *Thunderbird/Enigmail* erzeugt werden. Allerdings wird dabei immer ein RSA-Schlüssel mit einer Länge von 2048 Bit erzeugt. Für die Erzeugung eines noch sichereren Schlüssels mit einer Länge 4096 Bit empfehle ich die Verwendung der Kommandozeile oder der nativen Schlüsselbund-Verwaltung. Die Schlüsselerzeugung in *Enigmail* beschreibe ich dennoch.

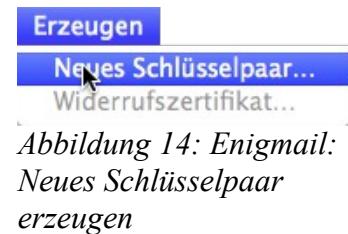


Abbildung 14: Enigmail:  
Neues Schlüsselpaar  
erzeugen

### 7.1.2.1 Schlüsselerzeugung mit *Enigmail / Thunderbird*

Hier startet man in *Thunderbird* über den Menüpunkt *Enigmail* → *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

- 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

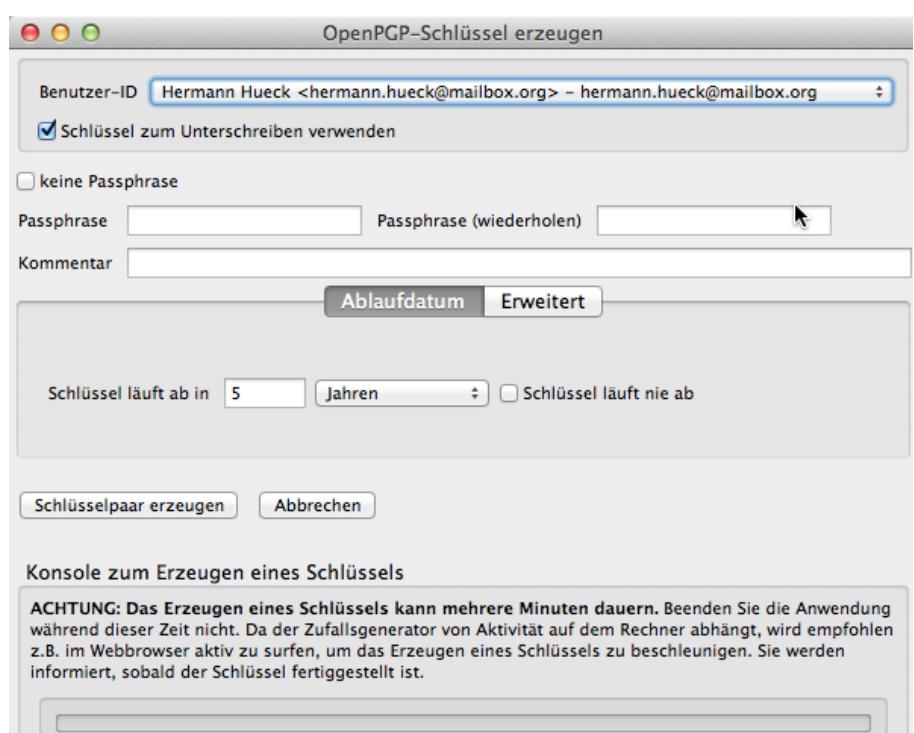


Abbildung 15: Enigmail: Dialog zum Erzeugen eines neuen Schlüsselpaars

Missbrauch. Die Passwortregeln (siehe Kap. 12.1) gelten auch hier. Der Schlüssel wird nach der Erstellung in der Schlüsselliste angezeigt.

- Einen optionalen Kommentar
- Die Gültigkeitsdauer des Schlüssels

Mit einem Klick auf *Schlüssel erzeugen* wird ein Schlüssel für das angegebene Email-Konto erzeugt. (Dieser Schlüssel ist ein RSA-Schlüssel mit einer Länge von 2048 Bit.)

Anschließend wird man aufgefordert, ein Widerrufszertifikat zu erzeugen. Dies lässt sich auch später noch nachholen. (siehe Kap. 7.1.3)

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

OpenPGP-Schlüssel verwalten						
Suchen nach:		<input checked="" type="checkbox"/> Standardmäßig alle Schlüssel anzeigen				
Name	Schlüssel-ID	Typ	Schlüsselgültigkeit	Besitzervertrauen	Ablaufda...	Fingerabdruck
ct magazine CERTIFICATE <pgpCA@ct.heise.de>	D4FFB000	öffentlich	vertraut	vertraut	A3B5 24C2 01A0 D0F2 355E 5D1F 2BAE 3CF6 DAFF B000	
ct magazine pgpCA CommunicationKey 2014 <pgpCA@ct.heise.de>	6431F7C0	öffentlich	vertraut	–	E1B A70A 415A 635D 6954 2140 DDF9 ED3D 6431 F7C0	
GPGTools Team <team@gpgtools.org>	00D026C4	öffentlich	absolut	absolut	18.08.15 85E3 BF69 0468 44C1 EC9F B078 76D7 8F05 00D0 36C4	
Hermann Hueck (mailbox.org inbox mail encryption)	99CF9EFD	öffentlich + privat	absolut	absolut	11.04.18 DC21 AB5E 03CF 26AE 1648 081D 9A9B C754 99CF 9EFD	
Hermann Hueck <hermann.hueck@mail.com>	C418C4A8	öffentlich + privat	absolut	absolut	18.07.18 4D2C 246E 667B 3516 2001 1A25 331C 8604 C418 C4A8	
<b>Hermann Hueck &lt;hermann.hueck@mailbox.org&gt;</b>	<b>29686D0D</b>	<b>öffentlich + privat</b>	<b>absolut</b>	<b>absolut</b>	<b>11.04.18 2982 F808 65FE 1F59 657D C23C BBBF D074 2968 6D0D</b>	
mailbox.org Support-Team (mailbox.org Support-Team) <support@mailbox.org>	18A24864	öffentlich	–	–	15.02.18 D296 D15F A40F 33BE 7914 AEC2 854F 2088 18A2 4864	

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

Unter folgendem Weblink wird die Schlüsselgenerierung mit *Thunderbird/Enigmail* sehr detailliert erläutert:

[http://www.thunderbird-mail.de/wiki/Enigmail\\_OpenPGP\\_-\\_Schl%C3%BCsselverwaltung#Ein\\_Schl.C3.BCsselpaar\\_erzeugen](http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Schl%C3%BCsselverwaltung#Ein_Schl.C3.BCsselpaar_erzeugen)

### 7.1.2.2 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 Suite* (Mac OS X) oder die Kommandozeile für die Schlüsselgenerierung nutzen. Bei der Schlüsselerzeugung stehen dann mehr Optionen zur Verfügung, z.B. die Festlegung der Schlüssellänge oder des Ablaufdatums des Schlüssels.

- 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* die plattform-spezifischen Tools *Gpg4win* bzw. *GPG Suite* oder auch 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. Für diesen Fall ist *Enigmail* nicht das

geeignete Werkzeug. Mit *Gpg4win* oder *GPG Suite* oder auf der Kommandozeile ist dies kein Problem.

### 7.1.2.3 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](http://gpg4win.de/handbuecher/einsteiger_7.html).

### 7.1.2.4 Schlüsselerzeugung mit der *GPG Suite*

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

### 7.1.2.5 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
```

## 7.1.3 Das Widerrufszertifikat

Unbedingt sollte man zu jedem eigenen Schlüssel ein **Widerrufszertifikat (KRC, Key Revocation Certificate)** erzeugen und an einem sicheren Ort (CD oder USB-Stick, der nur für Schlüssel verwendet wird) speichern. Mit dem Widerrufszertifikat kann man einen öffentlichen Schlüssel widerrufen, auch wenn man die Passphrase nicht kennt (z.B. wenn man die Passphrase vergessen hat).

Erlangt eine fremde Person Zugriff auf das Widerrufszertifikat (z.B. wenn ein Hacker in den eigenen Rechner einbricht und das Widerrufszertifikat stiehlt), so kann diese den Schlüssel, für den das Zertifikat erstellt wurde, widerrufen. Der widerrufene Schlüssel wird damit unbrauchbar.

Die Erstellung eines Widerrufszertifikates (in einer externen Datei) kann man mit Enigmail erledigen oder mit den Plattform-spezifischen 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.

Das exportierte Widerrufszertifikat sollte auf einem externen Speichermedium gesichert und dann vom Rechner gelöscht werden (siehe Kap. 7.1.6.1).

### 7.1.4 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.

Das Kommando

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

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

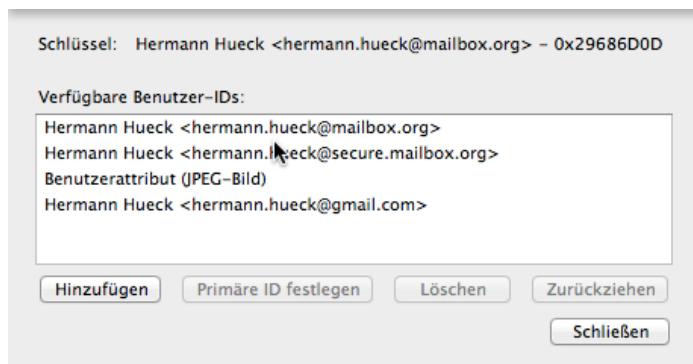


Abbildung 17: Enigmail: Liste der Benutzer-IDs eines Schlüssels

### 7.1.5 Die wichtigsten PGP-Schlüsseleigenschaften

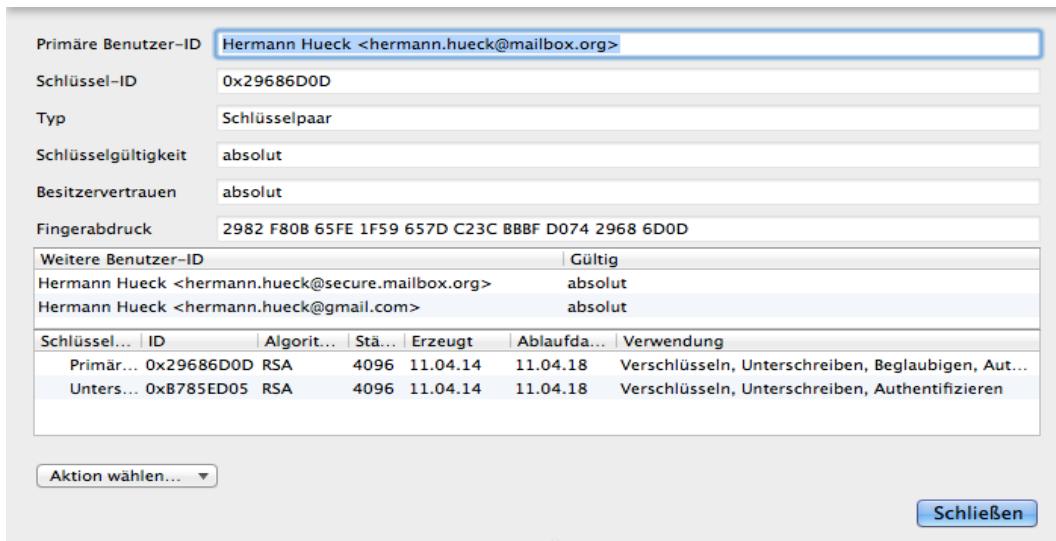


Abbildung 18: Enigmail: Schlüsselattribute

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 Schüssel. Damit lässt sich auch die Zusammengehörigkeit der beiden Schlüssel eines Schlüsselpaares 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 Benutzer-ID 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.

## 7.1.6 Schlüssel exportieren und importieren

In der Schlüsselverwaltung kann man einen Schlüssel in eine Datei exportieren, um ihn z.B. 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.

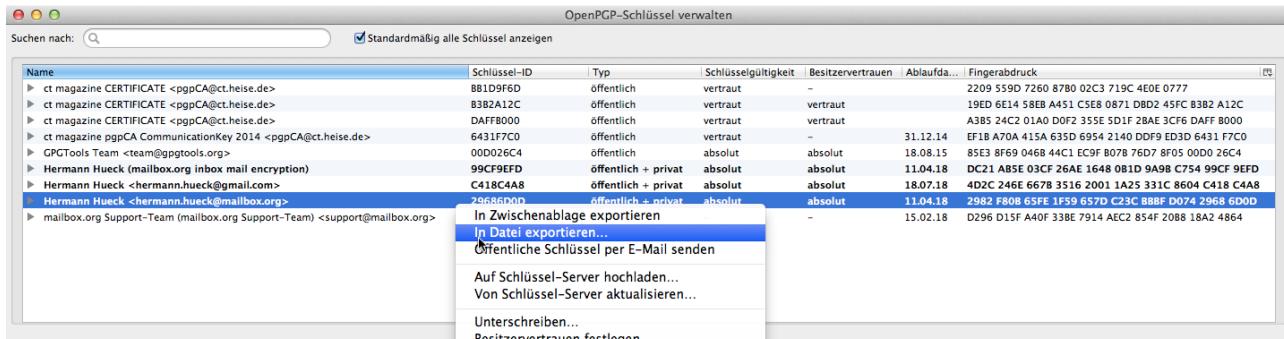


Abbildung 19: Enigmail: Den markierten Schlüssel exportieren

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.

### 7.1.6.1 Schlüssel sicher aufbewahren

Das eigene Schlüsselpaar (bzw. die eigenen Schlüsselpaare, so man mehrere hat) sollte man nicht verlieren. Damit es auch einen Festplattencrash überlebt, muss es exportiert und auf ein externes Medium (USB-Stick, Speicherkarte oder CD) gesichert und gut aufbewahrt werden. Das Widerrufszertifikat kann man dabei gleich mit sichern. Nach der erfolgreichen Sicherung sollte das exportierte Schlüsselpaar und das Widerrufszertifikat von der Festplatte des Rechners gelöscht werden.

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. (Mit einem Ende-zu-Ende verschlüsselnden Cloud-Dienst (siehe Kap. 13.3.2.5 und 13.3.2.6) wäre dies allerdings ein gangbarer und sicherer Weg der Schlüsselübertragung.)

Was für den exportierten privaten Schlüssel gilt, gilt genau so für das Widerrufszertifikat.

## 7.1.7 Konfiguration der Key-Server (Enigmail)

Bevor man den/die (öffentlichen) Schlüssel mit einem Schlüssel-Server synchronisiert (siehe Kap. 7.1.8), 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 *Enigmail* → *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

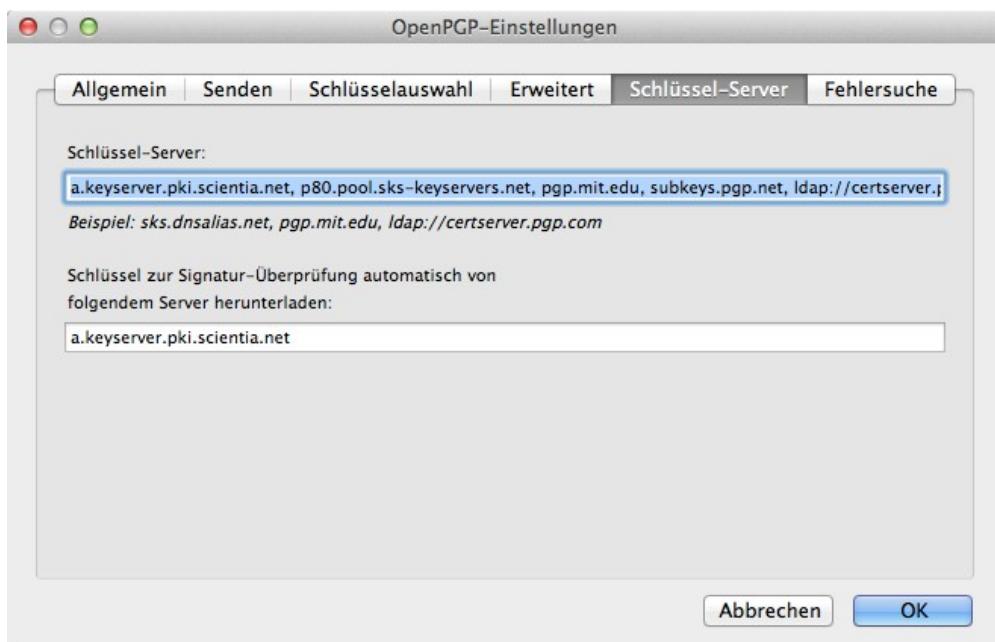


Abbildung 20: *Enigmail*: Konfiguration der Schlüssel-Server

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. die *GPG Suite* 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.

### 7.1.8 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. Zweckmäßig ist es, man gibt 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 dauerte er mehr als eine Stunde.

### 7.1.9 Schlüssel beglaubigen



Abbildung 22: Enigmail: Den markierten Schlüssel unterschreiben/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. 5.3).

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. 7.1.8). Nach dem Import muss ich diesen Schlüssel ebenfalls signieren und als Besitzervertrauen „vollständiges Vertrauen“ eintragen.

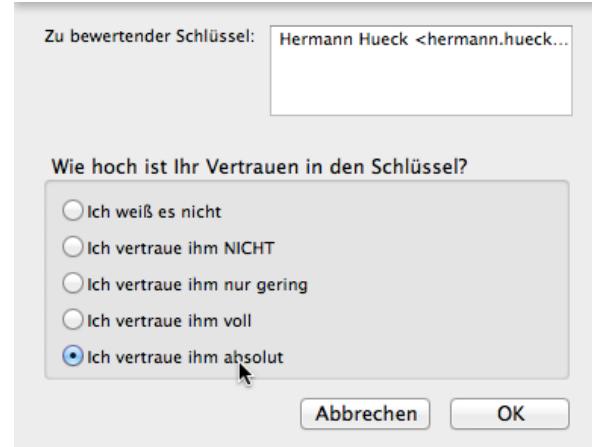


Abbildung 23: Enigmail: Definieren des Besitzervertrauens

#### 7.1.9.1 Verifikation von Schlüssel-Identität und Benutzer-Identität

Bevor ich einen fremden Schlüssel (den öffentlichen Schlüssel eines Kommunikationspartners) beglaubige, muss ich ihn 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. 7.1.8).

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 die Freundin oder den 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 muss auch der Fingerabdruck des Schlüssels geprüft werden (s.u.).
- **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 das Papier geschriebene Fingerabdruck mit dem des importierten, öffentlichen Schlüssels übereinstimmt. Ist der Schlüsselinhhaber ein guter Bekannter, den ich an der Stimme erkennen kann, kann er den 40-stelligen Fingerabdruck dem Beglaubigenden auch am Telefon vorlesen.

Wenn man in der *Enigmaile*-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 erwünscht. Will man nicht alle, sondern nur eine bestimmte Benutzer-ID (sprich: Email-Adresse) eines Schlüssels beglaubigen, so muss man dies in der plattform-spezifischen Schlüsselverwaltung *Gpg4win* oder *GPG Suite* 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 beglaubigt ihn, d.h. er signiert ihn. Danach lädt er meinen von ihm beglaubigten öffentlichen Schlüssel auf einen Key-Server hoch (siehe Kap. 7.1.8). Ich und andere Benutzer können ihn von dort wieder aktualisieren (sprich: herunterladen); so haben wir die 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 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.

### 7.1.9.2 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. 7.1.8) und schickt ihn an die Heise-Mail-Adresse [pgpCA@ct.heise.de](mailto: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 verschlüsselte Bestätigungsmaile. Für einen Schlüssel mit drei Benutzer-IDs erhält man drei Bestätigungsmaile. 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 Bestätigungsmaile befindet sich ein Bestätigungslink. (Wäre man nicht der Inhaber des privaten Schlüssels, so könnte man die Email des Heise-Verlags nicht entschlüsseln, den Bestätigungslink also auch nicht lesen und ihn demzufolge auch nicht anklicken.)
- 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 des 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. 7.1.8).

- Den Schlüssel des c't-Magazins, mit dem der eigene Schlüssel signiert/beglaubigt wurde, muss man von einem Key-Server herunterladen (siehe Kap. 7.1.8) 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 und bei den deutschen PGP-Benutzern geschätzte Institution ist, ist der Wert dieser Schlüssel-Beglaubigung sehr hoch einzustufen. D.h. wenn Heise meinen Schlüssel beglaubigt, dann genießt auch mein von Heise beglaubigter Schlüssel ein sehr hohes Vertrauen.

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>.

## 7.2 Thunderbird für PGP-Nutzung konfigurieren

Der ganze Aufwand der Schlüsselverwaltung und -beglaubigung dient letztlich dazu, signierte und verschlüsselte Mails versenden zu können. Außerdem können verschlüsselte Mails empfangen, einer Signaturprüfung unterzogen, entschlüsselt und dann auch gelesen werden. 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.

Nach der Festlegung der globalen Einstellungen (in Kap. 7.2.1) aktiviert man PGP für ein bestimmtes Mail-Konto und ordnet dem Konto ein Schlüsselpaar zu (siehe Kap. 7.2.2). Danach sind die konten-spezifischen Einstellungen für den Mailversand konfigurieren.

### 7.2.1 Globale Enigmail-Einstellungen für alle Konten

Die hier zu treffenden Einstellungen gelten für alle Konten, für die PGP aktiviert und einen privaten Schlüssel festgelegt hat (siehe Kap. 7.2.2).

Unter *Enigmail* → *Einstellungen...* → *Allgemein* sollte man zunächst die *Experten-Optionen und -Menüpunkte anzeigen*.

Unter *Enigmail* → *Einstellungen...* → *Senden* definiert man die konten-übergreifenden Einstellungen für den Mailversand. Hier ist es zweckmäßig, die

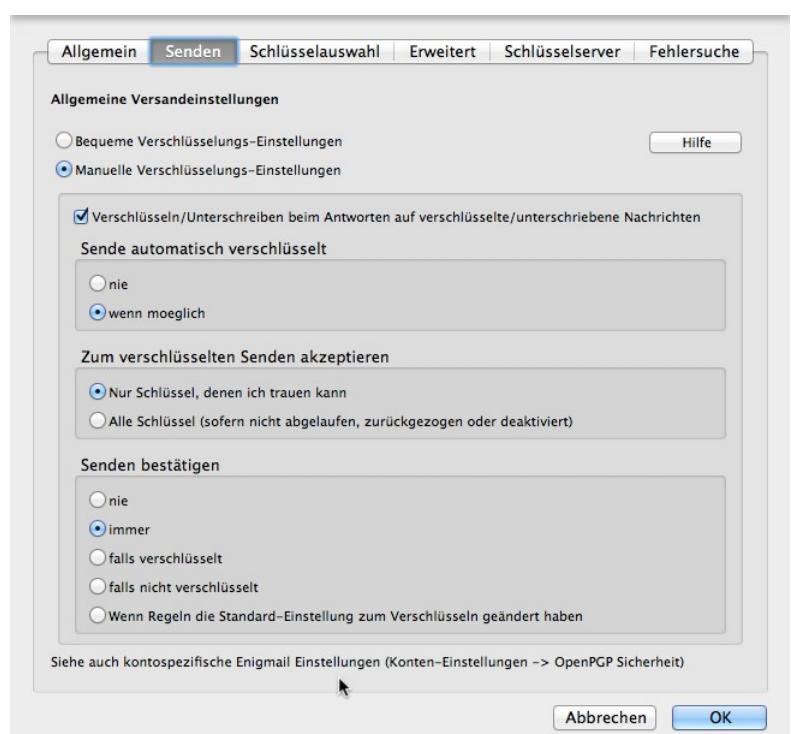


Abbildung 24: Enigmail: Globale Einstellungen für den Mailversand

manuellen Einstellungen zu wählen.

- Man definiert, unter welchen Umständen beim Antworten auf eine verschlüsselte oder signierte Mail die Antwort-Mail ebenfalls verschlüsselt, bzw. signiert werden soll.
- Man legt fest, welche Schlüssel zum verschlüsselten Senden verwendet werden sollen – alle Schlüssel im Schlüsselbund oder nur diejenigen, die man als vertraut markiert hat (siehe Kap. 7.1.9).
- Schließlich kann man noch festlegen, unter welchen Umständen *Thunderbird* vor dem Versand dem Benutzer eine Versand-Bestätigung abverlangen soll. Bei der Einstellung immer wird dem Benutzer jedes Mal vor dem Versenden angezeigt, ob die Mail unverschlüsselt oder verschlüsselt, ob sie signiert oder unsigniert ist und ob das Verschlüsselungsformat PGP/MIME verwendet wird oder nicht. Erst wenn man den Mailversand mit den angezeigten Einstellungen bestätigt, wird die Mail wirklich versandt.

### 7.2.2 Enigmail-Einstellungen für jedes Mail-Konto

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

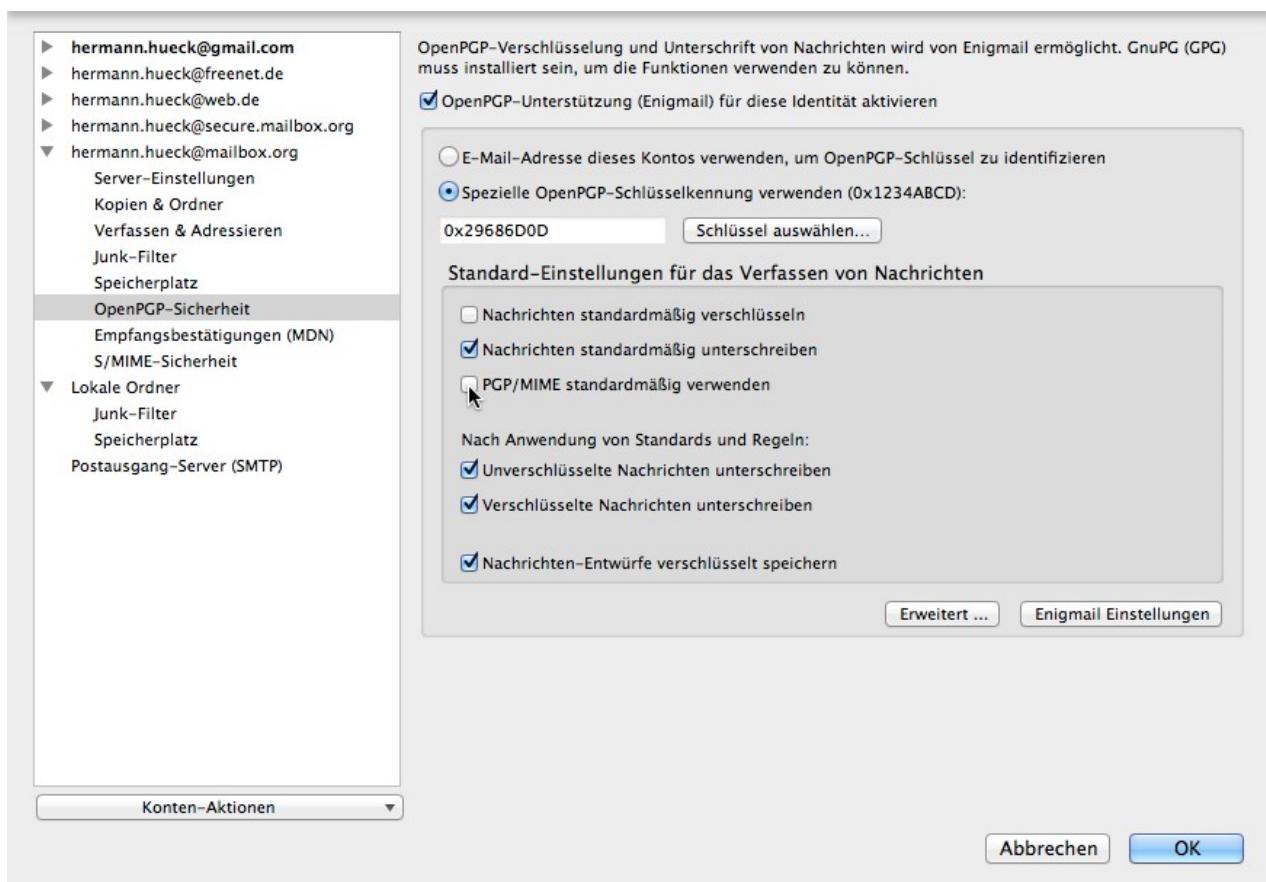


Abbildung 25: Enigmail: PGP-Einstellungen für ein Mail-Konto

Die konten-spezifischen Einstellungen überschreiben ggf. die globalen Einstellungen, die in Kapitel 7.2.1 vorgenommen wurden.

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

[http://www.thunderbird-mail.de/wiki/Enigmail\\_OpenPGP - Einstellungen#OpenPGP-Sicherheit](http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP - Einstellungen#OpenPGP-Sicherheit)

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 Einstellungen sind die Standardeinstellungen für den Mail-Versand, die aus diesem Konto abgeschickt werden. Sie können auch vor dem Versenden einer Mail noch geändert werden:

- *Nachrichten standardmäßig verschlüsseln*. Dies ist in der Regel nur sinnvoll, wenn man viele öffentliche Schlüssel von Kommunikationspartnern im eigenen Schlüsselbund hat. Beginnt man gerade mit der Verwendung von PGP, ist diese Voreinstellung nicht zu empfehlen.
- *Unverschlüsselte Nachrichten standardmäßig unterschreiben*. Dies ist sinnvoll, falls man meist unterschriebene Nachrichten versenden will.
- *Immer PGP/MIME verwenden*. Diese Option würde ich abwählen. Zurzeit empfehle ich noch den Verzicht auf die Verwendung von PGP/MIME. Damit wird dann das klassische, sogenannte Inline-PGP verwendet.

Bei Inline-PGP wird die Mail mit allen ihren Anhängen in einem Paket verschlüsselt. Mit PGP/MIME werden der Mail-Inhalt und jeder Anhang separat verschlüsselt. Eine mit PGP/MIME verschlüsselte Mail besteht aus mehreren verschlüsselten Teilen. PGP/MIME ist das modernere Format für den Mailversand. Da es noch viele Tools gibt, die dies nicht unterstützen, ist aktuell eher davon 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 Android-Smartphone oder -Tablet unterstützen dieses Format aktuell noch nicht (siehe Kap. 10.3.2).

Wird statt des Format PGP/MIME Inline-PGP verwendet, so muss man auch auf die HTML-Formatierung der Mails verzichten (siehe Kap. 7.2.3). Die HTML-Formatierung bringt das klassische Inline-PGP aus dem Tritt.

- *Unverschlüsselte Nachrichten unterschreiben*. Diese Einstellung greift erst, wenn Empfängerregeln und die Standardvorgaben ausgewertet wurden und aus dieser Auswertung keine Entscheidung bezüglich der Signierung unverschlüsselter Nachrichten getroffen werden konnte.
- *Verschlüsselte Nachrichten standardmäßig unterschreiben*. Diese Einstellung greift erst, wenn Empfängerregeln und die Standardvorgaben ausgewertet wurden und aus dieser Auswertung keine Entscheidung bezüglich der Signierung verschlüsselter Nachrichten getroffen werden konnte.
- *Nachrichten-Entwürfe verschlüsselt speichern*. Diese Einstellung legt fest, ob Nachrichten-Entwürfe verschlüsselt oder unverschlüsselt gespeichert werden sollen.

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.

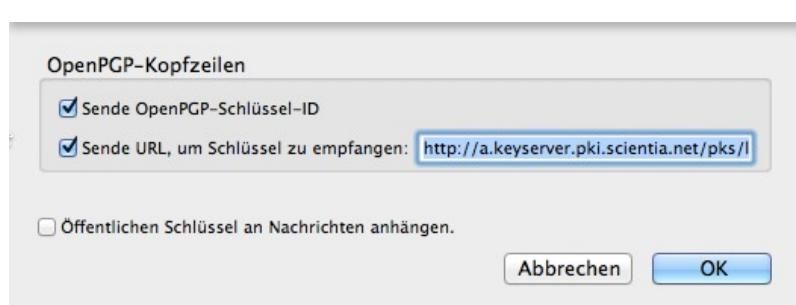


Abbildung 26: Enigmail: Weitere PGP-Optionen

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 Anhang hinzugefügt. Damit kann man den eigenen öffentlichen Schlüssel an die Kommunikationspartner verteilen. Das ist jedoch nicht 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 vom Schlüssel-Server herunter und importiert ihn. Man muss den Schlüssel also nicht mit jeder Mail mitschicken.

### 7.2.3 Text-Mails statt HTML-Mails

Beim Verzicht auf PGP/MIME (siehe Kap. 7.2.1) wird automatisch das ältere Verschlüsselungsformat Inline-PGP verwendet. Dabei kann es Probleme mit HTML-Mails geben, die dazu führen können, dass 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 ist normalerweise eingeschaltet. Durch Entfernen des Häkchens in der Checkbox wird das Text-Format eingestellt.

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

Aktiviert man PGP/MIME (siehe Kap. 7.2.1), kann man für das Verfassen der Mails auch das HTML-Format verwenden.

### 7.2.4 Die verschlüsselt versendeten Mails lesbar machen

Eine weitere *Enigmail*-Einstellung verhindert, dass man die eigenen verschlüsselt versendeten Mails nicht mehr lesen kann.

Diese Einstellung ist in *Thunderbird* zu finden unter *Enigmail → 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.

### 7.2.5 Mails automatisch entschlüsseln/überprüfen

Der Menüpunkt *Enigmail → Automatisch entschlüsseln/überprüfen* tut, was er sagt. Ist diese Option aktiviert, dann werden empfangene, verschlüsselte und signierte Mails beim Öffnen automatisch entschlüsselt und einer Signaturprüfung unterzogen, wenn der Schlüssel des Absenders im Schlüsselbund vorliegt. Liegt der Schlüssel des Absenders nicht vor, so bietet Thunderbird an, diesen auf dem Schlüssel-Server zu suchen und ggf. herunterzuladen und in den Schlüsselbund zu importieren.

## 7.3 PGP mit Thunderbird nutzen

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.

### 7.3.1 Mailversand

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 *Enigmail*. Beim Klick auf das Symbol lassen sich die PGP-Versand-Optionen anzeigen oder ändern. Man kann bestimmen, ob vor dem Versand ...

- 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,
- ob PGP/MIME als Verschlüsselungsformat verwendet werden soll
- und ob den Schlüsseln aller Empfänger dieser Mail vertraut werden soll.

Nachricht wird nicht verschlüsselt	►
Nachricht wird unterschrieben	►
PGP/MIME wird verwendet	►
Temporär den Schlüsseln aller Empfänger vertrauen	

Abbildung 27: PGP-Versand-Optionen

### 7.3.2 Mailempfang

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 (siehe Kap. 7.2.5).

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

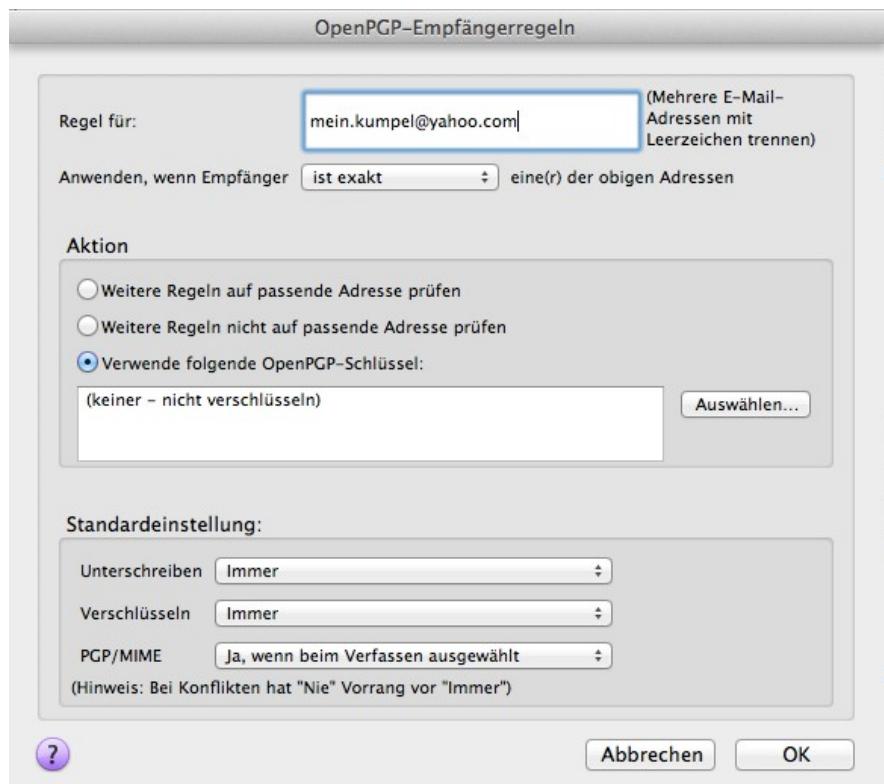


Abbildung 28: Neue Empfängerregel erstellen

### 7.3.4 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. Die geeigneten Voreinstellungen haben wir in Kapitel 7.2.1 bereits konfiguriert.

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 Suite* 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 und kann sie signieren/beglaubigen, die Vertrauensstufe – wie zuvor beschrieben – auf ein adäquates Level einstellen und so allmählich den Anteil der verschlüsselten Kommunikation erhöhen.

### 7.3.5 Pflege des Schlüsselbundes

Von Zeit zu Zeit sollte man die öffentlichen Schlüssel im Schlüsselbund aktualisieren. Sowohl der eigene öffentliche Schlüssel als auch die Schlüssel der Kommunikationspartner können seit der letzten Aktualisierung weitere Beglaubigungen anderer Nutzer erhalten haben. Den Nutzen dieser Beglaubigungen erhalte ich nur, wenn ich meinen Schlüsselbund in regelmäßigen Abständen mit dem Schlüssel-Server synchronisiere. Dies erledigt man in *Thunderbird* unter *Enigmail* → *Schlüssel verwalten...* → *Schlüssel-Server* → *Alle Schlüssel aktualisieren*.

Da die Schlüsselbund-Pflege auf dem Android-Gerät nicht so komfortabel zu bewerkstelligen ist, kann man die aktualisierten öffentlichen Schlüssel exportieren (siehe Kap. 7.1.6), die Datei mit den Schlüsseln auf das Android-Gerät übertragen (siehe Kap. 10.2.2) und dort wieder in die Schlüsselverwaltung importieren (siehe Kap. 10.2.3).

## 7.4 Verschlüsselte Mails auf dem Zweitrechner

Möchte man auf einen zweiten Rechner ebenfalls auf die verschlüsselte Email-Kommunikation zugreifen, so muss man die gesamte beschriebene Einrichtungsprozedur auf diesem System wiederholen.

Dies ist nur sinnvoll, wenn man die Mail-Konten auf dem Erstrechner bereits mit IMAP konfiguriert hat. Dabei bleiben die Mails zentral beim Provider gespeichert. Mit IMAP können beliebig viele eigene Geräte, auch Smartphones und Tablets für den Zugriff auf den Mail-Account eingerichtet werden. Auch auf dem Zweitrechner ist der Abruf der Mails nicht mit POP sondern mit IMAP zu konfigurieren.

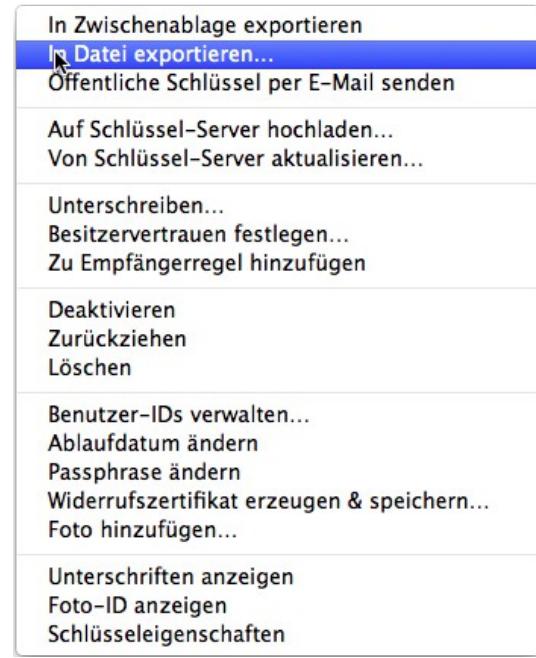


Abbildung 29: Enigmail: Schlüssel-Export

**Auf dem Zweitrechner erzeugt man keine neuen Schlüssel.** Stattdessen überträgt man den gesamten Schlüsselbund des Erstrechners auf den zweiten. 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 Rutsch zu auf dem ersten PC zu exportieren, auf den zweiten zu übertragen und dort wieder zu importieren.

Dabei kann man so vorgehen:

- Die *Enigmail*-Schlüsselverwaltung auf dem ersten PC 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 und in eine Datei speichern will.
- Die fremden Schlüssel (Schlüssel-Typ: öffentlich) markieren.  
Dann im Kontextmenü *In Datei exportieren ...* auswählen. Schließlich in eine Datei speichern.
- Beide Dateien auf den anderen Rechner übertragen. Wichtig ist dabei, dass die Übertragung auf einen **sicheren Übertragungsweg** 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 einen verschlüsselten Cloud-Speicher kommt durchaus in Frage (siehe Kap. 13.3.2.5 und 13.3.2.6).)
  - 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, Speicherplatte, CD), das ja auch als Sicherungsmedium dienen kann. 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 von den Festplatten der beiden Rechner löschen.

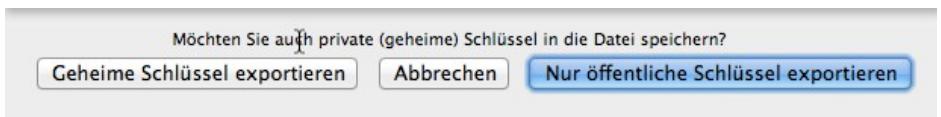


Abbildung 30: *Enigmail: Auch die geheimen Schlüssel exportieren?*

Wenn die Schlüssel im Schlüsselbund des Zweitrechners vorliegen, kann man dort die Zuordnung der privaten Schlüssel zu den eingerichteten Mail-Konten und die konten-spezifische PGP-Konfiguration – wie in Kapitel 7.2 beschrieben – vornehmen.

## 7.5 Zusammenfassung

In diesem Kapitel ging es um die Nutzung von PGP mit *Thunderbird*. Wenn alles eingerichtet ist, so ist das Empfangen und Versenden signierter und chiffrierter Emails recht einfach. Die Vorbereitung dazu, die Einrichtung vor der ersten Nutzung, ist allerdings nicht mit drei Mausklicks erledigt.

Wir mussten zunächst ein plattform-spezifisches Schlüsselverwaltungsprogramm (*Gpg4win* oder *GPG Tools Suite*) installieren. Dies war die Voraussetzung, damit das *Thunderbird*-Add-on *Enigmail* funktioniert. *Enigmail* arbeitet dann in gleicher Weise auf allen PC-Betriebssystemen. Man hat also zwei Tools, die jedoch auf denselben Schlüsselbund zugreifen.

Mit *Enigmail* kann man nur einen 2048-Bit-RSA-Schlüssel erzeugen. Möchte man einen stärkeren Schlüssel, z.B. einen Schlüssel mit einer Länge von 4096 Bit, so muss man bei der Erzeugung des Schlüssels zum plattform-spezifischen Tool oder zur Kommandozeile greifen. Bei allen weiteren Schlüsselverwaltungsarbeiten ist das in *Thunderbird* integrierte *Enigmail* völlig ausreichend.

Bei der Schlüsselerzeugung oder nachträglich kann man ein Widerrufszertifikat erstellen und in eine externe Datei (außerhalb des Schlüsselbundes) speichern. Sollte man seinen privaten Schlüssel einmal verloren haben, so kann man den öffentlichen Schlüssel dennoch mit dem Widerrufszertifikat für ungültig erklären und damit unbrauchbar machen.

Bei der Schlüsselerzeugung wird dem Schlüssel normalerweise eine Benutzer-ID und damit auch Email-Adresse zugeordnet. Will man denselben Schlüssel für mehrere Email-Adressen verwenden, so kann dem Schlüssel weitere Benutzer-IDs mit jeweils einer Email-Adresse hinzufügen.

Wir haben uns dann in Kapitel 7.1.5 die Eigenschaften eines PGP-Schlüssels angesehen.

Um die Schlüssel auf einem externen Speichermedium zu sichern, kann man sie in Dateien auf einem externen Speichermedium exportieren. Muss man die Schlüssel wiederherstellen (z.B. nach einer Neuinstallation des Rechners), so kann man sie aus den Dateien auf dem Speichermedium wieder importieren. Auf einem externen Medium gesicherte Schlüssel lassen sich auch auf einem Zweitrechner oder auf einem Smartphone oder einem Tablet wieder importieren.

Der Austausch öffentlicher Schlüssel verschiedener Benutzer erfolgt normalerweise über allgemein zugängliche Key Server. Es genügt, den eigenen Schlüssel auf einen Schlüssel-Server hochzuladen, da dieser seinen Schlüsselbestand mit anderen Key Servern abgleicht. Zunächst sollte man die von der Schlüsselverwaltung (sowohl der plattform-spezifischen als auch *Enigmail*) zu verwendenden Key Server konfigurieren. Sind diese konfiguriert, so kann man in der Schlüsselverwaltung den eigenen öffentlichen Schlüssel auf einen Key Server hochladen oder die Schlüssel von Kommunikationspartnern, die ebenfalls PGP einsetzen, von dort herunterladen.

Vertraut man dem Schlüssel eines Kommunikationspartners, dann ist es zweckmäßig, diesen öffentlichen Schlüssel zu beglaubigen/signieren. Zuvor sollte man allerdings prüfen, ob der betreffende Schlüssel und die zugeordneten Email-Adressen der betreffenden Person gehören. Nach der Verifikation signiert man also den öffentlichen Schlüssel des Kommunikationspartners mit dem eigenen privaten Schlüssel, man ordnet ihm eine Vertrauensstufe (in der Regel "Volles Vertrauen") zu und lädt den beglaubigten Schlüssel wieder auf den Key Server hoch. Der Kommunikationspartner sollte mit dem eigenen öffentlichen Schlüssel genauso verfahren. Den beglaubigten öffentlichen Schlüssel kann man nun wieder in den eigenen Schlüsselbund reimportieren.

Die c't-Kryptokampagne erlaubt es, den eigenen öffentlichen Schlüssel vom Heise-Verlag beglaubigen zu lassen. Damit erlangt der eigene Schlüssel auch eine höhere Glaubwürdigkeit bei anderen Benutzern, wenn diese dem Schlüssel des Heise-Verlags vertrauen. Der Fingerabdruck des

Heise-Schlüssels ist in jeder c't abgedruckt ist und kann somit einfach und von jedem verifiziert werden.

Bis hierher haben wir uns nur mit der Verwaltung der Schlüssel beschäftigt, mit deren Erzeugung, Export in bzw. Import aus Dateien, mit der Schlüssel-Synchronisation zwischen dem Schlüsselbund und den Key Servern und mit der Beglaubigung von öffentlichen Schlüsseln der Kommunikationspartner. Nun geht es um die Konfiguration von *Thunderbird/Enigmail* für die Verwendung der Schlüssel beim Versand und Empfang von Nachrichten. Die meisten Einstellungen betreffen den Versand, für den Empfang lässt sich nur einstellen, ob die empfangenen Nachrichten automatisch oder erst nach Aufforderung des Benutzers entschlüsselt werden sollen.

Für den Nachrichten-Versand hat *Enigmail* eine mehrstufige Konfiguration, bei der die spezifische Konfiguration immer die allgemeinere übertrumpft.

- In der globalen Konfiguration werden die Einstellungen für alle *Thunderbird*-Mail-Konten getroffen.
- Die konten-spezifische Konfiguration ist für jedes *Thunderbird*-Mail-Konto vorzunehmen. Sie überschreibt ggf. die globalen Einstellungen. Hier muss man PGP erst aktivieren und dem Konto einen Schlüssel zuordnen, damit PGP mit dem betreffenden Konto verwendet werden kann.
- In den Empfängerregeln kann man für jeden Empfänger spezielle Einstellungen vornehmen. Diese können die globalen und die konten-spezifischen Einstellungen überschreiben.
- Die nachrichten-spezifischen Einstellungen können alle anderen überschreiben. Sie können vor dem Absenden der Nachricht festgelegt werden.

In allen Konfigurationsstufen geht es im Wesentlichen darum, ob eine Nachricht verschlüsselt und/oder signiert werden soll und welches PGP-Format dabei verwendet werden soll: das modernere PGP/MIME oder das klassische Inline-PGP. Verzichtet man auf PGP/MIME und verwendet Inline-PGP, so dürfen die signierten/verschlüsselten Mails nicht HTML-formatiert sein. Man muss also bei Inline-PGP die HTML-Formatierung für den Nachrichten-Versand abschalten. PGP/MIME hingegen verträgt sich mit HTML.

Ist alles richtig eingestellt, so unterscheiden sich Empfang und Versand signierter und/oder verschlüsselter Mails praktisch nicht mehr vom gewohnten Empfang und Versand gewöhnlicher, unsignierter, unverschlüsselter Mails.

Zweckmäßig ist es, alle ausgehenden Mails zu signieren. Damit zeigt man den Kommunikationspartnern an, dass man einen PGP-Schlüssel besitzt und verschlüsselte Nachrichten empfangen kann. Verschlüsseln kann man nur Mails an die Empfänger, deren öffentlichen Schlüssel man in den eigenen Schlüsselbund (meist von einem Key Server) importiert hat.

Möchte man PGP auf einem weiteren PC einsetzen, so sind keine neuen Schlüssel zu erzeugen, sondern die Schlüssel aus dem Schlüsselbund des ersten PC zu exportieren und auf dem zweiten zu reimportieren. Die weitere Konfiguration ist analog zu der auf dem ersten Rechner vorzunehmen.

## 7.6 Links zu diesem Kapitel

- *Gpg4win*, Download unter <http://www.gpg4win.de/> oder unter <http://www.heise.de/download/gpg4win-e2d914fd06f82399ae36052e8362b95c-1398246471-2619466.html>
- *GPG Suite* 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](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](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 Suite:  
<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](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](http://gpg4win.de/handbuecher/einsteiger_7.html)
- Schlüsselgenerierung unter Mac OS X mit GPG Suite:  
<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](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](http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP_-_Einstieg)

## 8 PGP – Was noch wichtig oder interessant ist

Wir haben nun den PC für die PGP-Nutzung konfiguriert. Dieses Kapitel behandelt einige Themen, die für den PGP-Nutzer außerdem wichtig oder von Interesse sind.

### 8.1 Fallstricke beim Einsatz von GnuPG

Die Fallstricke beim Einsatz von GnuPG sind in einem kurzen, jedoch sehr lesenswerten Online-Artikel beschrieben. Dieser ist auf der Website des Heise-Verlags für jedermann unter folgender URL erreichbar:

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

In diesem Artikel wird übrigens die Verwendung von PGP/MIME empfohlen (siehe Kap. 7.2.1). Solange die Unterstützung dieses Formats für mein Android-Smartphone und -Tablet nicht verfügbar ist, werde ich dieser Empfehlung nicht folgen und das klassische Inline-PGP verwenden.

### 8.2 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 heißt 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 eigenen 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 (durch eine starke Passphrase) geschützt 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. (Verschlüsselte Mails lassen sich auch nicht nach bestimmten Begriffen durchsuchen.)
- Er kann keine Signaturprüfung bei eingegangen 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. 9 und Kap. 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. 8.3).

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 – also auf den eigenen Rechnern. Damit könnte man sich die Installation eines Mail-Client wie *Thunderbird* sparen.

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.

Yahoo hat übrigens angekündigt, diese Google-Erweiterung (wenn sie verfügbar ist) zu nutzen und in seine Webmail-Schnittstelle einzubinden. Es ist nicht auszuschließen, dass noch weitere Mail-Anbieter auf diesen Zug aufspringen und die PGP-Verbreitung damit eine größere Dynamik bekommt. Bis jetzt ist dies jedoch Zukunftsmusik.

An dieser Stelle will ich nochmals die generelle Webmail-Warnung aussprechen.

Die Webmail-Nutzung am fremden Rechner ist besonders riskant. 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. 4.5).

Doch auch am eigenen Rechner ist es nicht ungefährlich, mit dem Browser auf den Mail-Account zuzugreifen. Der Browser ist heute eine der bevorzugten Angriffsflächen auf den Rechner des Benutzers. Man muss manchmal nur eine infizierte Website besuchen, um sich ein Schadprogramm auf den eigenen Rechner zu holen. Verwendet man zum Mail-Zugriff nicht den Browser, sondern den Mail-Client, so ist die Gefahr, dass der Mail-Account gekapert wird, deutlich geringer.

Z.B. ist JavaScript im Browser normalerweise aktiviert, im Email-Client ist es in der Regel deaktiviert (siehe Kap. 4.4.1). Auch das Nachladen externer Inhalte lässt sich im Email-Client sehr einfach, im Browser jedoch kaum verhindern (siehe Kap. 4.4.4).

### 8.3 Die Webmail-Alternative – *Thunderbird To Go auf dem USB-Stick*

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 dann der Stick anschließen und betreiben.

Ich will diese Alternative (die ich mehr der Vollständigkeit wegen aufgenommen habe) hier nicht vertiefen. Sie ist eher für technisch Versierte als für den normalen Internetnutzer geeignet. Deshalb verweise ich 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](http://www.thunderbird-mail.de/wiki/Portable_Thunderbird)

Der zweckmäßige Weg, um auch unterwegs PGP-verschlüsselte Mails senden und empfangen zu können ist sicherlich, PGP auf dem Smartphone einzurichten.

## 8.4 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. 2.8) 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.

Auf dem Android-Smartphone oder -Tablet lässt Mail-Verschlüsselung mit PGP durchaus realisieren. Dies ist in den folgenden Kapiteln 9 und 10 beschrieben.

## 8.5 Das verschlüsselte Postfach von *mailbox.org*

Das hier beschriebene Feature ist eine Besonderheit meines Providers *mailbox.org*. Ob dies auch von anderen Providern angeboten wird, ist mir nicht bekannt. Ich habe bislang von keinem anderen Anbieter gehört, dass er dieses Feature anbietet.

*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 mit dem normalen öffentlichen Schlüssel, kann man nicht mehr feststellen, wer die Mail verschlüsselt hat, der Absender der Mail oder der Provider *mailbox.org*. Möglich wird die Unterscheidung, wenn man ein zweites Schlüsselpaar erzeugt (siehe Kap. 7.1.2). 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 zweiten 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 alle eingehenden Mails verschlüsselt. Sie sind damit über Webmail nicht mehr lesbar, da der Browser sie nicht entschlüsseln kann.

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

## 8.6 Zusammenfassung

In diesem Kapitel ging es um all die Themen rund um PGP, die nicht direkt die Nutzung von PGP mit *Thunderbird* und *Enigmail* betreffen.

Ich habe erläutert, warum Mail-Verschlüsselung mit dem Webmail-Zugriff im Browser nicht verträglich ist. Den Schlüssel beim Mail-Provider zu hinterlegen verbietet sich und der Browser hat auch keinen Zugriff auf die Schlüssel im Schlüsselbund.

Webmail kann wichtig und praktisch sein, wenn man nicht am heimischen Rechner sitzt. Will oder muss man der Verschlüsselung wegen auf Webmail verzichten, so bietet sich zwei Alternativen für die Mail unterwegs.

Mit *Thunderbird portable* kann man die *Thunderbird*-Installation, die *Thunderbird*-Konfiguration, den Schlüsselbund und auch die lokal gecachte Mail auf dem USB-Stick mit sich führen und an jedem Windows-Rechner betreiben. Sehr praktisch und laien-tauglich ist diese Lösung allerdings nicht. Die bessere Alternative ist, ein mobiles Gerät (Smartphone oder Tablet) für den Mail-Zugriff mit PGP einzurichten.

Mit dem iPhone oder iPad geht das nach meinem Kenntnisstand noch nicht so komfortabel. Allerdings kann ich dazu keine sehr kompetente Antwort liefern, da ich selbst keine iOS-Geräte nutze. Mit Android-Geräten klappt das recht gut. Dies wird in den folgenden Kapiteln behandelt.

Das letzte Thema dieses Kapitels war das verschlüsselte Postfach meines Providers *mailbox.org*. Damit ist es möglich, alle bei diesem Provider eingehenden Mails direkt nach dem Eintreffen verschlüsseln zu lassen.

## 8.7 Links zu diesem Kapitel

- 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](http://www.thunderbird-mail.de/wiki/Portable_Thunderbird)
- Das verschlüsselte Postfach von *mailbox.org*:  
<https://mailbox.org/ihr-e-mail-postfach/#Datenschutz>

## 9 PGP auf dem Android-Gerät – Schnelleinstieg für Ungeduldige

Dieses Kapitel liefert die Schritt-für-Schritt-Anleitungen für PGP auf einem Android-Gerät (Smartphone oder Tablet). Es verzichtet auf Screenshots, auf ausführliche Erläuterungen und auf die Darstellung verschiedener Optionen, sondern zeigt nur den von mir favorisierten Weg. Hier werden die Schritte aufgeführt, die durchzuführen sind, um PGP mit *APG* und *K-@ Mail* auf dem Android-Gerät einzurichten und zu nutzen. Die einzelnen Schritte haben Verweise ins Kapitel 10, sodass die ausführlichen Erläuterungen bei Bedarf schnell auffindbar sind.

Dieses Kapitel geht davon aus, dass der Zugriff auf das Mail-Konto auf dem PC mit IMAP bereits konfiguriert ist. Dabei bleiben die Mails zentral auf dem Mail-Server des Providers gespeichert. Wird auf dem PC mit POP auf das Mail-Konto zugegriffen, dann werden die Mails auf den betreffenden PC heruntergeladen und in der Regel auf dem Server gelöscht. In diesem Fall macht es keinen Sinn, auf einem weiteren Rechner oder Smartphone oder Tablet auf dasselbe Konto zuzugreifen.

### 9.1 PGP-Schlüsselverwaltung mit APG auf dem Android-Gerät

Analog zum PC benötigt man auch unter Android einen Schlüsselbund und eine App mit der man diesen Schlüsselbund verwaltet.

Die auf dem PC exportierten Schlüssel müssen in den Schlüsselbund auf dem Android-Smartphone oder -Tablet importiert werden. Dazu müssen die Schlüssel mit Hilfe eines Übertragungsmediums vom PC auf das Android-Gerät gebracht werden.

Welches Übertragungsmedium ist ab besten geeignet? USB-Sticks kann man nicht ohne Weiteres an alle Android-Geräte anschließen. Speicherkarten funktionieren auch nicht an allen Android-Geräten, viele Geräte haben keinen Speicherkarten-Slot.

Ein weiterer Weg zur Übertragung der Schlüsseldateien ist die Kopplung des Android-Gerätes mit dem PC über ein USB-Kabel. Dieser Weg funktioniert mit jedem Android-Gerät und wird hier beschrieben (siehe auch Kap. 10.2.2).

- Schlüsselverwaltungs-App *APG* aus dem Google Play Store auf dem Android-Gerät **installieren** (siehe Kap 10.1.2)
- Die **Schlüssel-Server konfigurieren** (siehe Kap 10.2.1): In *APG* unter *Einstellungen → Allgemein → Schlüsselserver* die von der Schlüsselbund-Verwaltung zu verwendenden Schlüssel-Server konfigurieren:
  - a.keyserver.pki.scientia.net
  - p80.pool.sks-keyservers.net
  - pgp.mit.edu
  - subkeys.pgp.net
- **Schlüsseldateien auf das Android-Gerät übertragen** (siehe Kap 10.2.2)
  - Den USB-Anschluss als Medien-Gerät konfigurieren: Unter *Systemeinstellungen → Speicher → USB-Verbindung → Verbinden als* ist „*Mediengerät (MTP)*“ auszuwählen

- Das Android-Gerät an einer weiteren USB-Buchse mit dem PC koppeln. Unter Windows wird der interne Speicher des Android-Geräts sofort als virtuelles Laufwerk im Explorer sichtbar. Auf dem Mac ist eine zusätzliche Software erforderlich, um auf den Speicher des Android-Gerätes zuzugreifen (s.u.).
  - Die Schlüssel aus dem *Enigmaile*-Schlüsselbund auf dem PC in zwei Dateien (eine mit dem eigenen, privaten Schlüsselpaar und mit den öffentlichen Schlüsseln der Kommunikationspartner) exportieren (siehe Kap. 7.1.6)
  - Windows: Mit dem Windows-Explorer die beiden Schlüsseldateien in den *Download*-Ordner des Android-Geräts kopieren
  - Mac OS X: Dateiübertragungssoftware *Android-Filetransfer* von <http://www.android.com/filetransfer> herunterladen und installieren und starten
  - Mac OS X: Mit *Android-Filetransfer* die beiden Schlüsseldateien in den *Download*-Ordner des Android-Geräts kopieren. *Android-Filetransfer* wieder beenden
  - Die beiden Schlüsseldateien auf dem PC wieder löschen
  - Das Android-Gerät vom PC abkoppeln
- **Schlüssel in den Schlüsselbund importieren** (siehe Kap 10.2.3)
    - Auf dem Android-Gerät die App *APG* starten
    - Das eigene Schlüsselpaar in den Schlüsselbund importieren: Unter *Importieren → Datei* die Datei mit dem eigenen Schlüsselpaar im Download-Ordner auswählen. Die Schlüssel aus der Datei werden angezeigt. Die zu importierenden Schlüssel markieren. Beim Tippen auf den Button „*Ausgewählte Schlüssel importieren*“ werden diese in den Schlüsselbund importiert.
    - Die öffentlichen Schlüssel der Kommunikationspartner in den Schlüsselbund importieren: Unter *Importieren → Datei* die Datei mit den öffentlichen Schlüsseln der Kommunikationspartner im Download-Ordner auswählen. Die Schlüssel aus der Datei werden angezeigt. Die zu importierenden Schlüssel markieren. Beim Tippen auf den Button „*Ausgewählte Schlüssel importieren*“ werden diese in den Schlüsselbund importiert.
  - Nach dem erfolgreichen Import die **Schlüssel-Dateien im *Download*-Ordner löschen**. Dies lässt sich mit einer Dateimanager-App direkt auf dem Android-Gerät erledigen. Alternativ kann man auch das Android-Gerät nochmals mit dem PC koppeln und die beiden Dateien vom PC aus löschen.
    - Das Android-Gerät wieder mit dem USB-Kabel an den PC anschließen
    - Mit dem Windows-Explorer oder auf dem Mac mit *Android-Filetransfer* die Dateien im *Download*-Ordner löschen
    - Das Gerät vom PC abkoppeln

## Links:

- OpenPGP-Unterstützung für Android mit *APG*:  
<https://play.google.com/store/apps/details?id=org.thialfihar.android.apg>

- Dateiübertragung zwischen Mac OS X und Android:  
<http://www.android.com/filetransfer/>

## 9.2 K-@ Mail für PGP-Nutzung konfigurieren

Wir haben nun die erforderlichen Schlüssel in den Schlüsselbund importiert. Nun benötigen wir eine Mail-App, die PGP unterstützt. Dies ist bei mir die App *K-@ Mail Pro* (als Alternativen stehen alle Apps der K-9 Familie zur Verfügung: siehe Kap. 10.1.1). Dieser App müssen wir mitteilen, dass sie den *APG* Schlüsselbund zur Verschlüsselung und Signierung der Mails verwenden soll.

Da das Verschlüsselungsformat PGP/MIME von *K-@ Mail* zurzeit noch nicht unterstützt wird, können nur Text-Mails versandt werden. Deshalb ist für den Versand das HTML-Format abzuschalten.

- **Mail-App *K-@ Mail Pro*** (oder eine andere App aus der K-9 Familie) aus dem Google Play Store **installieren** (siehe Kap. 10.1.2)
- *App K-@ Mail* starten und die Mail-Konten analog zu den Mail-Konten in Thunderbird einrichten. (Dies wird hier nicht näher beschrieben, da es keine PGP-spezifische Konfiguration ist.)
- Für jedes Konto, das mit PGP verwendet werden soll, **Kryptographie-Einstellungen vornehmen** (siehe Kap. 10.3.1):
  - OpenPGP Provider: *APG*
  - Automatisches Signieren abgehender Mails: *Signiere automatisch mit dem Signaturschlüssel, der zur E-Mail-Adresse des Kontos passt*
  - Verschlüsselung abgehender Mails: *Verschlüssele automatisch, wenn für den Empfänger ein öffentlicher Schlüssel vorhanden ist*
- **HTML-Format abschalten** (siehe Kap. 10.3.1): In den konten-spezifischen Einstellungen ist unter *Nachrichten senden → Formatierung* die Option „*Einfacher Text (keine Bilder und Formatierungen)*“ auszuwählen.

### Links:

- 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>

## 9.3 PGP auf dem Android-Gerät nutzen

Wir haben die Schlüssel importiert. Wir haben die geeigneten Einstellungen in *APG* und *K-@ Mail* vorgenommen. Nach diesen Vorbereitungen ist der Versand und Empfang signierter und verschlüsselter Mails nahezu trivial und weicht kaum von der Mail-Nutzung ohne PGP ab.

### 9.3.1 Versand

Das Senden einer Mail funktioniert so wie auch ohne PGP. 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 hier meist schon die richtige Auswahl getroffen.

Beim verschlüsselten Mailversand 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.

Der Versand von Mails im modernen Format PGP/MIME wird aktuell noch nicht unterstützt.

### 9.3.2 Empfang

Wird eine verschlüsselte Mail mit *K-@ Mail* auf dem Android-Gerät empfangen, ist sie zunächst noch nicht lesbar. Die App zeigt über der Mail eine Schaltfläche mit der Aufschrift „*Entschlüsseln*“. Beim Tippen auf diese Schaltfläche verlangt die App evtl. 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 den Text darstellen.

Wird eine Mail im Format PGP/MIME empfangen, so kann die App (da sie dieses Format noch nicht unterstützt) sie nicht entschlüsseln.

### 9.3.3 Schlüsselbund-Pflege

Von Zeit zu Zeit sollte man die öffentlichen Schlüssel im Schlüsselbund aktualisieren. Die Pflege des Schlüsselbundes ist allerdings nicht so komfortabel möglich wie in *Thunderbird/Enigmail*. So ist es zweckmäßig, die öffentlichen Schlüssel im *Enigmail*-Schlüsselbund nach der Synchronisation mit dem Schlüssel-Server (siehe Kap. 6.4.3) zu exportieren und von dort (wie in Kap. 9.1 beschrieben) in den Schlüsselbund auf dem Android-Gerät zu importieren.

## 9.4 Zusammenfassung

Die Zusammenfassung ist am Ende des ausführlichen Kapitels zur *Thunderbird/Enigmail*-Konfiguration zu finden (siehe Kap. 10.5).

## 9.5 Links zu diesem Kapitel

Die Link-Sammlung ist am Ende des ausführlichen Kapitels zur *Thunderbird/Enigmail*-Konfiguration zu finden (siehe Kap. 10.6).

## 10 PGP auf dem Android-Gerät – Ausführlicher Einstieg für Wissbegierige

Dieses Kapitel liefert den ausführlichen Einstieg in PGP auf dem Android-Gerät. Dabei wird ausführlich erläutert, wie PGP mit *APG* und *K-@ Mail* auf dem Android-Gerät eingerichtet und genutzt werden kann. Es versucht nicht nur die Frage „Wie muss ich vorgehen?“ zu beantworten, sondern auch „Warum ist das so?“ und „Welche anderen Optionen gibt es?“. Die gesamte Beschreibung ist einerseits detailreicher und möchte andererseits das Verständnis für das, was man tut, unterstützen.

Dieses Kapitel geht davon aus, dass der Zugriff auf das Mail-Konto auf dem PC mit IMAP bereits konfiguriert ist. Dabei bleiben die Mails zentral auf dem Mail-Server des Providers gespeichert. Wird auf dem PC mit POP auf das Mail-Konto zugegriffen, dann werden die Mails auf den betreffenden PC heruntergeladen und in der Regel auf dem Server gelöscht. In diesem Fall macht es keinen Sinn, auf einem weiteren Rechner oder Smartphone oder Tablet auf dasselbe Konto zuzugreifen.

### 10.1 Apps installieren

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.

#### 10.1.1 Die passenden Android-Apps

Für die Schlüsselbund-Verwaltung 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 (Oktober 2014) noch keine stabile Unterstützung des moderneren Verschlüsselungsformates PGP/MIME gegeben. Die Entwickler jedoch daran arbeiten, sodass die Situation in einigen Monaten 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 nicht oder nur geringfügig unterscheiden.

### 10.1.2 Installation der Apps

Die Apps installiert man wie üblich aus dem Google Play Store. Es ist zweckmäßig, vor der Mail-App (bei mir *K-@ Mail Pro*) die Schlüsselbund-Verwaltungs-App (bei mir *APG*) zu installieren. 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. 3.5.1) vorzunehmen.

Sendet man jetzt auf dem PC eine verschlüsselte Mail an sich selbst und versucht sie auf dem Smartphone bzw. Tablet im Posteingang der Mail-App zu öffnen, so kann diese jetzt noch nicht entschlüsselt werden (siehe Abb. 29).

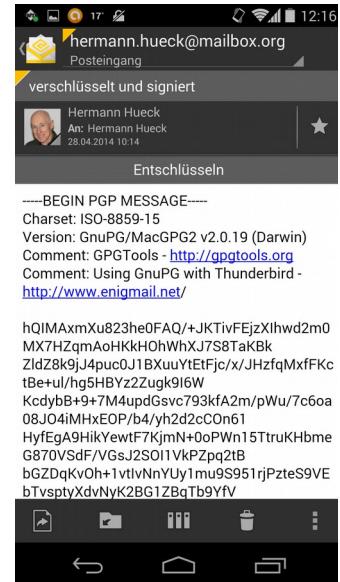


Abbildung 31: Android:  
*K-@ Mail*: Nur  
Zeichensalat – Die  
verschlüsselte Mail ist  
nicht lesbar.

## 10.2 Verwaltung des PGP-Schlüsselbundes mit APG

### 10.2.1 Konfiguration der Schlüsselserver

In *APG* unter *Einstellungen* → *Allgemein* → *Schlüsselserver* 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

### 10.2.2 Übertragung der Schlüssel auf das Android-Gerät

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. 7.4). Auch hier nochmals die Warnung, die Schlüssel niemals über einen unverschlüsselten Netzwerkkanal zu übertragen!

Für Android-Geräte bietet sich zusätzlich zu den in Kapitel 7.4 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 wird der interne Speicher des Android-Geräts sofort als virtuelles Windows-Laufwerk sichtbar. Nun kann man mit dem Windows-Explorer die exportierten Schlüssel direkt in einen Ordner des Android-Geräts (z.B. den Download-Ordner) übertragen. Die Übertragung ist unter Windows ein einfacher Kopiervorgang.

Auf dem Mac muss zunächst eine Dateübertragungssoftware von <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.

### 10.2.3 Schlüssel-Import in den Schlüsselbund

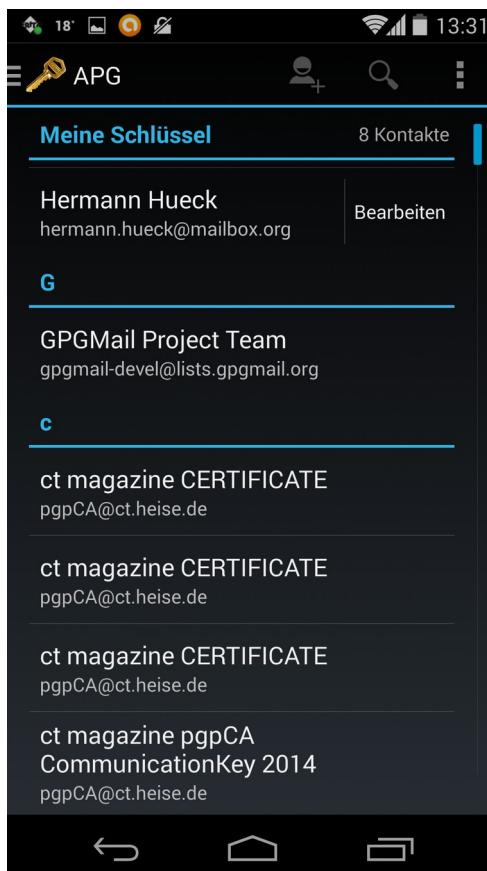


Abbildung 33: Android: Die importierten Schlüssel im APG-Schlüsselbund

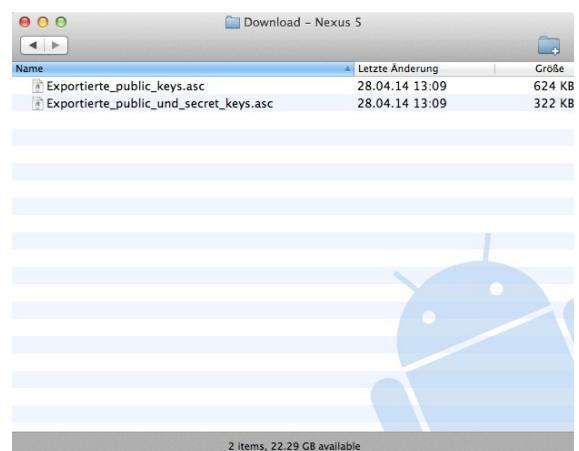


Abbildung 32: Android-Filetransfer auf dem Mac, Smartphone über USB angeschlossen: Die Schlüssel werden ins Download-Verzeichnis kopiert.

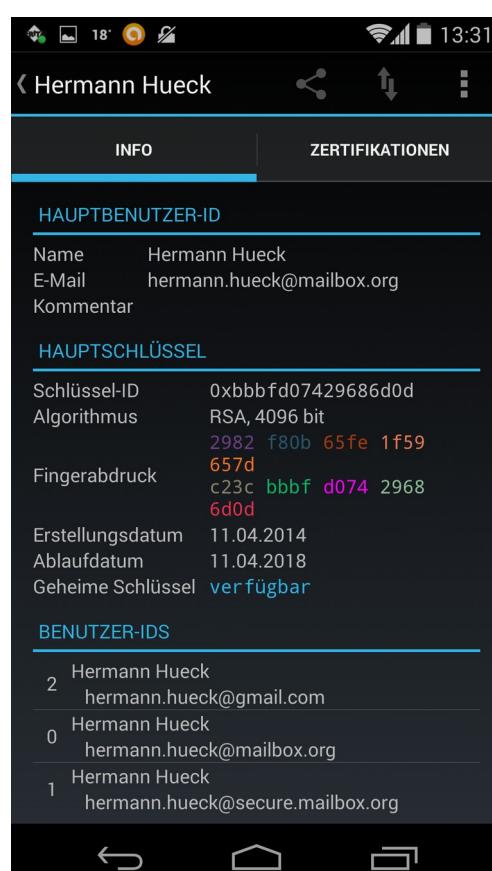


Abbildung 34: Android: APG-Schlüsseldetails: Ein Schlüssel mit drei Benutzer-IDs

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.

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.

## 10.3 Konfiguration der Mail-App für die Nutzung von PGP

### 10.3.1 K-@ Mail Einstellungen

Die verschlüsselte Mail, die vor dem Schlüssel-Import noch nicht lesbar war, lässt sich nun schon entschlüsseln und lesen (siehe Abb. 36).

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 (siehe Abb. 35).

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

### 10.3.2 Text-Mails statt HTML-Mails

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. 7.2.3) 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)*“ auszuwählen.

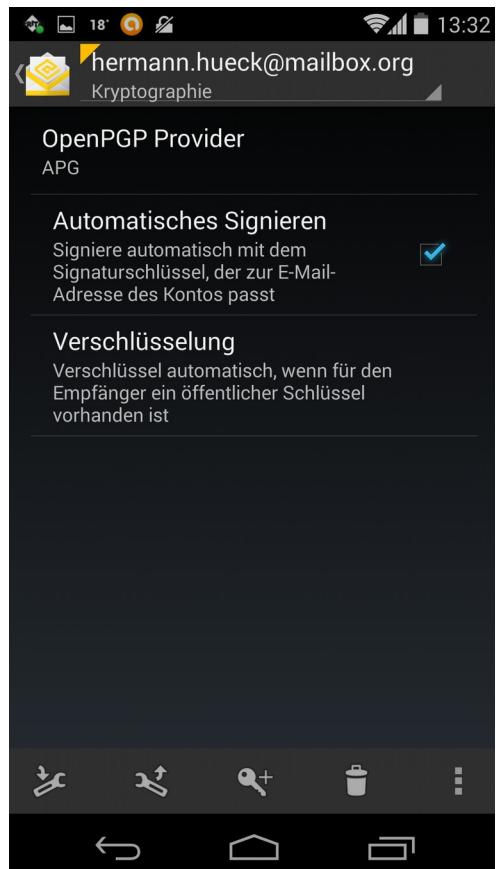


Abbildung 35: Android: K-@ Mail: Kryptographie-Optionen

Diese Einstellung ist für jedes Konto, das zum Versenden verschlüsselter Mails verwendet werden soll, vorzunehmen.

## 10.4 Nutzung von PGP

Nun sind alle Einstellungen vorgenommen. Verschlüsselte und signierte Mails können jetzt gesendet und empfangen werden.

### 10.4.1 Mailversand

Das Senden einer Mail funktioniert wie auch ohne die Verwendung von PGP 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.

Da das Verschlüsselungsformat PGP/MIME aktuell von *APG* noch nicht unterstützt wird, kann man 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 PC 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 PGP-Format bislang fehlt.

### 10.4.2 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 31 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 Abb. 36).



verschlüsselt und signiert

Hermann Hueck

- Senior Software Developer
- Scala, Java/JEE, Android, JavaScript



Abbildung 36: Android: K-@ Mail:  
Die Mail kann jetzt gelesen werden.

### 10.4.3 Schlüsselbund-Pflege

Von Zeit zu Zeit sollte man die öffentlichen Schlüssel im Schlüsselbund aktualisieren. Die Pflege des Schlüsselbundes ist allerdings nicht so komfortabel möglich wie in *Thunderbird/Enigmail*. So ist es zweckmäßig, die öffentlichen Schlüssel im *Enigmail*-Schlüsselbund nach der Synchronisation mit dem Schlüssel-Server (siehe Kap. 7.1.8) auf den USB-Stick zu exportieren und von dort auf das Android-Gerät übertragen (siehe Kap. 10.2.2) und in den Schlüsselbund zu importieren (siehe Kap. 10.2.3).

## 10.5 Zusammenfassung

Dieses Kapitel zeigte die Einrichtung eines Android-Geräts für die Nutzung von PGP.

Man installiert zunächst die passenden Apps: *APG* für die Verwaltung des Schlüsselbundes und eine App aus der K-9 Familie, z.B. *K-@ Mail Pro* als Email-Client mit PGP-Unterstützung. In *APG* sind – ebenso wie in der Schlüsselverwaltung auf dem PC – die zu verwendenden Schlüssel-Server einzustellen.

Die Schlüssel holt man sich am einfachsten aus der Schlüsselbund-Verwaltung am PC. Dazu exportiert man den/die eigenen privaten Schlüssel und auch die fremden öffentlichen Schlüssel auf dem PC in zwei Dateien. Danach schließt man das Android-Gerät mit dem USB-Kabel (als Medien-Gerät) an den PC an und kann nun unter Windows mit dem Windows-Explorer sofort auf das Android-Dateisystem zugreifen. Auf dem Mac muss zunächst *Android Filetransfer* installiert werden, damit man auf das Android-Dateisystem zugreifen kann. Die beiden exportierten Dateien überträgt man nun auf das Android-Gerät z.B. in den Download-Ordner.

Man trennt nun das Android-Gerät wieder vom PC und importiert die Schlüssel aus den beiden Dateien im Download-Ordner in den *APG*-Schlüsselbund. Die Schlüsseldateien sind nun zu löschen. Dazu kann man entweder einen Android-Dateimanager verwenden oder das Gerät nochmals mit dem PC koppeln, die Dateien mit dem Windows-Explorer bzw. auf dem Mac mit dem Program *Android Filetransfer* löschen und danach die USB-Kopplung zwischen PC und dem Gerät wieder trennen.

Nun sind in den Konto-Einstellungen jedes Mail-Kontos, bei dem PGP eingesetzt werden soll, die Kryptographie-Optionen zu setzen. Hier stellt man zweckmäßigerweise ein, dass *APG* als Schlüsselbund-Verwaltung verwendet, alle Mails unterschrieben und nur die Mails, für deren Empfänger ein öffentlicher Schlüssel im Schlüsselbund vorliegt, verschlüsselt werden sollen.

Da das moderne Format PGP/MIME leider noch nicht von den Mail-Apps der K-9 Familie unterstützt wird, kann man unter Android die Mails nur mit Inline-PGP verschlüsseln. Das klassische Inline-PGP-Format verträgt sich allerdings nicht mit HTML-Mails. Man muss deshalb – ebenfalls in den Kontoeinstellungen – das Mail-Format auf einfachen Text umstellen.

Möchte man nun eine verschlüsselte Mail öffnen, so sieht man zunächst nur den verschlüsselten Zeichensalat. Jedoch blendet die Mail-App einen Button mit der Aufschrift entschlüsseln ein. Nach dem Tippen auf den Button und der Eingabe der richtigen Passphrase ist der Zugriff auf den privaten Schlüssel möglich. Mit diesem wird die Mail dechiffriert und so lesbar gemacht.

Beim Mailversand kann man nun für jede Mail erneut festlegen, ob sie verschlüsselt und/oder signiert werden soll. Die Einstellung ist mit den Voreinstellungen, die man in den Kryptographie-Optionen vorgenommen hat vorbelegt. Meistens muss nichts geändert werden. Bevor eine verschlüsselte Mail versendet wird, muss man nochmals bestätigen, mit welchen Schlüsseln die Mail verschlüsselt werden soll. Dabei ist außer dem Schlüssel des Empfängers auch der eigene

Schlüssel auszuwählen. Andernfalls könnte man die Kopie der Mail im eigenen Gesendet-Ordner nicht mehr dechiffrieren und lesen.

Von Zeit zu Zeit sollten man die öffentlichen Schlüssel des Schlüsselbundes von den Key Servern aktualisieren. So erhält man für die Schlüssel, die bereits im Schlüsselbund sind, auch die aktuellen Beglaubigungen.

## 10.6 Links zu diesem Kapitel

- OpenPGP-Unterstützung für Android mit *APG*:  
<https://play.google.com/store/apps/details?id=org.thialfihar.android.apg>
- OpenPGP-Unterstützung für Android mit *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/>

## 11 PGP/MIME – Und es funktioniert doch.

In den vorangehenden Kapiteln hatte ich noch den Verzicht auf das moderne PGP/MIME-Format und stattdessen die Verwendung von klassischem Inline-PGP empfohlen. Der Grund dafür war, dass dieses Format auf dem Android-Gerät bislang nicht unterstützt wurde.

Seit Sommer 2014 liegt nun eine Unterstützung dafür vor. Adam Wassermann hat mit *PGP KeyRing* und mit *Squeaky Mail* zwei Apps programmiert, die auch PGP/MIME unterstützen. *PGP KeyRing* ist eine App zur Verwaltung des Schlüsselbundes, *Squeaky Mail* ist die dazu passende Mail-App. Dieses Tandem kann alternativ zu *APG* und *K-@ Mail* eingesetzt werden. Damit muss man auch nicht mehr auf HTML-formatierte Mails verzichten.

- Android-App *PGP KeyRing* zur Schlüsselbund-Verwaltung:  
<https://play.google.com/store/apps/details?id=com.imaeses.keyring.trial>
- Android Mail-App *Squeaky Mail* als Email-Client:  
<https://play.google.com/store/apps/details?id=com.imaeses.squeaky>

Es ist ebenfalls möglich, diese beiden Apps neben *APG* und *K-@ Mail* auf demselben Gerät zu betreiben. Dies praktiziere ich auf meinen Android-Geräten. Ich verwende *Squeaky Mail* nur, wenn ich mit PGP/MIME verschlüsselte Mails auf dem Android-Gerät lesen oder verfassen will. In allen anderen Fällen nutze ich *K-@ Mail*.

Ich werde vorläufig die vorangehenden Kapitel noch nicht umschreiben, sondern mich in diesem Kapitel nur auf die Änderungen zur Verwendung von PGP/MIME konzentrieren.

### 11.1 PGP/MIME auf dem Android-Gerät

Statt *APG* ist die App *PGP KeyRing* zu installieren und völlig analog zu konfigurieren. Statt *K-@ Mail Pro* ist *Squeaky Mail* zu installieren und – abgesehen von zwei Ausnahmen – analog zu konfigurieren.

1. In den Kryptographie-Einstellungen steht bei *Squeaky Mail* die Option „*Use PGP/MIME*“ zur Verfügung. Diese Option ist nun auszuwählen (vgl. Kap. 10.3.1 und Abb. 37).
2. Anders als in Kap. 10.3.2 lassen sich statt Text-Mails jetzt auch HTML-Mails verwenden.

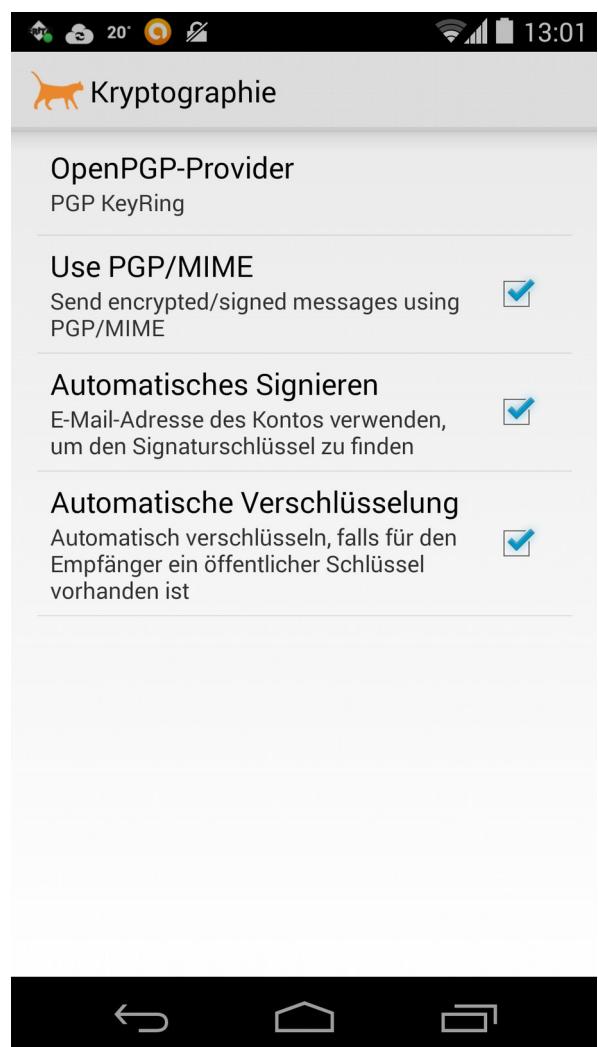


Abbildung 37: *Squeaky Mail*: Kryptographie-Optionen: PGP/MIME ist wählbar. (vgl. Abb. 37).

## 11.2 PGP/MIME auf dem PC

Da wir nun auf dem Android-Gerät mit PGP/MIME verschlüsselten Mails umgehen können, lässt sich auch *Thunderbird/Enigmail* auf dem PC entsprechend umstellen.

1. Anders als in Kapitel 7.2.1 empfohlen, wählen wir in den konten-spezifischen OpenPGP-Einstellungen die Option „*PGP/MIME standardmäßig verwenden*“ aus.
2. Wir müssen nun nicht mehr auf HTML-formatierte Mails verzichten. Anders als in Kapitel 7.2.3 beschrieben kann man im Konfigurationsordner *Verfassen und Adressieren* die Option „*Nachrichten im HTML-Format verfassen*“ wieder einschalten. Damit stehen beim Verfassen von Mails Text-Formatierungen wie Überschriften, Fettschrift, Kursivschrift, Aufzählungslisten und unterschiedliche Schriftarten wieder zur Verfügung.
3. Beim Verfassen einer Mail kann PGP/MIME für jede einzelne Mail im Menü *Enigmail* an- oder abgeschaltet werden.

## 11.3 Das verschlüsselte Postfach von *mailbox.org*

Wie in Kapitel 8.5 beschrieben, werden bei Verwendung des verschlüsselten Postfachs alle eingehenden unverschlüsselten Mails vom Mail-Provider verschlüsselt. Bei *mailbox.org* wird dabei das Format PGP/MIME verwendet. Da man die so verschlüsselten Mails nun auf dem PC und auf dem Android-Gerät entschlüsseln und lesen kann, kann man sich nun überlegen, dieses Feature in der Konfiguration des Mail-Kontos beim Provider wieder einzuschalten.

Dabei sollte man sich (ganz unabhängig von PGP/MIME) jedoch auch der Nachteile dieses Features bewusst sein. Die verschlüsselten Mails lassen nicht durchsuchen und sie können nicht über die Webmail-Schnittstelle entschlüsselt und gelesen werden (siehe Kap. 8.5).

## 11.4 Zusammenfassung

Nun kann man seit Sommer 2014 also doch PGP/MIME unter Android nutzen. Dies geht mit den Apps *PGP KeyRing* (statt *APG*) zur Schlüsselbund-Verwaltung und mit der Email-App *Squeaky Mail* (statt *K-@ Mail Pro*). Diese App ist ebenfalls ein Spross der K-9 Familie.

In den Kryptographie-Optionen der Mail-App kann man nun auch die Verwendung von PGP/MIME anhaken. Da PGP/MIME HTML-verträglich ist, lässt sich das Verfassen der Mails nun wieder auf das HTML-Format umstellen.

Kann man mit PGP/MIME chiffrierte bzw. signierte Mails unter Android lesen, so ist es sinnvoll, auch auf dem PC die Verwendung von PGP/MIME festzulegen und – wenn man möchte – die Mails im HTML-Format statt im reinen Text-Format zu verfassen.

Sind die Mail-Clients auf allen Geräten auf PGP/MIME eingestellt, so kann man als Kunde von *mailbox.org* auch erwägen, das verschlüsselte Postfach zu aktivieren. Nach der Aktivierung dieser Option werden alle beim Provider eingehenden Mails im Format PGP/MIME verschlüsselt.

## 11.5 Links zu diesem Kapitel

- PGP-Unterstützung für Android mit *PGP KeyRing*:  
<https://play.google.com/store/apps/details?id=com.imaeses.keyring.trial>
- Android Mail-App mit PGP/MIME Unterstützung: *Squeaky Mail*:  
<https://play.google.com/store/apps/details?id=com.imaeses.squeaky>

## 12 Passwortsicherheit und Rechnersicherheit

Der Anhang behandelt einige allgemeine Sicherheitsfragen wie Passwort- 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>

### 12.1 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. Warum? Wörterbuch-Attacken probieren einfach alle Wörter eines Wörterbuches aus. Ein Mensch würde Jahre dazu benötigen, ein leistungsfähiger Computer benötigt dazu nur Minuten und kann dazu auch die Wörterbücher mehrerer Sprachen heranziehen.
- Für verschiedene Dienste sind verschiedene Passwörter zu benutzen. Nicht selten verwenden Benutzer heute 50 und mehr Passwörter für verschiedene Dienste. Wird das Passwort eines Dienstes geknackt oder gestohlen, dann sind bei unterschiedlichen Passwörtern 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 für sicherheitskritische Accounts ein Änderungsintervall von ca. einem halben Jahr empfehlen. 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 oder Passwort-Safe ist installierbares Programm, das Passwörter verwaltet – in etwa analog zur Schlüsselverwaltung bei PGP. Der Zugriff auf die Passwortliste des Passwort-Managers ist mit einem starken Passwort, dem sog. Master-Passwort, zu 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.

Besonders zweckmäßig ist ein Passwort-Safe der auf unterschiedlichen Betriebssystemen (Windows, Mac OS X, Linux, Android und iOS) funktioniert und der die Passwortliste über einen Cloud-Anbieter zwischen verschiedenen Geräten synchronisieren kann. Dies ist auch bei einem Cloud-Provider wie *Dropbox*, der keine Ende-zu-Ende-Verschlüsselung anbietet, ungefährlich unter der Voraussetzung, dass die Passwortliste durch ein starkes Master-Passwort geschützt ist. Ein Dieb,

dem es gelingt, die Liste abzugreifen, müsste das Master-Passwort knacken, um an die Passwörter im Safe heranzukommen. Hier sollte man keine Scheu haben, ein Master-Passwort mit 15 oder mehr Zeichen zu wählen und Buchstaben, Ziffern und Sonderzeichen gut zu mischen.

## 12.2 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 (dies kann durchaus aufwändig sein)
- Je nach Betriebssystem einen aktuellen VirensScanner 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.

### 12.2.1 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) selbsttätig, 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/](http://secunia.com/vulnerability_scanning/personal/) ).
- **VirensScanner:** Einen Windows-Rechner ohne VirensScanner zu betreiben oder die Viren-Signaturen nicht mehrmals täglich zu aktualisieren, darf als grobe Fahrlässigkeit eingestuft werden. Die Viren-Signaturen aktualisiert der installierte VirensScanner per Voreinstellung in der Regel automatisch. Da Viren heute oft in sehr kurzer Abständen neu erstellt oder variiert werden, müssen permanent neue Viren-Signaturen auf ihren Servern bereitstellen. Man sollte darauf achten, dass die **Viren-Signaturen im stündlichen Rhythmus aktualisiert werden**.

### 12.2.2 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) selbsttätig, ob im Mac OS X App Store Updates vorliegen und aktualisiert sich. Auf diesem Wege werden 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. *Chrome* kann sich auch selbst aktualisieren. *Firefox* und *Thunderbird* machen mit einem Meldungsfenster darauf aufmerksam, wenn sie aktualisiert werden wollen, ebenso der *Adobe Reader* und *Java*. Mit einem Mausklick auf den Button „OK“ im Meldungsfenster kann man die Aktualisierung meist gleich starten.
  - 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. Allerdings ist es auch so, dass Programme, die nur installiert aber nicht aktiv sind, auch keinen Schaden anrichten können. Man kann sich auch angewöhnen, nach dem Start eines Programms dieses zunächst nach Aktualisierungen suchen zu lassen.
- **VirensScanner:** Früher konnte man einen Mac auch ohne VirensScanner 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 VirensScanner zu installieren und die Viren-Signaturen mehrmals täglich zu aktualisieren bzw. aktualisieren zu lassen.

### 12.2.3 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 alle Programme und Programm-Bibliotheken. Wird dabei der Linux-Kernel aktualisiert, ist ein Neustart des Rechners erforderlich.
- **VirensScanner:** 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-Virensaltern ebenso wenig lukrativ. Die Hersteller von Virensaltern bieten diese für Windows und meist für Mac OS X an. Es gibt jedoch keine VirensScanner für Linux. Der Linux-Anwender muss

zwar auf den Virenschanner verzichten, hat trotzdem ein System, das durch Viren fast nicht gefährdet ist.

## 12.2.4 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 Virenschanners ist weder sinnvoll noch möglich. Mit regelmäßigen Aktualisierungen des iOS-Betriebssystems und der installierten Apps lassen sich iOS-Geräte sehr sicher betreiben.

## 12.2.5 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 Update-Versorgung der älteren Geräte. 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.

Besser verhält es sich mit den Geräten der Nexus-Serie von Google. Bei diesen Geräten kommen auch die Betriebssystem-Updates von Google. Nach dem Erscheinen eines Updates stellt Google dieses sehr schnell für die Nexus-Geräte bereit. Die übrigen Hersteller liefern die Android-Updates mindestens ein halbes Jahr später aus. Oft dauert es noch länger und für ältere Geräte wird häufig kein Update mehr bereitgestellt.

- **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, doch 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 auf das sog. Side Loading verzichtet und die 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.
- **Virenschanner:** Man kann das Restrisiko durchaus noch etwas verkleinern, indem man sich eine Virenschanner-App auf das Android-Gerät installiert. Viele Hersteller von Virenschannern für Windows bieten mittlerweile auch eine Virenschanner-App für Android im Play Store an. Die Virenschanner-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 Virenschanner-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.).

## 12.3 Zusammenfassung

Auch die Passwortsicherheit und die Rechnersicherheit kommen indirekt der Email-Sicherheit zugute.

In diesem Kapitel haben wir Regeln für gute Passwörter aufgestellt. Das Verwalten vieler Passwörter lässt sich mit einem Passwort-Manager besser bewältigen.

Um den Rechner bzw. das mobile Gerät sicher zu machen und sicher zu halten, muss das Betriebssystem immer auf dem aktuellen Stand sein. Auch die installierten Programme sind auf dem aktuellen Stand zu halten. Unter Windows kann das Programm Secunia PSI dabei eine große Hilfe sein.

Vor Allem unter Windows benötigt man auch einen Virenschanner, der im Stundentakt mit aktuellen Virensignaturen versorgt wird.

Unter Android und iOS erhält man das Betriebssystem-Update, wenn es vom Hersteller bereitgestellt wird. Die Apps aktualisieren sich aus dem Google Play Store bzw. aus dem Apple App Store. Unter Android kann auch ein Virenschanner sinnvoll sein, um das System gegen Malware zu härten.

## 12.4 Links zu diesem Kapitel

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

## 13 Verschlüsselte Mails – und was noch?

Verschlüsselte Mails sind wichtig, aber noch lange nicht alles. Wir nutzen das Internet auch noch auf anderen Kommunikations-Kanälen:

- SMS
- Telefonie
- Groupware-Dienste (Adressbuch, Kalender, Aufgaben-Liste) in der Cloud
- Speicherdiensste: in der Cloud gespeicherte Ordner und Dateien
- Surfen im Web (Auf dieses sehr umfangreiche Thema werde ich hier nicht eingehen.)

Für alle diese Dienste gibt es sichere Alternativen oder Nutzungsempfehlungen. Um sie so ausführlich zu beschreiben, wie ich dies bei der Email getan habe, müsste ich den Umfang des vorliegenden Dokuments mindestens verdoppeln.

Ich will an dieser Stelle zunächst einige Konzepte erläutern. Ich werde mich dabei immer wieder auf die Email beziehen und – wo immer es sich anbietet – auch die Analogien aufgreifen und aufzeigen. Unser Wissen über Verschlüsselung kommt uns natürlich sehr zugute.

### 13.1 Grundsätzliches

Dieses Kapitel erläutert einige Konzepte, die für das Verständnis der nachfolgenden Kapitel hilfreich sind.

#### 13.1.1 Zero Knowledge

**Zero Knowledge** ist ein Begriff, der als Gütesiegel verschlüsselnder Dienste angesehen wird. Manche Dienstleister werben sogar mit diesem Begriff.

**Zero Knowledge** bedeutet: **Der Provider weiß nichts. Er kennt meine Daten nicht.**

Dies bedeutet konkret, dass der Provider nur verschlüsselte Daten (zur Aufbewahrung oder Weiterleitung) erhält. Er darf jedoch niemals den Schlüssel zu den Daten und damit den Zugang zu den unverschlüsselten Inhalten erhalten. Der Provider bleibt also, was den Inhalt meiner Daten angeht, „unwissend“. Beispielsweise hat die Post „kein Wissen“ über den Inhalt des Briefes, den ich ihr zum Transport übergebe.

Dies hat nicht nur den Vorteil, dass der Provider meine Daten nicht kennt. Selbst wenn er gezwungen wird, die Daten an Behörden oder Geheimdienste herauszugeben, so erhalten diese ebenfalls nur die verschlüsselten Daten, die sie nicht lesen können. Auch wenn Hacker in die Systeme meines Providers einbrechen, sind meine Daten dennoch geschützt. Auch die Hacker können nur meine verschlüsselten Daten stehlen. Um sie zu entschlüsseln und zu lesen, müssten sie zusätzlich den Schlüssel aus meinem Schlüsselbund stehlen.

Verschlüsselte Mail ist ein Beispiel für **Zero Knowledge**. Eine mit einem öffentlichen PGP-Schlüssel verschlüsselte Mail kann nur vom Empfänger entschlüsselt und gelesen werden, solange dieser allein im Besitz des privaten Schlüssels ist. Der Mail-Inhalt bleibt vor dem Provider, vor Hackern und vor Geheimdiensten und Behörden verborgen. **Zero Knowledge** ist ein **Synonym für Ende-zu-Ende-Verschlüsselung**.

Mit Transport-Verschlüsselung sind die Daten nur verschlüsselt, solange sie unterwegs sind. Damit

alleine kann *Zero Knowledge* niemals implementiert werden. Um *Zero Knowledge* als Dienstmerkmal bereitzustellen, müssen immer die Daten verschlüsselt werden.

### 13.1.2 Open Source

**Open Source** kann ein wichtiges Merkmal für einen Dienst sein. Ist die Software, die den Dienst implementiert, *Open Source* Software, so ist der Quellcode offengelegt und für jedermann zugänglich. Sie kann also von jedem (der das erforderliche Know-how dazu hat) überprüft werden.

Ist die Software eines Dienst-Providers *Closed Source*, so bleiben die Funktionen und Qualitätsmerkmale, mit denen der Provider seinen Dienst bewirbt, eine pure Behauptung, der ich Glauben schenken kann oder auch nicht. Ich muss dem Provider vertrauen. *Open Source* Software hingegen ist grundsätzlich überprüfbar.

Allerdings muss man hinzufügen, dass *Open Source* keine Garantie dafür ist, dass die Software tatsächlich auch überprüft wird. Dies hat sich jüngst bei der Bibliothek *OpenSSL* gezeigt. Über Jahre war in dieser weit verbreiteten Bibliothek ein schwerer Fehler (*Heartbleed*, siehe Kap. 14.5) verborgen, der erst im Frühjahr 2014 entdeckt wurde.

### 13.1.3 Was sind Cloud-Dienste?

Der Kerngedanke der Cloud (so wie sie der Privatnutzer kennt) ist die **Synchronisation von Daten zwischen verschiedenen vernetzten Geräten desselben Benutzers**.

(Die sog. Cloud hat viele Aspekte. Dies ist eine sehr vereinfachte Sicht der Cloud. Da dieser Aspekt für den Privatnutzer im Vordergrund steht und im Kontext dieses Dokuments ausreichend ist, bleibe ich bei dieser Vereinfachung.)

Ein paar Beispiele:

- Ich kopiere auf meinem Smartphone ein paar Fotos in einen neuen *Dropbox*-Ordner. „Wie von Zauberhand“ erscheint der neue *Dropbox*-Ordner mit dem Fotos auf meinem PC. Je nach Netzwerkbandbreite und Anzahl der Fotos dauert dieser Vorgang zwischen wenigen Sekunden und einigen Minuten. Bei einer sehr langsamen Netzverbindung und sehr vielen Fotos können auch Stunden daraus werden. (Speicher-Cloud oder Cloud Storage oder Online-Festplatte)
- Ich erzeuge einen neuen Termin in meinem Kalender auf dem PC. Im Handumdrehen erscheint der Termin auch im Kalender auf meinem Tablet oder Smartphone. (Kalender-Cloud)
- Ich lösche einen Kontakt aus der Kontaktliste meines Tablet. Hat man die Kontaktliste auch anderen Geräten geöffnet, so kann man häufig zusehen, wie der Kontakt auch auf den anderen Geräten verschwindet. Manchmal geht's auch nicht so flott. Dann muss man ein wenig warten oder die Aktualisierung anstoßen, damit der Kontakt überall verschwindet. (kontakte-Cloud)
- Ein mit IMAP verwaltetes Email-Konto kann man als Email-Cloud betrachten. In *Thunderbird* auf dem PC verschiebe ich eine Mail aus dem Posteingang in einen anderen Ordner. In der Mail-App auf dem Tablet oder Smartphone kann man zusehen, wie die Mail aus dem Posteingang verschwindet. Und wenn man nachsieht, findet sich auch auf diesen Geräten die Mail im neuen Ordner.
- Ich lese ein E-Book auf meinem E-Book-Reader und unterbreche meine Lektüre auf Seite

83. Nach zwei Stunden möchte ich auf dem Tablet weiterlesen und öffne die Reader-App. Die Reader-App „weiß“, welches E-Book ich zuvor auf dem anderen Gerät gelesen haben und bietet mir sogar an, auf Seite 83 fortzufahren, auf der ich meine Lektüre zuvor unterbrochen habe. Die Cloud macht's möglich.

- Ich füge meinem Chrome-Browser auf dem PC ein neues Lesezeichen hinzu. Es dauert nicht lange, dann finde ich dieses Lesezeichen auch im Chrome-Browser auf dem Smartphone und kann die betreffende URL dann auch dort öffnen. Dies funktioniert auch mit Firefox und mit anderen Browsern. (Man könnte es als „Bookmark-Cloud“ bezeichnen.)

Was ist die Voraussetzung für diese automatische Synchronisation? Die Geräte benötigen eine Internet-Verbindung und sie müssen alle bei demselben Anbieter mit demselben Benutzer-Account angemeldet sein.

Meine Daten (Dateien, Kalender, Kontakte, Mails, E-Books, Bookmarks) werden zentral beim Cloud-Anbieter gespeichert. Die Geräte enthalten ein Kopie des zentralen Datenbestandes und synchronisieren sich permanent und automatisch.

Cloud-Dienste sind für den Benutzer sehr bequem. Ich ändere meine Daten auf einem Gerät. Die anderen Geräte synchronisieren sich automatisch. Das umständliche und zeitaufwändige Kopieren der Daten zwischen den Geräten gehört der Vergangenheit an.

Doch unter Sicherheitsgesichtspunkten habe ich mir ein Problem eingehandelt. Die Daten lagern – allermeist unverschlüsselt – beim Provider. Das zentrale Datenlager ist nicht mehr in meiner eigenen Verfügungsgewalt.

Der Provider kann auf meine unverschlüsselten Daten zugreifen. Hacker, die es schaffen, beim Provider einzubrechen, haben ebenfalls Zugriff auf meine Daten. Polizeiliche Ermittlungsbehörden können den Zugriff auf meine Daten vom Provider durch Gerichtsbeschluss erzwingen. Die Geheimdienste machen es zuweilen so wie die Behörden und erzwingen den Zugriff auf meine Daten oder sie verfahren wie die Hacker, brechen beim Provider ein und stehlen sich die Daten.

Mit dem Cloud-Sicherheitsproblem kann man nun auf verschiedene Arten umgehen:

- Man kann die Cloud-Dienste weiter nutzen und die **Sicherheitsproblematik ignorieren**. Man speichert nur vermeintlich „unkritische“ Daten in der Cloud. Da stellt sich natürlich sofort die Frage, ob z.B. Kalender und Kontakte als „unkritische“ Daten zu betrachten sind.
- Man kann **auf die Cloud-Dienste verzichten**. Solange man nur ein Gerät hat, z.B. einen PC, ist dieser Weg ohne große Komforteinbußen gangbar. Schon wenn man den alten PC durch einen neuen ersetzen will, muss man seine Daten von dem alten auf den neuen PC übertragen. Dies kann ohne Cloud schon recht aufwändig werden.
- Man kann in der eigenen Wohnung in der Kammer oder im Keller (mit *OwnCloud*) seine **eigene Cloud einrichten**. Man wird dadurch zum Cloud-Provider für sich die eigenen Daten, bzw. für die Daten der Familie. Dies hat den Vorteil, dass die Daten das eigene Heimnetz nicht verlassen. Allerdings setzt diese Strategie ein gewisses technisches Know-how voraus und ist auch mit etwas Zeitaufwand verbunden. Ein eigener Cloud-Server muss ja erst einmal eingerichtet und dann im laufenden Betrieb auch gepflegt werden.
- Man sucht sich einen **zuverlässigen Provider**, bei dem man die eigenen Daten gut aufgehoben glaubt. Wie beim Email-Service (siehe Kap. 3.3) benötigt man auch für andere Cloud-Dienste einen professionellen Anbieter, der sein „Handwerk“ versteht und bei dem die Sicherheit der Kundendaten hohe Priorität genießt. Doch auch ein zuverlässiger Provider

kann mir nie garantieren, dass andere keinen Zugriff auf meine Daten erhalten. Welcher Provider will heute allen Ernstes von sich behaupten, dass er niemals gehackt wird.

- Will man seine Daten in die Cloud verlagern und den Zugriff durch andere ausschließen, so muss man die **Daten verschlüsseln**. Wie bei der Email ist die Datenverschlüsselung kein Ersatz für einen zuverlässigen Provider. Diesen braucht man zusätzlich, denn die möglicherweise anfallenden Metadaten bleiben (wie bei der Email) unverschlüsselt und ebenfalls schutzwürdig.

## 13.2 Kommunikationsdienste

### 13.2.1 Verschlüsselte „SMS“ mit TextSecure

#### 13.2.1.1 SMS und Instant Messaging (*WhatsApp*)

SMS ist ein Kurznachrichtendienst für das Telefon. **SMS werden über das Sprachnetz übertragen und funktionieren deshalb ohne Internetzugang**. Sie funktionieren noch auf den alten Handys, mit denen noch keine Kommunikation über das Internet möglich war. Sie funktionieren auf einem modernen Smartphone auch dann, wenn der Internet-Zugriff (mobile Daten oder WLAN) abgeschaltet ist. Im Telefon muss eine SIM-Karte eingelegt und diese muss in das Mobilnetz eingebucht sein. (Gleiches gilt für MMS. MMS kann außer dem Nachrichtentext auch Multimedia-Daten (z.B. Fotos) übertragen.)

Die Adresse einer SMS ist die Telefonnummer. Deshalb wird eine SMS immer von dem Gerät empfangen, in das die SIM-Karte mit der Empfänger-Telefonnummer eingelegt ist.

SMS werden vom Mobilfunk-Provider in Rechnung gestellt. Sie werden – je nach Tarif – pro versendeter SMS oder (bei einer SMS-Flat) pauschal abgerechnet.

SMS werden unverschlüsselt über das Sprachnetz übertragen und können deshalb wie Telefongespräche abgehört werden.

Die SMS hat (zum Leidwesen der Mobilfunk-Provider) Konkurrenz bekommen durch sog. **Instant Messaging Services**. Der prominenteste Vertreter ist *WhatsApp*. (*WhatsApp* hat Ende August 2014 ca. 600 Millionen registrierte Mitglieder und wurde Anfang 2014 von Facebook für 19 Milliarden Dollar übernommen.)

Mehr zu Instant Messaging unter [http://de.wikipedia.org/wiki/Instant\\_Messaging](http://de.wikipedia.org/wiki/Instant_Messaging) und [http://en.wikipedia.org/wiki/Instant\\_Messaging](http://en.wikipedia.org/wiki/Instant_Messaging).

Mehr zu *WhatsApp* unter <http://de.wikipedia.org/wiki/WhatsApp> und <http://en.wikipedia.org/wiki/WhatsApp>.

Zu WhatsApp gibt es viele Alternativen, die allerdings viel weniger verbreitet sind: [http://de.wikipedia.org/wiki/Liste\\_von\\_mobilen\\_Instant-Messengern](http://de.wikipedia.org/wiki/Liste_von_mobilen_Instant-Messengern).

Was sind die Merkmale von *WhatsApp* und von ähnlichen Instant Messaging Services?

- Es ist ein Kurznachrichtendienst wie SMS, mit dem man auch Fotos und andere Multimedia-Daten übertragen kann (wie bei MMS).
- Anders als bei SMS werden die Nachrichten über das Internet übertragen. Man spricht von Push-Nachrichten. Dementsprechend benötigt man dafür ein Smartphone. Ein dummes, altes Handy genügt nicht, denn es erlaubt keinen Internetzugang.

Die Nachricht wird zunächst vom Absender-Gerät zum Server des Nachrichten-Dienstes

*WhatsApp* übertragen. Hat der Empfänger keine Internetverbindung, muss der Server die Nachricht zwischenspeichern. Ist der Empfänger wieder erreichbar, wird die Nachricht zu ihm übertragen (gepusht). Nach erfolgreicher Übertragung kann die Nachricht vom *WhatsApp*-Server gelöscht werden. *WhatsApp* behauptet dies jedenfalls. Ob und wann sie tatsächlich gelöscht wird, darüber haben Absender und Empfänger allerdings keine Kontrolle.

- Da *WhatsApp*-Nachrichten nicht über das Sprachnetz übertragen werden, tauchen sie auch nicht in der SMS-Abrechnung auf. Die Kosten für die Nutzung des Dienstes (früher kostenlos, heute ein Dollar pro Jahr) sind an den *WhatsApp*-Dienst abzuführen.
- Obwohl die Nachrichten über das Internet und nicht über das Sprachnetz übertragen werden, wird (analog zur SMS) eine Telefonnummer als Zieladresse verwendet.
- Um *WhatsApp* zu nutzen, muss man sich bei diesem Dienst registrieren. Bei SMS ist dagegen keine Registrierung erforderlich. Man kann allerdings nur mit Partnern kommunizieren, die ebenfalls bei *WhatsApp* registriert sind. Diese Einschränkung existiert bei SMS nicht. Ist der gesamte Bekanntenkreis bei *WhatsApp* registriert, dann stellt dies in der Regel keine große Einschränkung dar.
- Die Kommunikationsinfrastruktur ist bei Instant Messaging einfacher als bei der Email. Bei der Email sind in der Regel zwei Provider involviert, der des Absenders und der des Empfängers. Die Email wird auf drei Teilstrecken übertragen: vom Absender zum Absender-Provider, von dort zum Empfänger-Provider und schließlich zum Empfänger. Eine Push-Nachricht benötigt nur einen Provider und zwei Teilstrecken für die gesamte Übertragung: vom Absender zum Provider und von diesem zum Empfänger. Die Email ist ein Provider-übergreifender Nachrichtendienst; man kann eine Email an jede gültige Email-Adresse dieser Welt schicken, gleichgültig bei welchem Provider der Empfänger registriert ist. Instant Messaging beschränkt den Kreis der möglichen Nachrichtenempfänger auf die Benutzer, die bei dem betreffenden Dienst registriert sind. Die gesamte Kommunikation läuft dann über die Server des Providers. Provider-übergreifende Kommunikation ist prinzipiell nicht möglich.

Ist Instant Messaging mit *WhatsApp* nun eine Einschränkung? Meistens nicht, denn man kann sehr viele Nutzer über *WhatsApp* erreichen. Außerdem kann man sich bei mehreren Instant Messaging Diensten gleichzeitig registrieren und diese auch auf demselben Gerät parallel nutzen. Den Freunden, die man weder durch *WhatsApp* noch durch einen anderen Instant Messaging Service erreichen kann, kann man ja immer noch eine evtl. kostenpflichtige SMS schicken.

### 13.2.1.2 ***WhatsApp* unter dem Blickwinkel der Sicherheit und des Privatsphäre-Schutzes**

Die Geschichte von *WhatsApp* ist leider auch die Geschichte der Entdeckung vieler Sicherheitslücken und Datenschutzverletzungen (nachzulesen in den o.g. Wikipedia-Artikeln), von denen ich an dieser Stelle zwei schwerwiegende nennen will:

- Bis August 2012 wurden *WhatsApp*-Nachrichten unverschlüsselt über das Internet übertragen. Die gesamte *WhatsApp*-Kommunikation lies sich damit sehr leicht abgreifen und mitlesen oder aufzeichnen.
- Anfang 2013 wurde bekannt, dass die *WhatsApp*-App das gesamte Adressbuch auf dem Smartphone ausliest und auf die Server von *WhatsApp* überträgt. Dabei wurden auch

Kontakteinträge von Benutzern „gestohlen“, die nicht bei *WhatsApp* registriert waren.

Trotz der immer wieder bekannt gewordenen Sicherheitsmängel ist *WhatsApp* sehr beliebt geblieben und hat immer weitere Benutzer hinzugewonnen. Allein in Deutschland sind es mittlerweile mehr als 30 Millionen registrierte Nutzer.

Bei *WhatsApp* dürften mittlerweile die größten Sicherheitslücken geschlossen sein. Transport-Verschlüsselung ist seit August 2012 implementiert. Der zentrale Kritikpunkt ist die **fehlende Ende-zu-Ende-Verschlüsselung**.

Die Nachrichten werden auf den *WhatsApp*-Servern unverschlüsselt zwischengespeichert. *WhatsApp* ist eine werbefreier Dienst und wirbt damit, die Daten nicht dauerhaft zu speichern und nicht auszuwerten. Die Nutzer haben jedoch keine Kontrolle darüber und müssen dem Dienst vertrauen.

Um den Zugriff auf die Nachrichten durch den Provider (oder durch andere, die sie dem Provider stehlen oder durch Gerichtsbeschluss einfordern) von vorn herein auszuschließen, muss das Zero Knowledge Prinzip gelten. Die Nachrichten sind zu verschlüsseln, sodass der Provider sie gar nicht kennt. Auch Datendiebe sind dann machtlos.

### 13.2.1.3 Alternativen zu *WhatsApp*

Einige alternative Messenger bieten Ende-zu-Ende-Verschlüsselung an. Drei davon waren in letzter Zeit in Deutschland öfters im Gespräch. Auf diese will ich mich hier beschränken.

- **Threema** von der Schweizer Firma Threema GmbH: <https://threema.ch/de> und <https://play.google.com/store/apps/details?id=ch.threema.app>
- **TextSecure** von Open Whispersystems: <https://whispersystems.org/> und <https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms>
- **SIMSmee** von der Deutschen Post AG: <http://sims.me/> und <https://play.google.com/store/apps/details?id=dev.de.dpag.simsme>

Unter diesen drei Messengern ist **Threema** wohl der mit dem höchsten Nutzer-Komfort und mit dem größten Funktionsumfang. Er unterstützt auch verschlüsselte Sprachnachrichten und den Schlüsselaustausch über das Scannen des QR-Codes. *Threema* ist allerdings nicht Open Source und kann deshalb nicht überprüft werden. Als die NSA-Affäre bekannt wurde, sind viele Nutzer sensibler für die Datensicherheit geworden und sind von *WhatsApp* zu *Threema* gewechselt oder sie haben begonnen, *WhatsApp* und *Threema* parallel zu benutzen.

Bei **TextSecure** wurde der Quelltext offen gelegt (Open Source). Dies war für mich das entscheidende Argument für die Installation dieser App auf meinem Android-Smartphone. Edward Snowden hat in einem Interview Moxie Marlinspike, den Hauptentwickler von *TextSecure* für die Offenlegung ausdrücklich hervorgehoben und die Verwendung dieser App empfohlen. *TextSecure* ist nur für Android-Smartphones verfügbar, iPhone-Nutzer installieren und nutzen stattdessen die App *Signal* vom selben Hersteller.

In diesem Jahr hat die Deutsche Post AG den Messenger **SIMSmee** herausgebracht. Auch diese App verschlüsselt die Nachrichten und bietet Unterstützung für Übertragung von Multimedia-Daten. Es dürfte allerdings fraglich sein, ob diese App außerhalb von Deutschland jemals eine große Verbreitung findet.

Ich selbst habe weder mit *SIMSmee* noch mit *Threema* eigene Erfahrungen gesammelt und möchte zum Vergleich der Messenger nochmals auf die schon oben genannte Wikipedia-Seite verweisen:

[http://de.wikipedia.org/wiki/Liste\\_von\\_mobilen\\_Instant-Messengern](http://de.wikipedia.org/wiki/Liste_von_mobilen_Instant-Messengern). In dieser Liste ist allerdings SIMSMe noch nicht vertreten.

### 13.2.1.4 TextSecure auf dem Android-Smartphone nutzen

In den nachfolgenden Unterkapiteln beschreibe ich die wichtigsten Aspekte der Nutzung von *TextSecure*. Weitere Hilfe zur App findet sich im Support Center von Open Whispersystems unter <http://support.whispersystems.org/>.

#### 13.2.1.4.1 Einrichtung

Die App ist wie üblich aus dem Google Play Store zu installieren. Nach dem ersten Start der App wird diese eingerichtet. Dazu muss man ...

- seine Mobilrufnummer eingeben (für die Registrierung beim *TextSecure*-Dienst)
- ein Passwort eingeben und durch eine zweite Eingabe bestätigen. Dieses dient der Verschlüsselung der auf dem Gerät gespeicherten Nachrichten.
- Optional kann man die alten Nachrichten importieren, sodass innerhalb von *TextSecure* verfügbar sind.

Nun muss man noch ein wenig warten, denn die folgenden Schritte werden automatisch und ohne Benutzereingriff von der App und dem *TextSecure*-Server durchgeführt:

- Die App auf dem Smartphone erzeugt einen neuen Schlüsselbund,
- generiert ein Schlüsselpaar aus öffentlichem und privatem Schlüssel und legt diese im Schlüsselbund ab.
- Sie überträgt die Mobilrufnummer zum *TextSecure*-Server.
- Der *TextSecure*-Server sendet eine Bestätigungsmitteilung an mein Smartphone.
- Die App auf dem Smartphone empfängt die Bestätigungsmitteilung.
- Diese bestätigt den Empfang dieser Nachricht wieder an den Server und sendet dabei den öffentlichen Schlüssel mit.
- Nun kann der Server sicher sein, dass die Rufnummer zu dem neuen Gerät gehört und registriert die Rufnummer, die Gerätekennung und den zugehörigen öffentlichen Schlüssel.

Nach der kurzen Wartezeit ist die Einrichtung abgeschlossen und man kann loslegen.

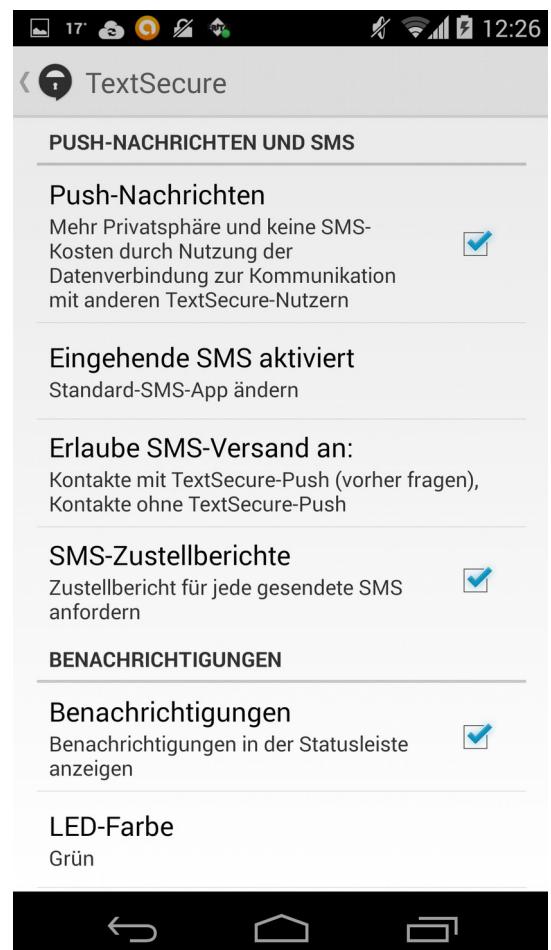


Abbildung 38: TextSecure: Einstellungen

### 13.2.1.4.2 Nachrichten-Versand

Kommuniziert man das erste Mal mit einem Partner, dessen Nummer ebenfalls bei *TextSecure* registriert ist, dann wird man von der App gefragt, ob man mit diesem Partner verschlüsselte Nachrichten austauschen will. Bestätigt man dies, dann erhält man automatisch den öffentlichen Schlüssel des Partners und der Partner erhält den eigenen Schlüssel vom *TextSecure*-Server. Die öffentlichen Schlüssel werden in den Schlüsselbund „eingehängt“.

*TextSecure* unterstützt drei Arten von Text-Nachrichten: verschlüsselte Push-Nachrichten, verschlüsselte SMS und unverschlüsselte SMS.

- **Verschlüsselte Push-Nachrichten** funktionieren nur bei bestehender **Internetverbindung**. Diese werden verwendet, wenn der Empfänger der Nachricht ebenfalls bei *TextSecure* registriert ist und eine Internetverbindung besteht. Push-Nachrichten erscheinen nicht auf der SMS-Rechnung.  
Die verschlüsselte Nachricht wird zunächst vom Absender-Gerät zum Server des Nachrichten-Dienstes *TextSecure* übertragen. Hat der Empfänger keine Internetverbindung, muss dieser die Nachricht zwischenspeichern. Ist der Empfänger wieder erreichbar, wird die Nachricht übertragen (gepusht) und nach erfolgreicher Übertragung vom Server gelöscht. Verschlüsselte Nachrichten sind für den Provider nur Datenmüll. Sie zu speichern, bringt ihm keinen Nutzen (Zero Knowledge).
- **Verschlüsselte SMS über das Sprachnetz** werden verwendet, wenn keine Internetverbindung besteht. Zunächst scheitert die Übertragung, wenn das Gerät keinen Internetzugang hat. Dann kann man die Nachricht nochmals antippen und den Versand als SMS wählen.  
Dies funktioniert allerdings nur, wenn der öffentliche Schlüssel des Empfängers der Nachricht bereits im eigenen Schlüsselbund vorliegt. Der Kommunikationspartner muss also ebenfalls bei *TextSecure* registriert sein und der Schlüsselaustausch muss bei einer früheren Push-Nachricht schon vollzogen worden sein. Verschlüsselte SMS werden vom Mobilfunk-Provider je nach vereinbartem Tarif berechnet.
- Normale, **unverschlüsselte SMS über das Sprachnetz** werden verwendet, wenn der Empfänger nicht bei *TextSecure* registriert ist. Diese sind als SMS kostenpflichtig.

### 13.2.1.4.3 Das Passwort

Das bei der Einrichtung vergebene Passwort wird nicht zur Nachrichtenverschlüsselung verwendet. Diese funktioniert ganz unabhängig vom Passwort.

Das Passwort wird verwendet zur Verschlüsselung der auf dem Smartphone gespeicherten Nachrichten. Diese ist optional.

Man kann bei der Ersteinrichtung das Passwort leer lassen oder man kann es nachträglich löschen. In diesem Fall werden die Nachrichten unverschlüsselt auf dem Gerät abgelegt. Dennoch werden die Nachrichten zur Übertragung verschlüsselt.

### 13.2.1.4.4 Gerätewechsel

Will man seine SIM-Karte in einem anderen Gerät verwenden, dann sollte man die SIM-Karte nicht einfach aus dem alten Gerät herausnehmen und in das neue einstecken. Man verfährt stattdessen folgendermaßen:

- Auf dem alten Smartphone in den *TextSecure*-Einstellungen die Push-Nachrichten deaktivieren. Dadurch wird auch die Registrierung auf dem *TextSecure*-Server für die Rufnummer und der öffentliche Schlüssel für das alte Gerät zurückgenommen.
- Das alte Gerät ausschalten und die SIM-Karte entnehmen
- Die SIM-Karte ins neue Gerät stecken und dieses einschalten
- Auf dem neuen Gerät *TextSecure* installieren und einrichten (siehe Kap. 13.2.1.4.1). Wurde *TextSecure* schon früher installiert und eingerichtet, dann müssen nur die Push-Nachrichten in den Einstellungen wieder aktiviert werden. Dadurch wird die Rufnummer und der öffentliche Schlüssel für das neue Gerät auf dem *TextSecure*-Server registriert.
- Die Einrichtung ist abgeschlossen und verschlüsselte Push-Nachrichten können nun versendet werden.

### 13.2.2 Abhörsichere Telefonate mit RedPhone

Die **RedPhone** ist eine Telefonie-App für Android. Sie erlaubt verschlüsselte und damit abhörsichere Telefonate:

<https://play.google.com/store/apps/details?id=org.thoughtcrime.redphone> .

Die App stammt ebenfalls von Moxie Marlinspike von Open Whispersystems (<https://whispersystems.org/>).

Sie funktioniert ähnlich wie *TextSecure* und ist auch völlig analog einzurichten.

Die App ist nur für Android-Smartphones verfügbar. iPhone-Nutzer können die App *Signal* vom selben Hersteller verwenden.

Weitere Hilfe zu *RedPhone* findet sich im Support Center von Open Whispersystems unter <http://support.whispersystems.org/> .

## 13.3 Cloud-Dienste

Wie oben beschrieben (siehe Kap. 13.1.3) ist die Synchronisation seiner Daten zwischen verschiedenen vernetzten Geräten ein für den Benutzer zentrales Merkmal von Cloud-Diensten.

Mit IMAP verwaltete Email ist das erste Beispiel für einen solchen Dienst. Die Mails werden in einer Struktur von Ordnern auf dem Server des Providers gespeichert. Über das IMAP-Protokoll können mehrere Clients auf diesen Mail-Bestand zugreifen und die Mails anzeigen, löschen, verschieben oder Ordner anlegen, löschen und umbenennen. Da die Modifikationen, die ein Client vornimmt, an zentraler Stelle auf dem Server durchgeführt werden, wird die Änderung die ein Client auf einem Gerät vornimmt meist innerhalb von kurzer Zeit auch für die anderen Clients auf den anderen Geräten sichtbar.

Dieser Synchronisationseffekt, der das lästige und zeitaufwändige Übertragen von Daten zwischen verschiedenen Geräten überflüssig macht, ist nicht nur bei der Email sondern auch bei anderen Diensten erwünscht.

Ich möchte hier auf die Groupware-Dienste (Kontakte Kalender, Aufgabenliste) und die Speicherdiensste à la Dropbox eingehen und sie unter Sicherheitsgesichtspunkten etwas genauer betrachten.

### 13.3.1 Groupware-Dienste beim Email-Provider in Anspruch nehmen

Bei den Groupware-Diensten ist mir kein Angebot bekannt, das die Zero Knowledge Anforderung erfüllt. Um so wichtiger ist hier die Wahl eines **zuverlässigen Providers**.

Die meisten Email-Provider bieten auch Groupware-Dienste an. So hat man mit der Wahl des Email-Providers (siehe Kap. 3.3) meist auch schon seinen Provider für die zentrale Ablage von Kontakten, Kalendern und Aufgabenlisten gefunden.

Der Zugriff auf Kalender, Kontakte und Aufgabenlisten ist – ähnlich wie bei der Email – mit dem Webbrowser möglich. Man ruft die URL des Providers auf, man meldet sich mit Benutzernamen und Passwort an, danach hat man Zugriff auf die Mails (nur unverschlüsselte), auf die Kontaktliste, auf den Kalender und auf die Aufgabenlisten. Das Zugriffsprotokoll ist in diesem Fall HTTPS.

Dieser Zugriffsweg ist jedoch nicht ausreichend. In der Regel möchte man nicht mit dem Browser auf die Mails zugreifen, sondern mit einem spezialisierten Email-Client wie *Outlook* oder *Thunderbird* oder *K-@ Mail* auf dem mobilen Gerät. Analog verwendet man auch auf dem PC oder dem Smartphone für die Verwaltung der Kontaktdateien, des Termine und Aufgaben je einen spezialisierten Client.

Wie wir wissen (siehe Kap. 2.5) ist IMAP das Protokoll für den den Zugriff auf den Mail-Bestand. Das Standard-Protokoll für den Zugriff auf die Kontakte ist CardDAV. Zum Zugriff auf Kalender und Aufgabenlisten wird standardmäßig CalDAV verwendet.

#### 13.3.1.1 Zugriff mit CardDAV und CalDAV

Das Basisprotokoll für diese beiden Protokolle ist **WebDAV** (**W**ebsite **D**istributed **A**uthoring and **V**ersioning).

Das **CardDAV**-Protokoll (**C**ard **D**istributed **A**uthoring and **V**ersioning) ist eine WebDAV-Spezialisierung zur Verwaltung zentral gespeicherter Kontakt-Daten. (Ein Kontakt wird in einer einzeln vCard gespeichert, daher kommt die Protokollbezeichnung CardDAV.)

Das **CalDAV**-Protokoll (**C**alendaring **E**xtensions to **W**ebsite **DAV**) ist eine WebDAV-Spezialisierung zur Verwaltung zentral gespeicherter Kalenderdaten. Da eine einzelne Aufgabe einem Termin – technisch gesehen – recht ähnlich ist, unterstützt dieses Protokoll auch Aufgabenlisten. (Eine Aufgabe ist gewissermaßen ein Termin ohne Datum.)

Ich gehe auf diese Protokolle an dieser Stelle nicht näher ein, sondern verweise den Interessierten nur auf folgende Links im Netz:

- WebDAV: <http://en.wikipedia.org/wiki/WebDAV>
- CardDAV: <http://en.wikipedia.org/wiki/CardDAV> und <http://carddav.calconnect.org/>
- CalDAV: <http://en.wikipedia.org/wiki/CalDAV> und <http://caldav.calconnect.org/>

Für die erforderlichen Konfigurationsschritte ist ein tieferes Verständnis dieser Protokolle nicht erforderlich.

Was benötigt man nun, um auf das Adressbuch oder den Kalender (oder die Aufgabenlisten) nicht mit dem Browser sondern mit einem lokal installierten Programm zuzugreifen, sodass man mit dem lokalen Programm auf die Einträge (Kontakte, Termin, Aufgaben) zugreifen und sie anlegen, ändern, löschen und anzeigen kann? Grundsätzlich benötigt man ...

- einen CardDAV- bzw. CalDAV-Protokolltreiber

- und ein Adressbuch- oder Kalenderprogramm oder ein Programm zur Anzeige und Verwaltung von Aufgabenlisten.

### 13.3.1.2 Zugriff auf das zentrale Adressbuch auf dem PC mit *Thunderbird*

Um das beim Provider gespeicherte zentrale Adressbuch in *Thunderbird* verfügbar zu machen, ist das Add-on **Sogo Connector** zu installieren (analog zur Installation von *Enigmail*, siehe Kap. 7.1.1). Dies ist ein Treiber für die Protokolle CardDAV und für CalDAV. Ein separates Adressbuch-Programm ist nicht erforderlich, denn *Thunderbird* kann nicht nur Emails, sondern auch Adressbücher verwalten.

Der *Sogo Connector* versetzt *Thunderbird* in die Lage, auf CardDAV-Adressbücher an einer bestimmten Adresse zuzugreifen. Um den CardDAV-Zugriff zu konfigurieren, benötigt man folgende Zugangsdaten:

- **CardDAV-URL:** Als Adresse dient eine HTTPS-Url, die man bei seinem Provider in Erfahrung bringen muss. Meist wird man auf den Hilfeseiten des Providers fündig und findet dort die Adresse bzw. das Adressschema. Bei *mailbox.org* ist dies z. B. <https://dav.mailbox.org/carddav/XXX> (XXX = Ihre Ordner-ID). Dies kann jedoch bei jedem Provider anders aussehen.
- **Benutzername** zur Anmeldung beim Provider
- **Passwort** zur Anmeldung

Mit diesen Informationen kann man in *Thunderbird* ein neues Adressbuch anlegen. Dazu verfährt man folgendermaßen:

- Unter dem Menüpunkt *Fenster* → *Adressbuch* die Adressbuch-Verwaltung öffnen
- Unter *Datei* → *Neu* → *Remote-Adressbuch* öffnet sich ein Dialog, in dem die oben aufgeführte CardDAV-URL für ein neues Adressbuch einzutragen ist. Dabei ist die Option „*Nur lesbar*“ zu setzen.
- Beim Klick auf „OK“ wird das Adressbuch erstellt. Vor der ersten Synchronisation erscheint ein Fenster zur Eingabe der Anmeldedaten.
- Die Synchronisation startet und die auf dem Server gespeicherte Kontaktliste wird in *Thunderbird* sichtbar.

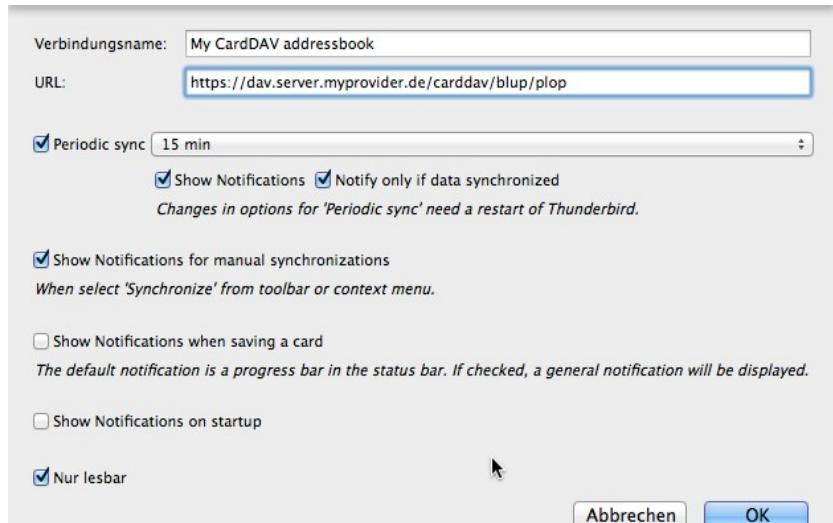


Abbildung 39: *Thunderbird*: Ein Remote-Adressbuch erstellen

Der Zugriff auf Remote-Adressbücher ist leider instabil, sodass Adressen manchmal verstümmelt werden. Aus diesem Grund empfehle ich, nur lesenden Zugriff auf das entfernte Adressbuch zu erlauben. Als Vorsorge gegen Adressverluste sollte das Adressbuch in regelmäßigen Abständen

gesichert/exportiert werden.

Bei mehreren Kontaktlisten ist dieser Vorgang für jede Liste zu wiederholen.

### 13.3.1.3 Zugriff auf den zentralen Kalender auf dem PC mit *Thunderbird*

Um den beim Provider gespeicherten zentralen Kalender in *Thunderbird* verfügbar zu machen, sind die Add-ons **Sogo Connector** und **Lightning** zu installieren (analog zur Installation von *Enigmail*, siehe Kap. 7.1.1). *Sogo Connector* ist ein Treiber für die Protokolle CardDAV und für CalDAV. *Lightning* erweitert *Thunderbird* um die Kalenderfunktionen.

Der *Sogo Connector* versetzt *Thunderbird* in die Lage, auf CalDAV-Kalender an einer bestimmten Adresse zuzugreifen. Um den CalDAV-Zugriff zu konfigurieren, benötigt man folgende Zugangsdaten:

- **CalDAV-URL:** Als Adresse dient eine HTTPS-URL, die man bei seinem Provider in Erfahrung bringen muss. Meist wird man auf den Hilfeseiten des Providers fündig und findet dort die Adresse bzw. das Adressschema. Bei *mailbox.org* ist dies z. B. <https://dav.mailbox.org/caldav/XXX> (XXX = Ihre Ordner-ID). Dies kann jedoch bei jedem Provider anders aussehen.
- **Benutzername** zur Anmeldung beim Provider
- **Passwort** zur Anmeldung

Mit diesen Informationen kann man in *Thunderbird* einen neuen Kalender anlegen. Dazu verfährt man folgendermaßen:

- Unter dem Menüpunkt *Termine und Aufgaben* → *Kalender* die Kalender-Verwaltung öffnen
- Unter *Datei* → *Neu* → *Kalender* öffnet sich ein Dialog, in dem man zwischen einem lokalen und einem Netzwerk-Kalender auswählen kann.
- Man wählt „Netzwerk-Kalender“ und klickt auf „Fortsetzen“. Ein weiterer Dialog verlangt die Auswahl des Kalender-Formats und die Kalender-URL. Man wählt das Format „CalDAV“, gibt die oben aufgeführte CalDAV-URL für den neuen Kalender ein.
- Beim Klick auf „Fortsetzen“ öffnet sich ein weiterer Dialog, in dem man den Namen des Kalenders, seine Farbe und die Email-Adresse des Accounts erfasst.
- Beim Klick auf „Fortsetzen“ wird der Kalender erstellt. Vor der ersten Synchronisation erscheint ein Fenster zur Eingabe der Anmeldedaten.
- Die Synchronisation startet und der auf dem Server gespeicherte Kalender wird in *Thunderbird* sichtbar.

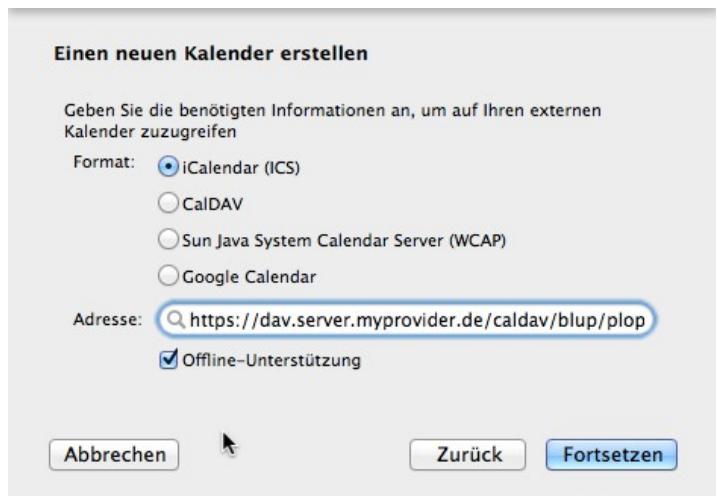


Abbildung 40: *Thunderbird: Einen Netzwerk-Kalender erstellen*

Bei mehreren Kalendern ist dieser Vorgang für jeden zu wiederholen.

### 13.3.1.4 Zugriff auf eine zentrale Aufgabenliste auf dem PC mit Thunderbird

Eine Aufgabenliste nur eine besondere Art von Kalender. Deshalb kommt hier ebenfalls das CalDAV-Protokoll zur Anwendung. Die Konfiguration funktioniert exakt wie die im vorigen Kapitel beschriebene Kalender-Konfiguration. Statt der CalDAV-URL eines Kalenders gibt man die CalDAV-URL einer Aufgabenliste an.

Bei mehreren Aufgabenlisten ist dieser Vorgang für jede Liste zu wiederholen.

### 13.3.1.5 Zugriff auf das zentrale Adressbuch auf dem Android-Gerät

Um eine beim Provider gehostete Kontaktliste auf dem Android-Gerät verfügbar zu machen, benötigt man eine Kalender-App und einen CardDAV-Protokolltreiber. Auch der Protokolltreiber ist eine App, in der man (analog zum *Sogo Connector* in *Thunderbird*) die Zugriffseinstellungen vornehmen kann.

Android hat mehrere Kontakte-Apps im Play Store Angebot. Die vorinstallierte **Kontakte-App** von Google erfüllt den Zweck.

Auch bei den CardDAV-Protokolltreiber-Apps stehen mehrere zur Auswahl. Auf meinen Android-Geräten ist **CardDAV-Sync beta** von Marten Gadja im Einsatz. Obwohl die App noch das *beta* im Namen trägt, hatte ich bisher keine Schwierigkeiten mit der App:  
<https://play.google.com/store/apps/details?id=org.dmfs.carddav.Sync>

Hier die Schritte zur Installation von *CardDAV-Sync beta* und zur Erstellung eines neuen CardDAV-Kontos:

- die App auf dem Android-Gerät installieren
- die App starten
- ein neues CardDAV-Konto anlegen und die Zugangsdaten eingeben:
  - CardDAV-URL oder CardDAV-Server
  - Benutzername
  - Passwort
- eines der auf dem Server verfügbaren Adressbücher auswählen.
- einen Namen für das neue CardDAV-Konto vergeben

Nach der Erstellung des neuen Kontos ist die Kontakte-App zu starten. Bei mehreren auf dem Gerät verfügbaren Konten kann man hier das Konto auswählen, das die in der App anzuzeigenden Kontakte bereitstellen soll.

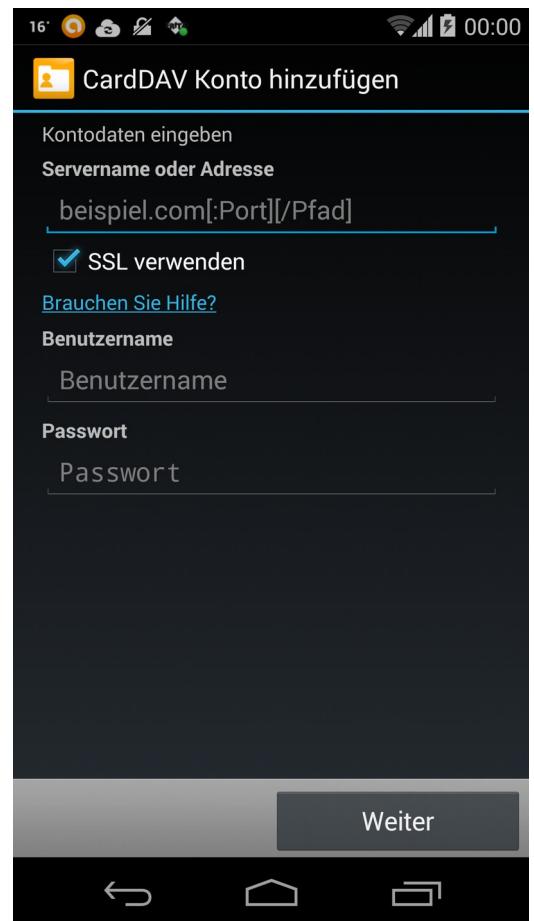


Abbildung 41: CardDAV-Sync: Neues CardDAV-Konto erstellen

### 13.3.1.6 Zugriff auf den zentralen Kalender auf dem Android-Gerät

Um einen beim Provider gehosteten Kalender auf dem Android-Gerät verfügbar zu machen, benötigt man eine Kalender-App und einen CalDAV-Protokolltreiber. Auch der Protokolltreiber ist eine App, in der man (analog zum *Sogo Connector* in *Thunderbird*) die Zugriffseinstellungen vornehmen kann.

Android hat eine große Auswahl an Kontakte-Apps im Play Store Angebot. Die vorinstallierte **Kalender-App** von Google erfüllt den Zweck genauso. Sehr gerne nutze ich auch die App **aCalendar** von Tapir Apps GmbH:

<https://play.google.com/store/apps/details?id=org.withouthat.acalendar>

Auch bei den CalDAV-Protokolltreiber-Apps stehen mehrere zur Auswahl. Auf meinen Android-Geräten ist **CalDAV-Sync** von Marten Gadja im Einsatz:  
<https://play.google.com/store/apps/details?id=org.dmfs.caldav.lib>

Völlig analog zur CardDAV-Konfiguration sind die Schritte zur Installation von *CalDAV-Sync* und zur Erstellung eines neuen CalDAV-Kontos:

- die App auf dem Android-Gerät installieren
- die App starten
- ein neues CalDAV-Konto anlegen und die Zugangsdaten eingeben:
  - CalDAV-URL oder CalDAV-Server
  - Benutzername
  - Passwort
- Nun wird die Liste der auf dem Server verfügbaren Kalender und Aufgabenlisten angezeigt. Diejenigen, die auf dem Android-Gerät bereitstehen sollen, sind (durch das Setzen eines Hakens) auszuwählen.
- einen Namen für das neue CalDAV-Konto vergeben

Nach der Erstellung des neuen Kontos ist die Kalender-App zu starten. Bei mehreren auf dem Gerät verfügbaren Kalender-Konten kann man hier die Kalender auswählen, die die in der App anzuzeigenden Kalenderdaten bereitstehen sollen.

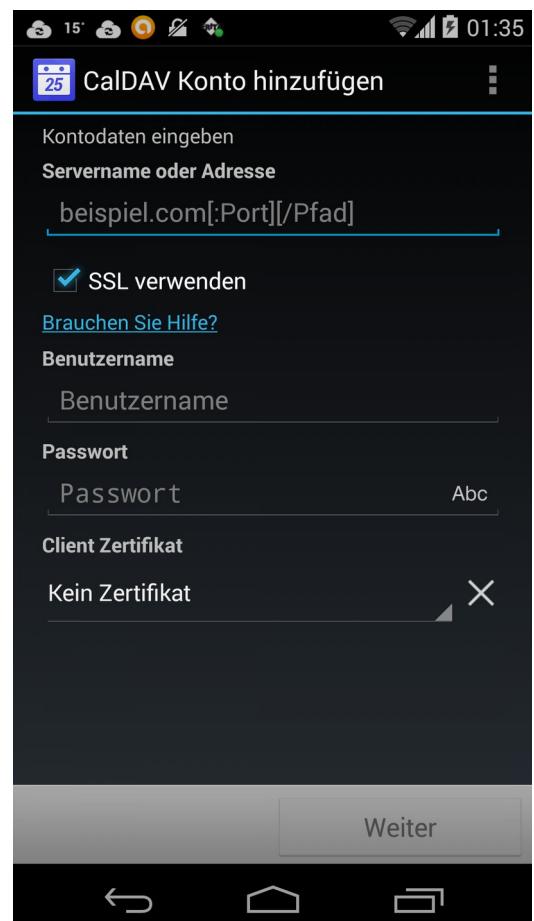


Abbildung 42: CalDAV-Sync: Neues CalDAV-Konto erstellen

### 13.3.1.7 Zugriff auf die zentralen Aufgabenlisten auf dem Android-Gerät

Eine Aufgabenliste nur eine besondere Art von Kalender. Mit der oben beschriebenen CalDAV-Konfiguration ist das Konto für die Aufgabenlisten bereits konfiguriert.

Nun benötigt man noch eine App zur Anzeige und Bearbeitung der Aufgabenlisten. Auch hier gibt es wieder eine reichhaltige Auswahl im Google Play Store. Auf meinen Android-Geräten kommt die App **Tasks** von Marten Gadja zum Einsatz:

<https://play.google.com/store/apps/details?id=org.dmfs.tasks>

Bei der Erstellung des CalDAV-Kontos hat man schon die Aufgabenlisten, die auf dem Gerät bereitstehen sollen, konfiguriert. In der Aufgaben-App *Tasks* kann man davon die Listen auswählen, die in der App angezeigt werden.

### 13.3.1.8 CardDAV und CalDAV mit anderen Programmen und Geräten

In den vorangehenden Kapiteln habe ich gezeigt, wie man CardDAV und CalDAV mit Thunderbird nutzt. Dies funktioniert auf den gängigen PC-Betriebssystemen Windows, Mac OS X und Linux. Ich habe auch gezeigt, wie man diese Protokolle auf dem Android-Smartphone oder Tablet einrichten kann.

Für beide Protokolle gibt es Clients für andere Systeme oder Programme. Neben Erweiterungen für *Outlook* gibt es die passenden Clients für iOS und für Windows Phone.

Eine Übersicht über die verfügbaren CardDAV-Clients ist hier zu finden:

<http://carddav.calconnect.org/implementations/clients.html>

Eine Übersicht über die verfügbaren CalDAV-Clients ist hier zu finden:

<http://caldav.calconnect.org/implementations/clients.html>

### 13.3.2 Cloud Storage

**Cloud Storage – Dateien und Ordner über die Cloud synchronisieren** und damit auf allen Geräten denselben Datenbestand zur Verfügung zu haben – dafür steht der Name *Dropbox*. *Dropbox* hat diese Art Dienst salonfähig gemacht und gilt auch heute noch als Referenz für Online-Speicher.

#### 13.3.2.1 Viele Cloud Storage Anbieter

Doch es gibt unzählige weitere Anbieter, die Cloud Storage anbieten. Alle großen Internetkonzerne bieten neben vielen anderen Diensten auch die Speicher-Dienste an: *Amazon Cloud Drive*, *Apple iCloud*, *Google Drive*, *Microsoft OneDrive*. Andere sind spezialisierte Cloud Speicher Dienste: *Bitcasa*, *Box*, *Dropbox*, *MediaFire*, *Mega*, *SugarSync* und viele weitere. Viele Mail-Provider bieten ebenfalls Online-Festplatten (so wird diese Dienstart im Deutschen auch gerne bezeichnet): *GMX*, *WEB.DE*, *1&1, freenet*, *MyKolab*, *mail.de*, *mailbox.org* – eine kleine Aufzählung ohne Anspruch auf Vollständigkeit. Selbstverständlich sind auch die Telekommunikationsanbieter mit von der Partie: *Telekom*, *Vodafone* und *O2*, um die großen deutschen Anbieter zu nennen, haben Online-Speicher im Angebot.

Einen Vergleich der Cloud Storage Anbieter ist auf folgenden Wikipedia-Seiten zu finden:

- [http://en.wikipedia.org/wiki/Comparison\\_of\\_file\\_synchronization\\_software](http://en.wikipedia.org/wiki/Comparison_of_file_synchronization_software)
- [http://en.wikipedia.org/wiki/Comparison\\_of\\_file\\_hosting\\_services](http://en.wikipedia.org/wiki/Comparison_of_file_hosting_services)

Diese Liste ist sehr umfangreich und kann doch nur unvollständig sein. Viele Mail- und Telekom-Provider sind auf der Liste nicht aufgeführt.

#### 13.3.2.2 Funktionsweise der Synchronisation

Damit die automatische Synchronisation der Ordner und funktioniert, ist ein Synchronisations-Client auf dem Rechner oder auf dem mobilen Gerät zu installieren. Nach der Installation wird der Client mit den Zugangsdaten (Benutzername und Passwort) beim Cloud Storage Server angemeldet.

Er erhält damit Zugriff auf die Ordner und Dateien auf dem Server. Danach überwacht der Client ständig die Dateistruktur auf dem Client und dem Server und gleicht sie aneinander an. Gibt es eine Änderung (Hinzufügung, Löschung oder Modifikation einer Datei oder eines Ordners) auf dem Server, so wird die Änderung sofort auf dem lokalen System nachgezogen. Umgekehrt werden Änderungen auf dem Client sofort auf dem Server nachgezogen.

Am Beispiel *Dropbox* sieht die Installation und Anwendung so aus: Man lädt den *Dropbox*-Client von der Website von *Dropbox* (<https://www.dropbox.com/>) herunter, man installiert den Client auf dem PC und startet ihn. Man gibt seine *Dropbox*-Zugangsdaten ein und gibt noch an, welche Ordner synchronisiert werden sollen. Standardmäßig wird im Benutzerordner ein Unterordner *Dropbox* angelegt, der dann synchronisiert wird.

Nun muss man sich um nichts mehr kümmern. Alles was man in den *Dropbox*-Ordner wirft wird automatisch synchronisiert. Auch der Neustart des Rechners ist kein Problem. Nach dem Hochfahren und der Benutzeranmeldung wird der Dienst automatisch gestartet, er meldet sich automatisch wieder beim *Dropbox*-Server an (das Passwort merkt er sich; man muss es nicht jedes Mal neu eingeben) und überwacht permanent die Daten auf dem Server und den lokalen *Dropbox*-Ordner und alle seine Unterordner auf Änderungen und gleicht diese miteinander ab.

Unter Android installiert man die *Dropbox*-App aus dem Google Play Store (<https://play.google.com/store/apps/details?id=com.dropbox.android>). Nach der Anmeldung mit den *Dropbox*-Zugangsdaten beginnt die Synchronisation und stellt die Dateien in der *Dropbox*-App zur Verfügung.

*Dropbox* bei Wikipedia: <http://de.wikipedia.org/wiki/Dropbox> und [http://en.wikipedia.org/wiki/Dropbox\\_\(service\)](http://en.wikipedia.org/wiki/Dropbox_(service))

*Dropbox* ist nur der prominenteste Vertreter dieser Gattung. Alle anderen Cloud Speicher Dienste funktionieren im Prinzip genau so.

### 13.3.2.3 Weitere Merkmale der Cloud Storage Dienste

Synchronisation von Ordnern und Dateien, dies ist das zentrale Cloud Storage Feature, das alle Provider anbieten. Doch gibt weitere Merkmale, in denen sie sich unterscheiden und die auch für die Provider-Auswahl ausschlaggebend sein können.

- Unterstützung für verschiedene Betriebssysteme. Z.B. stellen nicht alle Anbieter auch einen Synchronisations-Client für Linux zur Verfügung.  
So gibt es für *Google Drive* keinen offiziellen Linux Sync-Client von Google. Linux-Nutzer, die dennoch ihre Dateien über *Google Drive* synchronisieren wollen, finden eine Alternative in dem Client von *insync* (<https://www.insynchq.com/>).
- Sharing von Ordnern und Dateien zwischen verschiedenen Benutzern
- Öffentliche Ordner
- Ordner als Fotogalerie
- Datei-Versionierung: erlaubt die Wiederherstellung gelöschter Dateien oder alter Dateiversionen
- Eignung für Online-Backup
- Zwei-Faktor-Authentifizierung: Zusätzlich zu Benutzername und Passwort muss man bei der Anmeldung ein weiteres Secret (z.B. ein per SMS zugestelltes Einmalpasswort)

angeben. Dadurch kann der Account nicht so leicht von anderen gekapert werden.

- Ende-zu-Ende-Verschlüsselung
- Alle bieten ihren Dienst mit einem kostenlosen Freispeicher-Angebot zwischen zwei und fünfzig GB. Wer mehr Speicher braucht, kann diesen im monatlichen oder jährlichen Abo erwerben. Z.B. sind bei *Dropbox* (Stand Oktober 2014) 1000 GB für monatlich 10,- € erhältlich. Andere Anbieter sind noch günstiger.

### 13.3.2.4 Sicherheitsfragen

Transport-Verschlüsselung ist heute kein außergewöhnliches Feature mehr, durch das sich ein Provider hervorheben kann. Doch damit sind die Daten nur verschlüsselt, solange sie durch das Internet reisen. Auf dem Server des Providers liegen die Daten dennoch unverschlüsselt. Damit haben außer dem Provider auch Hacker, die dort eindringen, und staatliche Stellen, die ihre Herausgabe erzwingen, darauf Zugriff.

Tatsächlich behaupten viele Provider, dass sie die Daten auch auf ihren Servern verschlüsseln. Allerdings liegen die Schlüssel ebenfalls auf den Servern der Provider. Dies ist nur ein kleiner Sicherheitsgewinn, da der Provider auch die Schlüssel dazu hat. Ein Datendieb muss dann nicht nur die Daten stehlen sondern auch noch die Schlüssel. Die Hürde für den Hacker oder die NSA ist ein wenig höher, jedoch nicht unüberwindbar. Das Problem dabei ist, dass der Schlüssel nicht vom Anwender kontrolliert wird.

Gegen die Kompromittierung meiner Daten ist – wie auch bei anderen Anwendungen – nur ein Kraut gewachsen: **Zero Knowledge** alias **Ende-zu-Ende-Verschlüsselung**. Was der Provider nicht kennt, das kann ihm auch nicht gestohlen oder durch Zwang abgepresst werden. Die Daten müssen auf meinem Gerät (PC, Tablet oder Smartphone) verschlüsselt werden, bevor sie dieses verlassen, und sie werden auch nur auf meinen Geräten wieder entschlüsselt. Der Schlüssel dazu muss immer auf meinen Geräten verbleiben und darf niemals auf den Server wandern.

Dies ist technisch nicht ganz richtig. Der Schlüssel wird sehr wohl auf den Server des Providers übertragen. Allerdings wird der Schlüssel zuvor mit meinem Passwort verschlüsselt und kann dann auch nur mit meinem Passwort wieder entschlüsselt werden. So kann auch der verschlüsselte Schlüssel über die Cloud zwischen den Geräten synchronisiert werden. Will ich auf meinen Geräten auf die Daten zugreifen, so muss ich zunächst das Passwort eingeben, um den Schlüssel zu entschlüsseln. Danach erst ist der Schlüssel nutzbar, sodass die Ordner und Dateien entschlüsselt werden.

Es gibt zwei Strategien, die Ende-zu-Ende-Verschlüsselung zu realisieren:

- Man wählt einen sicheren Provider, der Ende-zu-Ende-Verschlüsselung anbietet (siehe Kap. 13.3.2.5).
- Man wählt einen unsicheren Anbieter und verschlüsselt die Dateien mit einem zusätzlichen Verschlüsselungsprogramm (siehe Kap. 13.3.2.6).

### 13.3.2.5 Cloud-Speicher mit Ende-zu-Ende-Verschlüsselung

In diesem Kapitel stelle kurz die mir bekannten Provider vor, die Ende-zu-Ende-Verschlüsselung anbieten. Ein weiteres Auswahlkriterium ist, dass der betreffende Dienst die wichtigsten PC- und Mobil-Betriebssysteme unterstützt: Windows, Mac OSX, Linux, Android und iOS. Manche Dienste haben keinen Synchronisations-Client für Linux im Angebot.

Eine Information will ich noch vorausschicken. Die Entscheidung für einen Provider schließt den anderen nicht aus. Es ist kein Problem, mehrere Cloud Storage Dienste auf demselben System zu

installieren und zu nutzen. Ich hatte schon bis zu acht Dienste parallel installiert und hatte keine Schwierigkeiten damit.

- **Bitcasa:** ist ein sehr ausgereifter und benutzerfreundlicher Dienst. Dieser Dienst ist wohl der einfachste Einstieg in verschlüsselten Cloud Speicher für Privatnutzer.
  - Ende-zu-Ende-Verschlüsselung: ja
  - Unterstützte Betriebssysteme: Windows, Mac OS X, Linux, Android, iOS, Windows Phone, FirefoxOS
  - Freispeicher: 6 GB (bis zu 20 GB durch Empfehlungen)
  - Datei-Versionierung: ja
  - Open Source: nein
  - Website: <https://bitcasa.com/>
  - Wikipedia: <http://en.wikipedia.org/wiki/Bitcasa>
- **Mega:** ist das Angebot von Kim Dotcom aus Neu Seeland. Der Dienst sticht durch sein großes Freispeicher-Angebot heraus. Er ist im Datei-Manager und im Browser sehr komfortabel zu benutzen. Der Dienst bietet bislang allerdings noch keine Datei-Versionierung, mit der sich alte Dateiversionen wieder herstellen lassen. Außerdem gibt es für den kostenlosen Account eine Bandbreitenbegrenzung, die sich vermutlich erst bei häufiger Nutzung oder bei großen Up- und Downloads bemerkbar machen dürfte. (Bei einem kleinen Speed-Test habe ich 300 Fotos (1,4 GB) in ca. 30 Minuten hochgeladen. Für die normale Privatnutzung dürfte dies ausreichend sein.)  
(Kim Dotcom steht übrigens in den USA wegen Urheberrechtsverletzungen unter Anklage. Er soll auf seiner mittlerweile geschlossenen Plattform Megapload urheberrechtlich geschütztes Material zum illegalen Download zur Verfügung gestellt haben. Die USA hat einen Auslieferungsantrag an Neu Seeland gestellt, der aktuell (im Herbst 2014) noch nicht entschieden ist.)
  - Ende-zu-Ende-Verschlüsselung: ja
  - Unterstützte Betriebssysteme: Windows, Mac OS X, Linux, Android, iOS, Blackberry
  - Freispeicher: 50 GB
  - Datei-Versionierung: nein
  - Open Source: nein
  - Website: <https://mega.co.nz/>
  - Wikipedia: [http://en.wikipedia.org/wiki/Mega\\_\(service\)](http://en.wikipedia.org/wiki/Mega_(service))
- **SpiderOak:** ist primär ein Online-Backup-Dienst als ein Synchronisationsdienst. Er ist nicht sehr intuitiv zu bedienen und richtet sich mehr an kommerzielle Benutzer als an Privatkunden. Ein Teil des Quelltextes ist Open Source. Edward Snowden hat in einem Interview *SpiderOak* als sichere Alternative zu *Dropbox* und vielen anderen Speicher-Diensten hervorgehoben.
  - Ende-zu-Ende-Verschlüsselung: ja
  - Unterstützte Betriebssysteme: Windows, Mac OS X, Linux, Android, iOS
  - Freispeicher: 2 GB

- Datei-Versionierung: ja
- Open Source: teilweise
- Website: <https://spideroak.com/>
- Wikipedia: <http://en.wikipedia.org/wiki/SpiderOak>
- **TeamDrive**: ist eine deutscher Anbieter aus Hamburg. Er richtet sich ebenfalls mehr an kommerzielle als an Privatkunden. Außer zur Synchronisation eignet sich *TeamDrive* besonders auch für das Online-Backup.
  - Ende-zu-Ende-Verschlüsselung: ja
  - Unterstützte Betriebssysteme: Windows, Mac OS X, Linux, Android, iOS
  - Freispeicher: 2 GB (bis zu 10 GB durch Empfehlungen)
  - Datei-Versionierung: ja
  - Open Source: nein
  - Website: <http://www.teamdrive.com/>
  - Wikipedia: <http://en.wikipedia.org/wiki/TeamDrive>

*SpiderOak* und *TeamDrive* sind nicht ganz so komfortabel zu benutzen. Bei diesen Diensten steht die Backup-Funktion im Vordergrund. Sie richten sich eher an professionelle Kunden. Dem Privatkunden, der einen kostenlosen Account sucht, bieten sie nur 2 GB Freispeicher.

*Bitcasa* und *Mega* sind für die private Nutzung zu empfehlen. Sie bieten ausreichend Freispeicher (*Bitcasa* 6 GB, *Mega* 50 GB) und sind komfortabel zu bedienen. Ich empfehle, beide Dienste parallel zu installieren und zu nutzen und einfach auszuprobieren, an welchem man mehr Gefallen findet.

Dass der Schlüssel nicht auf die Server des Providers übertragen wird sondern auf meinen Geräten verbleibt, lässt sich nicht überprüfen. Damit bleibt die Ende-zu-Ende-Verschlüsselung bzw. Zero Knowledge eine Behauptung des Providers, der man vertrauen muss. Überprüfen lässt sie sich nicht.

Nur *SpiderOak* legt den Quellcode seiner Software offen (Open Source). Der Dienst wurde deshalb von Edward Snowden in einem Interview ausdrücklich als *Dropbox*-Alternative empfohlen.

### 13.3.2.6 Cloud-Speicher unverschlüsselt und ein zusätzliches Verschlüsselungsprogramm

Bei dieser Variante wählt man einen unsicheren Cloud Storage Anbieter. Ein Unterordner des Cloud Storage Hauptordners (und alle Dateien und Ordner, die darunter liegen) wird mit einem Verschlüsselungsprogramm verschlüsselt.

Das komfortabelste Verschlüsselungsprogramm für diesen Zweck, das zurzeit auf dem Markt verfügbar ist, ist sicherlich **Boxcryptor** (<https://www.boxcryptor.com/> und <http://de.wikipedia.org/wiki/BoxCryptor>). Auf Alternativen zu *Boxcryptor* werde ich an dieser Stelle nicht eingehen.

*Boxcryptor* wird in zwei Varianten angeboten: *Boxcryptor 2* und *Boxcryptor Classic*. Von beiden Varianten gibt es je eine kostenlose Version, die allerdings keine Verschlüsselung von Dateinamen

bietet. Da manchmal auch die Dateinamen bereits einiges über den Inhalt der Datei verraten, ist es sinnvoll, auch die Dateinamen zu verschlüsseln. Damit scheiden die kostenlosen Varianten für mich aus.

- **Boxcryptor 2** (<https://www.boxcryptor.com/de/preise>) ist die funktionsreichere Variante und wird im Abo-Modell vermarktet. Für die Unlimited Personal Edition sind aktuell (Okt. 2014) 36,- Euro pro Jahr zu entrichten. *Boxcryptor 2* unterstützt alle gängigen Betriebssysteme außer Linux (<https://www.boxcryptor.com/de/download>). Neben Windows, Mac OS X, Android, iOS werden auch Windows Phone und Blackberry unterstützt. Ein wesentliches Feature von *Boxcryptor 2* ist die Unterstützung von Gruppen. Damit lassen sich verschlüsselte Inhalte zwischen Benutzern derselben Gruppe teilen. Für kollaborative Umgebungen kann dies wichtig sein, für Privatnutzer meist nicht.
- **Boxcryptor Classic** (<https://www.boxcryptor.com/de/classic>) dürfte für Privatnutzer ausreichend sein. Diese Variante bietet neben der Unterstützung von Windows, Mac OS X, Android und iOS auch Linux-Unterstützung. Nur die kostenpflichtige Variante unterstützt die Verschlüsselung von Dateinamen. Die Classic Unlimited Personal Edition kostet aktuell (Okt. 2014) einmalig 35,- Euro.

*Boxcryptor* ist für alle gängigen Cloud Storage Provider verfügbar. Die lange Liste der unterstützen Cloud-Dienste findet sich unter <https://www.boxcryptor.com/de/provider>. Die bekannten großen Anbieter (*Dropbox*, *Google Drive*, *Microsoft OneDrive* etc.) sind alle dabei. Möglicherweise funktionieren auch weitere Dienste, die nicht auf der Liste aufgeführt sind.

In den folgenden Ausführungen werde ich mich auf *Dropbox* in Verbindung mit *Boxcryptor Classic* (Unlimited Personal Edition) beziehen und gehe davon aus, dass *Dropbox* bereits fertig und funktionsfähig auf dem PC (mit Windows, Mac OS X oder Linux) und auf einem Android-Gerät (Smartphone und/oder Tablet) eingerichtet wurde. Genau so gut wie *Dropbox* lässt sich fast jeder andere Anbieter gemeinsam mit *Boxcryptor* verwenden.

Nach der Installation von *Boxcryptor Classic* auf dem PC startet man die Anwendung. Bei der Einrichtung ...

- wählt man einen Cloud Storage Provider (oder einen Ordner im Dateisystem). Ich wähle hier *Dropbox*, um die verschlüsselten Dateien mit *Dropbox* zu synchronisieren. Der Inhalt dieses Ordners wird verschlüsselt.
- Man vergibt den Namen für ein virtuelles Laufwerk (Vorgabe: *Boxcryptor*). In diesem virtuellen Laufwerk wird der verschlüsselte Inhalt dem Benutzer unverschlüsselt bereitgestellt.
- Schließlich vergibt man ein gutes Passwort (siehe 12.1). Das Passwort dient der Verschlüsselung des Schlüssels.

Die Anwendung legt unterhalb des *Dropbox*-Ordners einen Unterordner an (Vorgabe: *Boxcryptor.bc*) an. Unterhalb des Ordners *Boxcryptor.bc* sind alle Ordner und Dateien verschlüsselt (siehe Abb. 43). Sie werden im virtuellen Laufwerk *Boxcryptor* unverschlüsselt zur Verfügung gestellt (siehe Abb. 44). Nun kann man ganze Ordnerstrukturen, die man verschlüsseln will, in das virtuelle Laufwerk kopieren oder verschieben. Diese werden dann automatisch von *Boxcryptor* verschlüsselt in *Dropbox/Boxcryptor.bc* gespeichert und in verschlüsselter Form mit dem *Dropbox*-Server und von dort mit den übrigen bei *Dropbox* angemeldeten Geräten synchronisiert. Der Schlüssel zur Verschlüsselung der Ordner und Dateien wird mit dem Passwort verschlüsselt und dann ebenfalls über *Dropbox* synchronisiert. Der Schlüssel zur Verschlüsselung der Ordner und

Dateien werden mit dem Passwort verschlüsselt und dann ebenfalls über *Dropbox* synchronisiert.

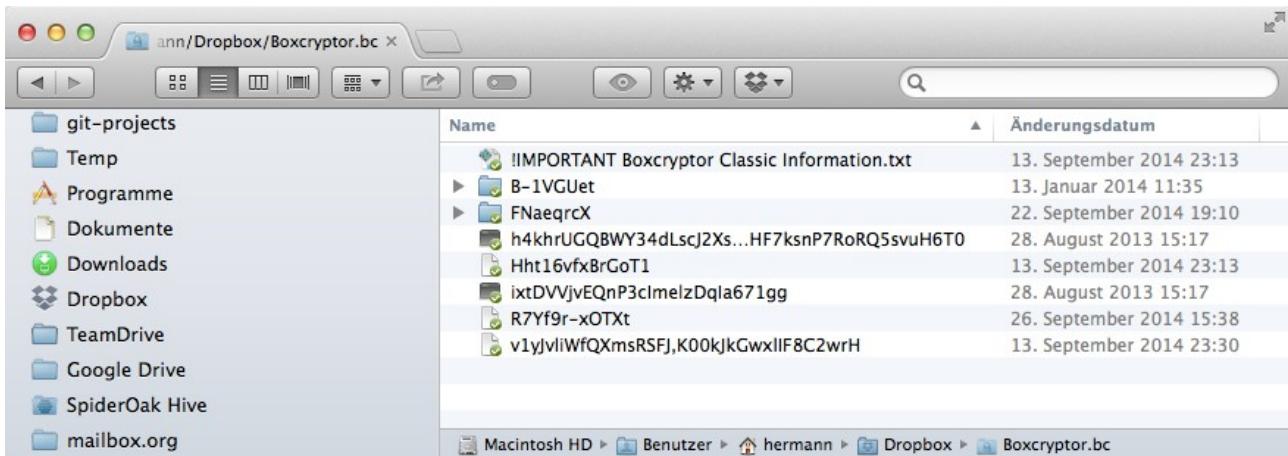


Abbildung 43: Boxcryptor: Der verschlüsselte Inhalt des verschlüsselten Ordners Boxcryptor.bc



Abbildung 44: Boxcryptor: Der entschlüsselte Inhalt des virtuellen Boxcryptor-Laufwerks

Unter Android sieht man in der *Dropbox*-App nur die synchronisierten, verschlüsselten Inhalte (siehe Abb. 45, links). Um diese zu entschlüsseln, ist die App *Boxcryptor Classic* aus dem Google Play Store zu installieren (<https://play.google.com/store/apps/details?id=com.boxcryptor.android>). Nach der Installation öffnet man die App, wählt wieder den *Dropbox* als Synchronisationsdienst aus und gibt das Passwort ein, das man bereits auf dem PC vergeben hat. Nun kann die App mit dem Passwort den verschlüsselten Schlüssel, der ebenfalls über die *Dropbox* synchronisiert wurde, entschlüsseln. Mit dem Schlüssel werden die Ordner und Dateien entschlüsselt und sichtbar gemacht. In der *Boxcryptor Classic* App stehen diese unverschlüsselt zur Verfügung und können uneingeschränkt betrachtet und bearbeitet werden (siehe Abb. 45, rechts).

Mir persönlich kommt diese Lösung sehr entgegen. So kann ich einerseits die schutzwürdigen Daten in *Boxcryptor* einsperren. Für die übrigen Daten, die ich nicht für schutzwürdig erachte, steht mir dennoch der gesamte Komfort des *Dropbox* Synchronisationsdienstes zur Verfügung. Welche Daten als schutzwürdig anzusehen sind, mag jeder selbst entscheiden.

Auch andere Kombinationen sind denkbar: für die verschlüsselten Daten *Boxcryptor* und *Google Drive* oder einen anderen Cloud-Dienst zu nutzen (*Google Drive* bietet 15 GB Freispeicher, steht allerdings nicht für Linux zur Verfügung, es sei denn man verwendet den Sync-Client von *insync*). Zusätzlich kann man *Dropbox* für die Daten verwenden, die man nicht verschlüsseln will (*Dropbox* bietet nur 2 GB Freispeicher, der sich jedoch durch Empfehlungen erhöhen lässt). Auch andere Synchronisationsdienste wie *Bitcasa* und/oder *Mega* lassen sich neben *Dropbox* und *Google Drive* parallel nutzen.

Auch an dieser Stelle möchte ich nochmals betonen, dass auch *Boxcryptor* keine Open Source Software ist. So muss man auch dieser Software vertrauen, dass sie korrekt verschlüsselt und keine Hintertüren enthält. Überprüfen lässt es sich nicht.

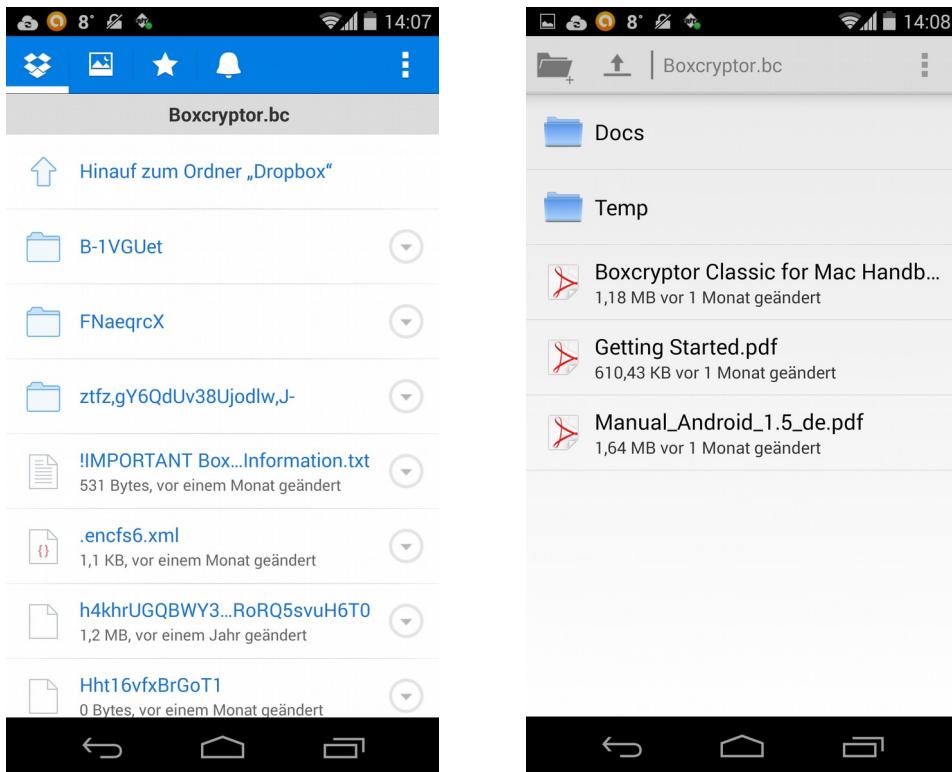


Abbildung 45: Daten in der Dropbox-App verschlüsselt, in der Boxcryptor-App entschlüsselt

## 13.4 Zusammenfassung

In diesem Kapitel machen wir uns das bei der Beschäftigung mit sicherer und verschlüsselter Email gewonnene Wissen zu Nutze und wenden auf einige andere Dienste an, die gerade beim Privatnutzer häufig in Gebrauch sind.

Zunächst werden ein paar Konzepte erläutert.

Bei **Zero Knowledge** kennt der Provider meine Daten nicht, da sie verschlüsselt sind. Zero Knowledge ist ein begriffliches Äquivalent zur Ende-zu-Ende-Verschlüsselung.

Bei **Open Source** wird die Software öffentlich verfügbar gemacht, sodass sie für jeden einsehbar und (mit dem erforderlichen fachlichen Know-how) überprüfbar ist. Bei Closed Source ist diese Überprüfbarkeit von vorn herein nicht gegeben. Man muss dem Softwareanbieter vertrauen, dass sie genau die Funktionen bietet, die der Anbieter angibt und die behaupteten Qualitätskriterien erfüllt.

Außerdem habe ich erläutert wie die **Synchronisationsfunktion** über die **Cloud** funktioniert und einige Anwendungsszenarien gezeigt, bei denen diese zur Anwendung kommt (Cloud-Speicher, Kalendersynchronisation etc.).

Nach der Erläuterung dieser Konzepte habe ich sichere Alternativen für einige häufig genutzte Dienste gezeigt, für SMS, Groupware-Dienste und für Cloud Storage.

SMS wird heute häufig durch sog. Instant Messenger ersetzt, deren prominentester Vertreter *WhatsApp* ist. Während SMS über das Sprachnetz übertragen werden, werden beim Instant

Messaging die Kurznachrichten über das Internet übertragen. Dabei laufen alle Nachrichten vom Absender zum Server des Providers und von dort zum Empfänger. Der Empfänger wird wie bei SMS über seine Telefonnummer adressiert. Für die Kommunikation müssen beide Partner bei dem Dienst registriert sein.

Bei der Übertragung werden die Nachrichten transport-verschlüsselt. Auf dem Server des Providers werden sie jedoch unverschlüsselt und damit ungeschützt zwischengespeichert. Durch die Verschlüsselung der Nachrichten kann man diese vor dem Zugriff durch Provider, Hacker oder Geheimdienste schützen (Zero Knowledge). Verschlüsseltes Instant Messaging bieten die Android-Apps *Threema*, *TextSecure* und *SIMSmee*. Ich habe *TextSecure* vorgestellt und gezeigt, wie man die App einrichtet und nutzt.

Für die Groupware-Dienste (Adressbuch, Kalender, Aufgabenlisten) in der Cloud ist mir kein Angebot bekannt, das Zero Knowledge unterstützt. Um so wichtiger ist es hier, einen vertrauenswürdigen Provider für diese Dienste zu haben. In der Regel ist dies der Email-Provider, denn fast alle Email-Provider bieten auch die gängigen Groupware-Dienste an.

Der Zugriff auf den zentralen Dienst läuft in der Regel über die Protokolle **CardDAV** und **CalDAV**. Zur Nutzung innerhalb von *Thunderbird* benötigt man die Add-ons *Sogo Connector* (ein Protokolltreiber für CardDAV und CalDAV) und *Lightning* (das die Funktionen für die Anzeige und Bearbeitung von Terminkalendern und Aufgabenlisten in *Thunderbird* bereitstellt). Unter Android habe ich die Verwendung von CardDAV-Sync und CalDAV-Sync als Protokolltreiber gezeigt. Zur Anzeige und Bearbeitung der Daten lässt sich grundsätzlich App für Kontakte, Kalender bzw. Aufgabenlisten verwenden. Zum Zugriff auf die beim Provider gespeicherten Daten mussten in den betreffenden Anwendungen nur die Zugangsdaten (CardDAV/CalDAV-URL oder -Server, Benutzername und Passwort) konfiguriert werden.

Schließlich ging es in diesem Kapitel um die Cloud-Speicher-Dienste (oder Cloud Storage) à la *Dropbox*. Ihre wichtigste Eigenschaft ist die **Synchronisation von Dateien und Ordnern**. Am Beispiel von *Dropbox* habe ich erläutert, wie diese Dienste grundsätzlich funktionieren. Die Dateien und Ordner werden über den *Dropbox*-Server abgeglichen. Sind mehrere Clients über denselben Account mit dem Server verbunden, synchronisieren sich die Clients so auch untereinander.

Allerdings werden die Daten nur transport-verschlüsselt über das Internet übertragen. Auf den Servern der Anbieter liegen die Daten unverschlüsselt. Mancher Anbieter speichert sie verschlüsselt, jedoch bleibt die Schlüsselgewalt beim Anbieter.

Einige wenige Anbieter werben auch mit dem Zero Knowledge Prinzip, bzw. Ende-zu-Ende-Verschlüsselung (*Bitcasa*, *Mega*, *SpiderOak*, *TeamDrive*). Dabei bleibt die Schlüsselgewalt ausschließlich auf dem Client, sodass die Daten auf den Servern nicht entschlüsselt werden können. Der Werbeaussage der Anbieter muss man allerdings Vertrauen schenken. Nur *SpiderOak* hat seine Software (teilweise) offengelegt, sodass eine Vereifizierung der Werbeaussage möglich ist. Edward Snowden hat eine ausdrückliche Empfehlung für die Nutzung von *SpiderOak* als *Dropbox*-Alternative gegeben. Die komfortableren Dienste sind *Bitcasa* und *Mega* mit respektablen Freispeicherangeboten von 6 GB (*Bitcasa*) und 50 GB (*Mega*).

Eine alternative Lösung zu einem Zero Knowledge Anbieter ist es, einen unsicheren Synchronisationsdienst (z.B. *Dropbox*) mit der Verschlüsselungssoftware *Boxcryptor* zu kombinieren. Dabei wird ein Unterordner (*Boxcryptor.bc*) des *Dropbox*-Ordners mit *Boxcryptor* verschlüsselt. Die entschlüsselten Daten des Unterordners werden als virtuelles Laufwerk bereitgestellt. Zusätzlich wird der Schlüssel mit dem *Boxcryptor*-Passwort verschlüsselt. Die verschlüsselten Ordner und Dateien und der verschlüsselte Schlüssel werden über den *Dropbox*-

Server mit anderen Clients auf anderen Geräten synchronisiert. Durch die Eingabe des Passworts wird auf jedem der Zugriff auf den verschlüsselten Schlüssel möglich, sodass die Ordner und Dateien so auch überall entschlüsselt vorliegen. Auf diese Weise lässt sich auch mit jedem unsicheren Anbieter Ende-zu-Ende-Verschlüsselung realisieren.

### 13.5 Links zu diesem Kapitel

- Instant Messaging bei Wikipedia: [http://de.wikipedia.org/wiki/Instant\\_Messaging](http://de.wikipedia.org/wiki/Instant_Messaging) und [http://en.wikipedia.org/wiki/Instant\\_Messaging](http://en.wikipedia.org/wiki/Instant_Messaging)
- WhatsApp bei Wikipedia: <http://de.wikipedia.org/wiki/WhatsApp> und <http://en.wikipedia.org/wiki/WhatsApp>
- Alternativen zu WhatsApp:  
[http://de.wikipedia.org/wiki/Liste\\_von\\_mobilien\\_Instant-Messengern](http://de.wikipedia.org/wiki/Liste_von_mobilien_Instant-Messengern)
- Verschlüsselnder Messenger Threema: <https://threema.ch/de> und <https://play.google.com/store/apps/details?id=ch.threema.app>
- Verschlüsselnder Messenger TextSecure: <https://whispersystems.org/> und <https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms>
- Verschlüsselnder Messenger SIMSme: <http://sims.me/> und <https://play.google.com/store/apps/details?id=dev.de.dpag.simsme>
- Hilfe zu TextSecure und RedPhone im Support Center von Open Whispersystems: <http://support.whispersystems.org/>
- Telefonie-App RedPhone für abhörsichere Telefonate im Google Play Store: <https://play.google.com/store/apps/details?id=org.thoughtcrime.redphone>
- WebDAV bei Wikipedia: <http://en.wikipedia.org/wiki/WebDAV>
- CardDAV: <http://en.wikipedia.org/wiki/CardDAV> und <http://carddav.calconnect.org/>
- CalDAV: <http://en.wikipedia.org/wiki/CalDAV> und <http://caldav.calconnect.org/>
- CardDAV-Sync beta von Marten Gadja im Google Play Store: <https://play.google.com/store/apps/details?id=org.dmfs.carddav.Sync>
- CalDAV-Sync von Marten Gadja im Google Play Store: <https://play.google.com/store/apps/details?id=org.dmfs.caldav.lib>
- Tasks von Marten Gadja im Google Play Store: <https://play.google.com/store/apps/details?id=org.dmfs.tasks>
- aCalendar von Tapir Apps GmbH im Google Play Store: <https://play.google.com/store/apps/details?id=org.withouthat.acalendar>
- Liste der CardDAV-Clients: <http://carddav.calconnect.org/implementations/clients.html>
- Liste der CalDAV-Clients: <http://caldav.calconnect.org/implementations/clients.html>
- Vergleich der Cloud Storage Anbieter bei Wikipedia: [http://en.wikipedia.org/wiki/Comparison\\_of\\_file\\_synchronization\\_software](http://en.wikipedia.org/wiki/Comparison_of_file_synchronization_software) und [http://en.wikipedia.org/wiki/Comparison\\_of\\_file\\_hosting\\_services](http://en.wikipedia.org/wiki/Comparison_of_file_hosting_services)
- Dropbox-Website: <https://www.dropbox.com/>

- *Dropbox*-App im Google Play Store:  
<https://play.google.com/store/apps/details?id=com.dropbox.android>
- *Dropbox* bei Wikipedia: <http://de.wikipedia.org/wiki/Dropbox> und  
[http://en.wikipedia.org/wiki/Dropbox\\_\(service\)](http://en.wikipedia.org/wiki/Dropbox_(service))
- *insync* Linux Sync-Client für *Google Drive*: <https://www.insynchq.com/>
- Verschlüsselungsprogramm *Boxcryptor*: <https://www.boxcryptor.com/> und  
<http://de.wikipedia.org/wiki/BoxCryptor>
- *Boxcryptor 2*: Preise und unterstützte Betriebssysteme:  
<https://www.boxcryptor.com/de/preise> und  
<https://www.boxcryptor.com/de/download>
- *Boxcryptor Classic*: <https://www.boxcryptor.com/de/classic>
- Von *Boxcryptor* unterstützte Cloud Storage Provider:  
<https://www.boxcryptor.com/de/provider>
- *Boxcryptor Classic* im Google Play Store:  
<https://play.google.com/store/apps/details?id=com.boxcryptor.android>

## 14 Verschlüsselte Übertragungskanäle – Wie sicher sind sie wirklich?

Dieses Kapitel liefert einen technisch etwas genaueren Blick auf Inhalte, die in Kapitel 3.2 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 bereits 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 kommuniziere mit dem Server und kommuniziert tatsächlich mit dem MITM (Man in the Middle). Der Server „denkt“ auch, er kommuniziere 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, das vom Client und von Server als echt anerkannt wird.

Für „gut gemachte“ Verschlüsselung gibt es einige Qualitäts-Kriterien, an denen man auch die Mail-Provider messen kann (siehe Kap. 3.3). Bei Einhaltung dieser Qualitätskriterien ist das Kapern von TLS-Verbindungen erheblich schwieriger.

- Unterstützung aller TLS-Versionen, auch der neuesten Version 1.2. SSL darf nicht mehr unterstützt werden (siehe Kap. 14.1). Wenn SSL auf dem Mail-Server abgeschaltet wurde, dann ist dieser auch nicht mehr durch den Poodle-Angriff (siehe Kap. 14.1.1) verwundbar.
- Unterstützung von PFS (siehe Kap. 14.2)
- Unterstützung von HSTS für die Webmail-Schnittstelle (siehe Kap. 14.3)
- Unterstützung von DANE (siehe Kap. 14.4)
- Nicht verwundbar durch die Heartbleed-Attacke

Ich werde diese eher technischen Kriterien in den nachfolgenden Unterkapiteln 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. 2.8). 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. 14.4) 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 erwerben.

## 14.1 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). Diese Version sollte unterstützt werden. 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 Webbrowser (HTTP-Client) und einem Web-Server (HTTP-Server) (siehe Kap. 2.4.1). 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 (**File Transfer Protokoll**) ist das Protokoll zur unverschlüsselten Dateiübertragung (siehe Kap. 2.4.1). FTPS (**FTP Secure**) 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. 2.4.1). 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. 14.4).

Weitere Informationen zu SSL und TLS unter:

[http://de.wikipedia.org/wiki/Transport\\_Layer\\_Security](http://de.wikipedia.org/wiki/Transport_Layer_Security)

Die Protokolle HTTP, FTP, SMTP und IMAP sind im Kapitel 2.4.1 im Glossar kurz erläutert. Detailliertere Informationen finden sich auf den deutschen und englischen Wikipedia-Seiten.

### 14.1.1 Poodle – Sicherheitslücke in SSLv3

Mitte Oktober 2014 wurde von Google-Sicherheitsexperten eine Sicherheitslücke im SSL-Protokoll der Version 3 unter dem Namen **Poodle** veröffentlicht.

Beim Verbindungsaufbau handeln Client und Server die zu verwendende SSL- oder TLS-Protokollversion miteinander aus. Sie einigen sich auf den kleinsten gemeinsamen Nenner. Ein Client der nur SSLv3 unterstützt, kann so den Server, der TLS 1.2 unterstützen würde, zwingen SSLv3 zu verwenden. Umgekehrt funktioniert dies genau so: Ein Server, der nur SSLv3 unterstützt, kann den Client auf diese Protokollversion herunterzwingen. Dieses Verfahren der Einigung auf den kleinsten gemeinsamen Nenner nennt man **Downgrading**. Verstehen beide eine höhere Protokollversion (z.B. TLS 1.2), dann einigen sie sich beim Verbindungsaufbau auf diese.

Beim *Poodle*-Angriff wird dieses Downgrading ausgenutzt. Dabei schaltet sich der Angreifer als MITM (Man in the Middle) schon beim Verbindungsaufbau zwischen den Client und den Server. Der Server hält den MITM für den Client und der Client hält den MITM für den Server. Bei der Aushandlung der Protokollversion zwingt der MITM Client und Server auf das unsichere SSLv3 herunter. Über das veraltete SSLv3 lassen sich dann beide Seiten relativ leicht angreifen.

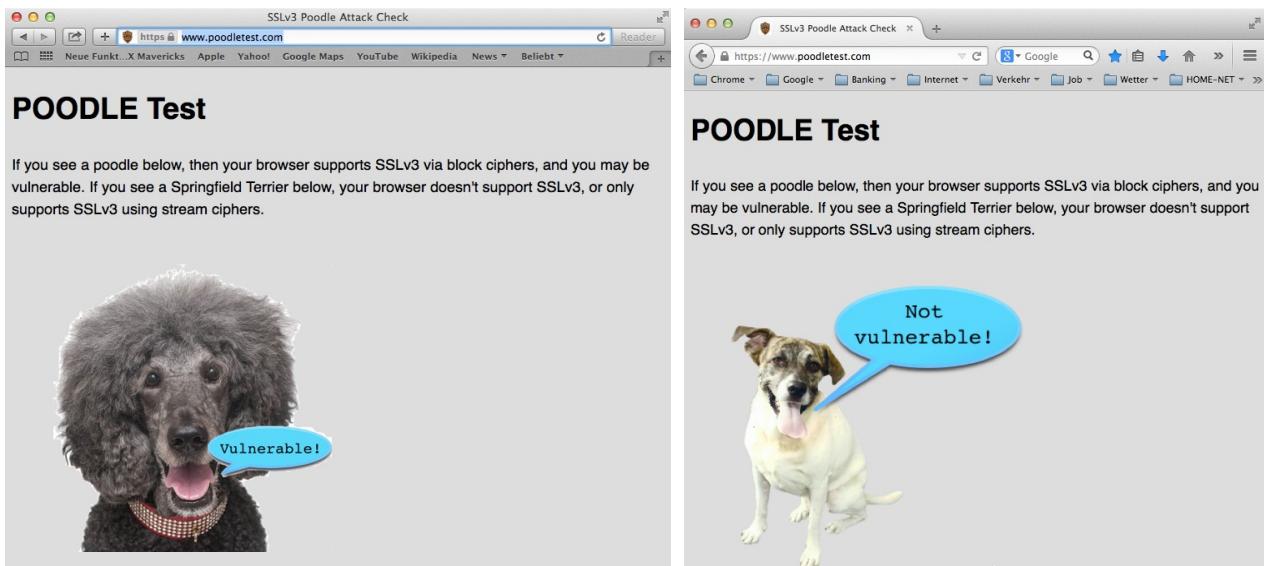


Abbildung 46: Poodle-Test mit einem verwundbaren (links) und einem nicht verwundbaren Browser (rechts)

Als Gegenmaßnahme muss SSLv3 clientseitig und serverseitig abgeschaltet werden. Dies ist heute auch gefahrlos und ohne Funktionseinbußen möglich. (Allein der uralte Browser Internet Explorer 6 versteht noch kein TLS und ist auf SSL angewiesen. Wer allerdings diesen Methusalem der Webbrowser heute noch verwendet, hat sowieso ein viel größeres Sicherheitsproblem.)

- Für die aktuellen Browser (Web-Clients) gibt es Möglichkeiten SSLv3 abzuschalten (siehe nachstehende Links). Der technisch weniger Versierte kann jedoch auf die Browser-Updates der kommenden Monate warten. In den neuen Browser-Versionen wird SSLv3 abgeschaltet sein. Dies haben einige Hersteller bereits angekündigt.
- Mail-Clients sind nicht ganz so gefährdet wie die Browser, da hier die Ausführung von Skripts in der Regel abgeschaltet ist. Doch auch bei diesen werden die Updates der kommenden Monate wohl die Unterstützung von SSLv3 deaktivieren.
- Für die Abschaltung von SSLv3 auf den Web-Servern sind die Betreiber der jeweiligen Web-Server zuständig.
- Für die Abschaltung von SSLv3 auf den Mail-Servern sind die Betreiber der jeweiligen Mail-Server zuständig. Wie schnell diese auf den Fehler reagieren, kann durchaus ein Bewertungskriterium für die Mail-Provider sein. Einige wenige Mail-Provider haben schnell reagiert und die SSL-Unterstützung bereits aufgegeben. Deren Server sind damit über den *Poodle*-Angriff auch nicht mehr verwundbar.

Ob der eigene Browser für *Poodle* anfällig ist, kann man gefahrlos auf <https://www.poodletest.com> testen. Ist er anfällig für *Poodle*, so erhält man einen Pudel mit der Sprechblase „Vulnerable!“. Ist er nicht anfällig, so erhält man einen Terrier mit der Sprechblase „Not Vulnerable!“ (siehe Abb. 46).

Heise Online hat über *Poodle* in mehreren Artikel berichtet. Hier sind die Links für diejenigen, die

sich genauer mit *Poodle* beschäftigen wollen:

- <http://www.heise.de/newsticker/meldung/Poodle-Experten-warnten-vor-Angriff-auf-Internet-Verschlüsselung-2424122.html>
- <http://www.heise.de/security/artikel/Poodle-So-funktioniert-der-Angriff-auf-die-Verschlüsselung-2425250.html>
- <http://www.heise.de/security/meldung/So-wehren-Sie-Poodle-Angriffe-ab-2424327.html>
- <http://www.heise.de/security/meldung/Angriff-auf-Verschlüsselung-Reaktionen-auf-die-Poodle-Luecke-2425244.html>
- <http://www.heise.de/newsticker/meldung/SSL-Verschlüsselung-Noch-viel-Arbeit-fuer-Mail-Provider-und-Banken-2429414.html>

## 14.2 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. **PFS verhindert die nachträgliche Entschlüsselung.**

Bei PFS werden für jede Kommunikationssitzung temporäre Schlüssel, 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. Somit ist die Aufzeichnung einer verschlüsselten Kommunikationssitzung in der Hoffnung auf nachträgliche Entschlüsselung nutzlos.

Weitere Infos zu PFS: [http://de.wikipedia.org/wiki/Perfect\\_Forum\\_Secrecy](http://de.wikipedia.org/wiki/Perfect_Forum_Secrecy)

## 14.3 HSTS (HTTP Strict Transport Security)

Dieses Protokoll spielt nur bei der Verwendung der Webmail-Schnittstelle, also beim Zugriff auf die Mails mit dem Webbrowser (siehe Kap. 4.5), eine Rolle. HSTS erzwingt den Zugriff auf den Web-Server des Mail-Providers mit HTTPS auch dann, 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>

## 14.4 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 Webbrowser 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 Googlemail-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 Googlemail-Server.

Beim Einsatz von DANE 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*, *I&I*, *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 (SMTPTS) ausgerichtet. DANE ist allgemeiner konzipiert. Es kann die TLS-verschlüsselte Übertragung für alle Protokolle (SMTPTS, IMAPS,

HTTPS, FTPS und alle anderen Protokolle, die TLS-Verschlüsselung verwenden) 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](http://www.heise.de/artikel-archiv/ct/2014/11/194_Geleitschutz)
- Deutscher Wikipedia-Eintrag zu DNS: [http://de.wikipedia.org/wiki/Domain\\_Name\\_System](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/Verschlüsselter-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>

## 14.5 Heartbleed

*Heartbleed* ist der Name eines schweren Fehlers in der *OpenSSL*-Bibliothek, der im Frühjahr 2014 entdeckt wurde und in der Fachpresse recht viel Wirbel verursacht hat.

Viele Client- und Server-Programme (auch Web-Clients und Web-Server sowie Mail-Clients und Mail-Server) verwenden zur Implementierung von SSL und die TLS die Software-Bibliothek *OpenSSL*. Ist in einer Software-Bibliothek ein Fehler, so betrifft er auch alle Programme, die diese Bibliothek verwenden. Da *OpenSSL* in sehr vielen Programmen für die Implementierung der Transportverschlüsselung verwendet wird, hat die schon lange schlummernde und nun entdeckte Sicherheitslücke mit einem Schlag viele Client- und Server-Programme angreifbar gemacht und so schnell große und traurige Berühmtheit erlangt.

Mittlerweile (Oktober 2014) ist der Fehler behoben und die Sicherheitslücke in der Bibliothek damit geschlossen. Entscheidend ist, dass die alte Version der *OpenSSL*-Bibliothek überall, wo sie verwendet wird, zügig durch die neue Version ersetzt wird.

Die Hersteller der Browser und Mail-Clients haben ihren Job weitgehend erledigt und neue Programmversionen bereitgestellt. Die Betreiber der Server müssen dies ebenfalls zügig tun. Ca. ein halbes Jahr nach Bekanntwerden von *Heartbleed*, sollte heute im Oktober 2014 kein Server mehr durch *Heartbleed* angreifbar sein. Dies gilt selbstverständlich für die Betreiber von Web-Servern genau so wie für die Betreiber von Mail-Servern, also für die Mail-Provider. Auch dies kann Bewertungskriterium für Mail-Provider sein. Ist ein Mail-Server heute noch für *Heartbleed* anfällig, so zeigt dies ein eher schwach entwickeltes Sicherheitsbewusstsein des betreffenden Providers.

## 14.6 Die Qualität der Transport-Verschlüsselung der Mail-Provider prüfen

Zur Anfälligkeit für *Poodle* und zur PFS-Unterstützung einiger bekannter Mail-Provider gibt es eine kleine Übersicht über die in Deutschland gängigen Mail-Provider auf Heise Online vom Oktober:  
<http://www.heise.de/newsticker/meldung/SSL-Verschlüsselung-Noch-viel-Arbeit-fuer-Mail-Provider-und-Banken-2429414.html>

Diese Übersicht zeigt den Status für 14 Mail-Provider im Oktober 2014. Aus der Tabelle lässt sich ablesen, welche Mail-Provider das veraltete Protokoll SSL bereits vollständig deaktiviert haben und damit auch nicht mehr durch den *Poodle*-Bug angreifbar sind und welche PFS bereits unterstützen.

Außerdem kann man auch selbst eigene Prüfungen vornehmen. Außerdem kann man an einigen URLs eigene Prüfungen vornehmen.

### 14.6.1 starttls.info testet Qualität der Transportverschlüsselung

Dazu gibt man im Browser die URL: ein. Auf der Seite <https://starttls.info> gibt es ein Eingabefeld, in das man eine Email-Adresse (z.B. *max.mayer@gmx.de*) eingeben kann. Es genügt jedoch auch die Email-Domain; dazu lässt man den *max.mayer* weg und man beschränkt sich auf den Teil nach dem @-Symbol, also *gmx.de*.

Für den Test mit meiner Email-Domain *secure.mailbox.org* erhielt ich im Oktober 2014 das Ergebnis als Übersicht in Abbildung 47, in der alle Server dieser Domain mit einem Score aufgelistet sind. Die Detailinformation zu jedem Server erhält man, wenn man den Pfeil rechts aufklappt (siehe Abb. 48). Dort lässt sich auch gut erkennen, ob der betreffende Provider noch das SSL-Protokoll unterstützt (siehe Kap 14.1.1).

Mail server	Result
<a href="#">mxtls1.mailbox.org</a>	Grade: A (94.8%)
<a href="#">mxtls2.mailbox.org</a>	Grade: A (94.8%)

Click the score for details. [Test another!](#)

This site is a beta. | Read [about this](#). | Check the [stats](#). Developed by [Einar Otto Stangvik](#).

Abbildung 47: *starttls.info*: Abfrage von *secure.mailbox.org* (Übersicht)

Es bietet sich an, die Email-Domains verschiedener Provider zu vergleichen. Ich habe diesen Vergleich für einige bekannte Provider und auch für meine Favoriten aus Kapitel 3.3.2 durchgeführt. Die Ergebnisse meiner Tests vom 18.07.2014 finden sich in Kapitel 14.6.4 in Tabelle 1.

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 wechselt. Er kann auch als

Auswahlkriterium beim Provider-Vergleich (siehe Kap. 3.3.1) herangezogen werden.

Mail server	Result
<a href="#">mx1.mailbox.org</a>	Grade: A (94.8%)
<a href="#">mx2.mailbox.org</a>	Grade: A (94.8%)

Click the score for details. [Test another!](#)

This site is a beta. | Read [about this](#). | Check the [stats](#). Developed by [Einar Otto Stangvik](#).

Abbildung 48: [starttls.info](https://starttls.info/): Abfrage von [secure.mailbox.org](https://secure.mailbox.org) (Detail-Ansicht für ersten Server)

Am 24.10.2014 habe ich dieselben Provider nochmals getestet und dabei das Ergebnis in Kapitel 14.6.4 in Tabelle 2 erhalten.

## 14.6.2 [tlsa.info](https://www.tlsa.info/) testet DANE-Unterstützung

Auch die DANE-Unterstützung lässt sich an der Web-Adresse <https://www.tlsa.info/> überprüfen. Ebenso wie unter <https://starttls.info> gibt man eine Mail-Domain in ein Suchfeld ein und erhält dann ein positives oder negatives Testergebnis. Das Ergebnis für [mailbox.org](https://mailbox.org) zeigt Abbildung 49.

Am 24.10.2014 habe ich auch diesen Test für die Provider-Liste durchgeführt und das Ergebnis in Tabelle 2 in Kap. 14.6.4 hinzugefügt.

The screenshot shows the results of a DNSSEC and DANE/TLSA test for the domain `mailbox.org`. The top navigation bar includes links for 'Statistics' and a search icon. The main title is 'DNSSEC and DANE/TLSA Test Results for domain `mailbox.org`'. Below this, a message states 'Test results last updated on Fri, Oct 24th 2014, 19:18. Refresh'. Two green buttons indicate 'Full DNSSEC Support' and 'Full DANE/TLSA Support'. The DNSSEC section shows that all MX records are secured. The DANE/TLSA section shows two entries for each MX record, both marked as 'Yes'. A 'Check another domain' section allows entering 'gmail.com' and has a 'Start Test' button. There is also a checkbox for 'Don't include in this site's statistics'.

MX Server	MX Priority	IP Adresses	DNSSEC	DANE/TLSA
<code>mx1.mailbox.org</code>	10	80.241.60.212	Yes	Yes 3 1 1 b1d1dabb6f2ab70502e39bb9df9367e10ca62cd8bcaa6cf038e2cb4ee492296 Yes 3 1 1 b80203715536f36ed6516db09668d63d490214c81b8c8384a74f952b09274479
<code>mx2.mailbox.org</code>	10	80.241.60.215	Yes	Yes 3 1 1 b1d1dabb6f2ab70502e39bb9df9367e10ca62cd8bcaa6cf038e2cb4ee492296 Yes 3 1 1 b80203715536f36ed6516db09668d63d490214c81b8c8384a74f952b09274479
<code>mx3.mailbox.org</code>	20	80.241.60.216	Yes	Yes 3 1 1 b1d1dabb6f2ab70502e39bb9df9367e10ca62cd8bcaa6cf038e2cb4ee492296 Yes 3 1 1 b80203715536f36ed6516db09668d63d490214c81b8c8384a74f952b09274479

Abbildung 49: `tlsa.info`: DANE-Unterstützung testen für `mailbox.org`

### 14.6.3 Weitere Tests unter `de.ssl-tools.net`

Eine weitere URL zum Test von Email-Providern ist: <https://de.ssl-tools.net/mailservers>.

Auch die Tests auf dieser Website sind in die Ergebnisse vom 24.10.2014 in Kapitel 14.6.4, Tabelle 2 eingeflossen.

### 14.6.4 Testergebnisse

Am 18.07.2014 habe ich diverse Provider auf `starttls.info` getestet (siehe Kap. 14.6.1). Die Ergebnisse finden sich in Tabelle 1.

Email-Domain	Provider	Score (starttls.info)
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 %

**Tabelle 1**, Testergebnisse vom 18.07.2014

Am 24.10.2014 habe ich dieselben Provider nochmals getestet unter *starttls.info* (siehe Kap. 14.6.1), *tlsa.info* (siehe Kap. 14.6.2) und unter *de.ssl-tools.net* (siehe Kap. 14.6.3). Die Ergebnisse finden sich in Tabelle 2.

Beim zweiten Test bei *starttls.info* haben *icloud.com* und *yahoo.com* ihren Score leicht verbessert. *telekom.de* ist in diesem Vergleich das Schlusslicht. Einen großen Sprung nach oben haben *mykolab.com* und *jpberlin.de* gemacht. Es bewegt sich etwas, aber einige Provider müssen ihre Hausaufgaben noch machen, um eine qualitativ hochwertige Transport-Verschlüsselung bereitstellen zu können.

Email-Domain	Provider	Heartbleed-Verwundbarkeit	Poodle-Verwundbarkeit	PFS-Unterstützung	DANE-Unterstützung (tlsa.info)	Score (starttls.info)
gmail.com	Google	nein	ja	ja	nein	90,6 %
outlook.com	Microsoft	nein	ja	ja	nein	90,6 %
icloud.com	Apple	nein	ja	ja	nein	83,2 %
yahoo.com	Yahoo	nein	ja	ja	nein	90,6 %
web.de	WEB.DE	nein	ja	ja	nein	90,6 %
gmx.net	GMX	nein	ja	ja	nein	90,6 %
freenet.de	Freenet	nein	ja	ja	nein	Error: Could not connect
telekom.de	Telekom	nein	ja	ja	nein	48,8 %
mykolab.com	MyKolab	nein	nein	ja	nein	92,0 %
posteo.de	Posteo	nein	ja	ja		Error: Connection rejected
jpberlin.de	JPBerlin	nein	nein	ja	teilweise	71,8 %
mailbox.org	mailbox.org	nein	nein	ja	ja	83,4 %
secure.mailbox.org	mailbox.org	nein	nein	ja	ja	94,8 %
mail.de	mail.de	nein	ja	ja	ja	90,6 %

**Tabelle 2**, Testergebnisse vom 24.010.2014

Die Probleme mit Heartbleed scheinen mittlerweile bei den getesteten Providern behoben zu sein. Die PFS-Unterstützung ist bei allen gegeben. Auf Poodle haben in den zwei Wochen seit Bekanntwerden der Lücke nur einige Pioniere reagiert und SSLv3 schon abgeschaltet. Auch die DANE-Unterstützung ist noch die Sache von einigen Pionieren und es sind weitgehend dieselben, die auch schnell auf Poodle reagiert haben. Ich nehme an, dass die „roten Flecken“ in der Poodle-

Spalte bei Posteo und mail.de im Laufe des November 2014 verschwinden werden. Die großen Provider zeigen sich durch die Bank tendenziell etwas träger in ihrer Bereitschaft, auf Sicherheitslücken zu reagieren und technische Innovationen aufzugreifen.

## 14.7 Zusammenfassung

In diesem Kapitel haben wir transport-verschlüsselte Übertragungskanäle technisch etwas genauer unter die Lupe genommen. Die Qualität der Transport-Verschlüsselung kann durchaus sehr unterschiedlich sein. Bei der Kommunikation zwischen Client und Server ist der Betreiber des Servers in erster Linie für eine qualitativ hochwertige Verschlüsselung zuständig – beim Mailverkehr also die Email-Provider. Deshalb kann man diese unter Anderem auch nach der Qualität der bereitgestellten Transport-Verschlüsselung bewerten.

Die Qualität der Transport-Verschlüsselung lässt sich an einigen Merkmalen festmachen.

Es sollten alle TLS-Versionen einschließlich der neuesten Version **TLS 1.2** unterstützt werden. **SSL** darf jedoch **nicht mehr unterstützt** werden, da dies durch das sog. Downgrading den *Poodle*-Angriff ermöglicht.

Die Unterstützung von **PFS** ist erforderlich, um die nachträgliche Entschlüsselung von früher mitgeschnittener Kommunikation zu verhindern. Bei PFS wird ein temporärer Sitzungsschlüssel erzeugt, der nach dem Ende der Kommunikationssitzung verworfen wird.

**HSTS** betrifft nur die Webmail-Schnittstelle. Dieses Protokoll erzwingt die Verwendung von HTTPS auch dann, wenn der unvorsichtige Nutzer die Webmail über HTTP abzurufen versucht.

Transport-Verschlüsselung basiert auf Zertifikaten, die von den sog. CAs (Certificate Authorities) signiert wurden. Baut ein Client zu einem Server eine verschlüsselte Verbindung auf, so vertraut er dem Server, wenn dessen öffentlicher Schlüssel von einer bekannten CA zertifiziert (signiert) wurde. Da sich jedoch gezeigt hat, dass die CAs grundsätzlich korrumptierbar sind, lassen sich die Schlüssel von Servern fälschen. Der Server erhält gewissermaßen eine gefälschten Ausweis. Der Client hält den Ausweis für echt und baut deshalb eine Verbindung zum falschen Server auf – im guten Glauben, dass es der richtige sei.

Diesem Problem versucht das **DANE**-Protokoll beizukommen, indem es die unumschränkte Macht der CAs einschränkt. Die Zertifikate werden über das sichere DNS Secure automatisch verteilt. DANE ist noch ein recht neuer Standard und wird bislang noch von sehr wenigen Email-Providern unterstützt.

**Heartbleed** ist ein Bug in der weit verbreiteten Software-Bibliothek *OpenSSL*, der im Frühjahr 2014 entdeckt wurde und mittlerweile geschlossen. Heute im Oktober 2014 sollte kein Mail-Provider mehr durch diese Sicherheitslücke angreifbar sein.

Auf verschiedenen Websites kann man die Qualität der Transportverschlüsselung unter den aufgeführten Qualitätskriterien testen. Die Ergebnisse meiner Tests habe ich in Kapitel 14.6.4 tabellarisch zusammengefasst.

## 14.8 Links zu diesem Kapitel

- Infos zu SSL und TLS unter: [http://de.wikipedia.org/wiki/Transport\\_Layer\\_Security](http://de.wikipedia.org/wiki/Transport_Layer_Security)
- Poodle-Test für den eigenen Browser: <https://www.poodletest.com>
- Links zum Poodle-Angriff bei Heise Online und Heise Security:  
<http://www.heise.de/newsticker/meldung/Poodle-Experten-warnten-vor-Angriff-auf-Internet->

[Verschlüsselung-2424122.html](#)

<http://www.heise.de/security/artikel/Poodle-So-funktioniert-der-Angriff-auf-die-Verschlüsselung-2425250.html>

<http://www.heise.de/security/meldung/So-wehren-Sie-Poodle-Angriffe-ab-2424327.html>

<http://www.heise.de/security/meldung/Angriff-auf-Verschlüsselung-Reaktionen-auf-die-Poodle-Luecke-2425244.html>

<http://www.heise.de/newsticker/meldung/SSL-Verschlüsselung-Noch-viel-Arbeit-fuer-Mail-Provider-und-Banken-2429414.html>

- Infos zu PFS: [http://de.wikipedia.org/wiki/Perfect\\_Forward\\_Secrecy](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](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/Verschlüsselter-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>
- Übersicht über die Qualität der Transport-Verschlüsselung bei den Mail-Providern im Oktober 2014: <http://www.heise.de/newsticker/meldung/SSL-Verschlüsselung-Noch-viel-Arbeit-fuer-Mail-Provider-und-Banken-2429414.html>
- Qualität der SSL/TLS-Verschlüsselung von Email-Providern testen: <https://starttls.info>
- DANE-Unterstützung von Email-Providern prüfen: <https://www.tlsa.info/>
- Email-Provider-Test: <https://de.ssl-tools.net-mailservers>

## 15 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 mit dem Android-Betriebssystem betrieben werden.
Asymmetrische Verschlüsselung	Für das Verschlüsseln und das Entschlüsseln gibt es zwei unterschiedliche, aber zusammengehörende Schlüssel – den <b>Public Key</b> und den <b>Private Key</b> . Was mit dem einen verschlüsselt wurde, kann nur mit dem anderen entschlüsselt werden.
Betriebssystem-Kernel	(oder Betriebssystem-Kern) siehe <b>Kernel</b>
CA	<b>Certificate Authority</b> : eine zugelassene Zertifizierungsstelle, die digitale Schlüssel für die verschlüsselte Kommunikation im Internet zertifiziert (beglaubigt).
CalDAV	<b>Calendaring Extensions for WebDAV</b> : Dieses Protokoll ist eine Spezialisierung des <b>WebDAV</b> -Protokolls zum Zugriff auf entfernte Kalenderdaten, also auf Termine und Aufgaben. Neben server-gespeicherten Kalendern werden auch Aufgabenlisten unterstützt.
CardDAV	<b>Card Distributed Authoring and Versioning</b> : Dieses Protokoll ist eine Spezialisierung des <b>WebDAV</b> -Protokolls zum Zugriff auf entfernte, server-gespeicherte Kontaktdaten.
Client	Ein Programm, das den von einem <b>Server</b> angebotenen Dienst nutzt. Z.B. nutzt ein <b>SMTP</b> -Client den von einem <b>SMTP</b> -Server angebotenen Mail-Versand-Dienst. Das <b>Protokoll</b> definiert die zulässigen die Nachrichten, die der Client und der Server miteinander austauschen. Das <b>SMTP</b> -Protokoll beschreibt die Nachrichten zwischen <b>SMTP</b> -Client und <b>SMTP</b> -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. Den Privatnutzern sind am ehesten die Cloud-Speicher-Dienste bekannt, die auch als <i>Online-Festplatten</i> bezeichnet werden. Der prominenteste 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 Synchronisation der Daten desselben Benutzers zwischen verschiedenen Geräten.
DANE	<b>DNS-based Authentication of Named Entities</b> : Siehe Kapitel 14.4

Begriff oder Abkürzung	Erläuterung
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	<b>Domain Name System:</b> 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: <a href="http://www.google.de">www.google.de</a> ). Mit Hilfe des DNS findet der Browser automatisch die IP-Adresse von <a href="http://www.google.de">www.google.de</a> heraus und verwendet diese, um diesen Server zu adressieren und eine Verbindung zu ihm herzustellen.
DSA	<b>Digital Signature Algorithm:</b> 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. Ein anderer, gerne verwendeter Begriff dafür ist <b>Zero Knowledge</b> .
FTP	<b>File Transfer Protocol:</b> Protokoll zur Übertragung von Dateien über das Netzwerk
GCHQ	<b>Government Communication Headquarters.</b> Britischer Geheimdienst, zuständig für Spionage im Internet.
GNU	<b>GNU is Not Unix:</b> GNU ist eine Organisation, die die Lizenzkosten-freie Verbreitung von Software propagiert.
PGP, GnuPG	<b>GNU Privacy Guard</b> ist die PGP-Implementierung von <b>GNU</b> .
HTTP	<b>Hypertext Transfer Protocol:</b> Protokoll zur Übertragung verlinkter Web-Seiten (Hypertext) zwischen vom Web-Server (HTTP-Server) zum Webbrower (HTTP-Client)
HTTPS	<b>HTTP Secure:</b> Sicheres HTTP wird verwendet, wenn Browser und Web-Server über einen verschlüsselten Transport-Kanal miteinander kommunizieren.
HTML	<b>Hypertext Markup Language:</b> HTML ist die Auszeichnungssprache für Webseiten. Neben dem eigentlichen Inhalt (dem Text) einer Webseite kann man in HTML auch deren Strukturierung (Titel, Überschriften, Absätze, Aufzählungen) Formatierung (verschiedene Schriftarten und -größen, Fett- und Kursivschrift) und Darstellung (Hintergrundfarbe, Rahmen, Schattierung) festlegen. Die Einbettung von Hyperlinks erlaubt es dem Betrachter der Seite, mit einem Mausklick auf eine andere Seite zu wechseln. Außerdem lassen sich Multimedia-Dateien (Bilder, Audios und Videos) einbetten. Auch die Einbettung von Programmen ( <b>JavaScript</b> und <b>Java-Applets</b> ) ist möglich. Diese werden ausgeführt, wenn die HTML-Seite geladen und angezeigt wird. HTML ist das Standard-Format von Webseiten. Diese werden in einem

Begriff oder Abkürzung	Erläuterung
	Browser-Fenster dargestellt. HTML kann jedoch auch als Email-Format verwendet werden. In diesem Fall wird es vom Email-Client angezeigt.
HSTS	<b>HTTP Strict Transport Security:</b> 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 14.3).
IMAP	<b>Internet Mail Access Protocol:</b> ein Protokoll zum Zugriff und Verwaltung der beim Provider gespeicherten, empfangenen und gesendeten Mails. Anders als bei <b>POP</b> bleiben die Mails beim Provider gespeichert. Dadurch ist der Zugriff mit vielen Geräten auf denselben Mail-Bestand möglich.
Implementation	Umsetzung, Realisierung, Erfüllung (eines Vertrages)
Inline-PGP	Inline-PGP ist das klassische Format für die Verschlüsselung und oder Signierung von Mails mit PGP. Dabei wird die gesamte Mail mit allen Anhängen inline (d.h. als ein zusammenhängender Block) verschlüsselt. Inline-PGP funktioniert nicht zuverlässig mit HTML-formatierten Mails. Deshalb muss man bei Inline-PGP die HTML-Formatierung für die Erstellung der Mails abschalten. Die Alternative ist das modernere Mailverschlüsselungsformat <b>PGP/MIME</b> . <b>PGP/MIME</b> kann auch HTML-formatierte Mails problemlos verschlüsseln und wieder entschlüsseln.
iOS	Betriebssystem der mobilen Geräte von Apple (iPhone, iPad und iPod).
Java	Java ist eine weit verbreitete Allzweck-Programmiersprache. Ein Java-Programm wird wie andere Programme auch auf dem System installiert. Damit ein Java-Programm auf einem System ausgeführt werden kann, muss auch die sog. Java Laufzeitumgebung (JRE = Java Runtime Environment) installiert sein. Anders als ein <b>Java-Applet</b> ist eine normale Java-Programm nicht auf einen Webbrower angewiesen.
Java-Applet	Ein Java-Applet ist ein (in der Regel kleines) <b>Java</b> -Programm, das innerhalb des Webbrowsers ausgeführt wird. Ein Java-Applet wird (anders als ein normales <b>Java</b> -Programm) nicht auf dem System installiert. Es wird in eine Webseite eingebettet und aus dem Web geladen und ausgeführt, wenn die Webseite ( <b>HTML</b> -Seite) vom Browser angezeigt wird. Ein Mail-Client wie <i>Thunderbird</i> kann ebenfalls Java-Applets ausführen, wenn das Applet in eine <b>HTML</b> -Mail eingebettet ist. Die automatische Ausführung von Java-Applets lässt sich sowohl im Browser als auch im Mail-Client deaktivieren. Dazu muss man in den Einstellungen des Mail-Client das Java-Plugin deaktivieren (siehe Kap. 4.4.2). Analog zum Mail-Client kann man das Java-Plugin auch im Browser deaktivieren, um die Ausführung von Java-Applets zu verhindern.
JavaScript	JavaScript ist eine Programmiersprache, die meist in <b>HTML</b> -Seiten oder in <b>HTML</b> -Mails eingebettet wird. Die Laufzeit-Umgebung von JavaScript-Programmen ist normalerweise der Webbrower. Doch auch ein Email-Client wie <i>Thunderbird</i> kann grundsätzlich JavaScript-Programme ausführen, die in <b>HTML</b> -Mails eingebettet sind. (Es gibt heute auch JavaScript-Programme, die

Begriff oder Abkürzung	Erläuterung
	unabhängig vom Browser oder Email-Client ablauffähig sind. Von diesen ist hier nicht die Rede.) Ähnlich wie <b>Java-Applets</b> werden JavaScript-Programme nicht auf dem System installiert sondern aus dem Web geladen. Die Ausführung von JavaScript im Browser ist in der Regel erwünscht. Würde man JavaScript im Browser deaktivieren, dann könnte man nicht mehr vernünftig im Web surfen, da sehr viele Webseiten ohne JavaScript nicht vernünftig funktionieren. Auf JavaScript im Mail-Client kann man jedoch gut verzichten. Die Ausführung von JavaScript, das in <b>HTML</b> -Mails eingebettet ist, ist gefährlich und deshalb in der Regel unerwünscht. Die Ausführung von JavaScript ist deshalb im Email-Client <i>Thunderbird</i> per Voreinstellung deaktiviert (siehe Kap. 4.4.1).
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 können z.B. auch alternative App Stores (z.B. der Cydia Store), die nicht Apples Segen haben, genutzt werden.
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-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 die Verwaltung der aktiven Programme, die 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	<b>Key Revocation Certificate:</b> Mit dem Widerrufszertifikat kann ein Schlüssel (auch ohne die Passphrase) widerrufen, d.h. für ungültig erklärt werden. Wie der private Schlüssel sollte auch das Widerrufszertifikat nicht in die Hände fremder Personen fallen.
Linux	Freies Betriebssystem für PCs, Server und mobile Geräte
Mac OS X	PC-Betriebssystem von Apple
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 <a href="mailto:joseph.mayer@web.de">joseph.mayer@web.de</a> den Mail-Alias <a href="mailto:joseph@mayer.de">joseph@mayer.de</a> einrichten, falls dieser

Begriff oder Abkürzung	Erläuterung
	Alias nicht bereits vergeben ist.
Mail-Provider	Siehe <b>Provider</b>
Malware	( <b>Malicious Software</b> ) Schädliche Software, die unerwünschte Aktivitäten auf einem Rechner ausführt. Dazu gehören Viren, Trojaner, Spähprogramme etc.
Metadaten	Daten über Daten
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	<b>Multi Purpose Internet Mail Extensions:</b> 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-Programm. 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/acs7-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 das Add-on <i>Enigmail</i> installiert, so kann <i>Thunderbird</i> das selbst erledigen und benötigt kein externes Zusatz-Programm.
NSA	National Security Agency. Amerikanischer Geheimdienst, zuständig für Spionage im Internet.
Open Source	Open Source Programme nennt man Programme, deren <b>Quelltext</b> 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 <b>Passphrase</b> bezeichnet. Für jeden privaten Schlüssel wird eine eigene Passphrase festgelegt.
Passwort-Manager	Passwort-Verwaltungsprogramm. Der heutige Internet-Benutzer hat häufig

Begriff oder Abkürzung	Erläuterung
oder Passwort-Safe	fünfzig und mehr (hoffentlich unterschiedliche) Passwörter für unterschiedliche Accounts. Diese kann er sich in der Regel nicht auswendig merken. Häufig 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. Der Passwort-Manager kann mit einem Schlüsselkasten verglichen werden und das Master-Passwort mit dem Schlüssel, das den Schlüsselkasten öffnet.
PFS	<b>Perfect Forward Secrecy:</b> Ein Verfahren, das auch das nachträgliche Entschlüsseln einer aufgezeichneten, verschlüsselten Kommunikation verhindert (siehe Kapitel 14.2).
PGP	<b>Pretty Good Privacy:</b> Ein Verfahren zur Verschlüsselung von Dateien und Nachrichten mit Hilfe asymmetrischer Verschlüsselung. Mit PGP ist Übertragung Ende-zu-Ende verschlüsselter Mails möglich. Bei PGP werden die öffentlichen Schlüssel von den Benutzern wechselseitig beglaubigt. Dadurch entsteht ein sogenanntes <b>WoT</b> oder Web of Trust (siehe dort).
PGP/MIME	Ein Verfahren zur Verschlüsselung und oder Signierung von Mails mit <b>PGP</b> . Dabei wird das <b>MIME</b> -Format eingehalten. Anders als beim klassischen <b>Inline-PGP</b> werden der Mail-Inhalt und jeder Mail-Anhang separat verschlüsselt bzw. signiert. PGP/MIME ist das modernere Verfahren, das jedoch von machen Tools noch nicht unterstützt wird.
POP	<b>Post Office Protocol:</b> Protokoll zum Abruf von Mails. Anders als bei <b>IMAP</b> 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 <b>asymmetrische Verschlüsselung</b> ).
Protokoll	Ein Protokoll ist ein Satz von Regeln, welche das Format, den Inhalt, die Bedeutung und evtl. auch die Reihenfolge von Nachrichten zwischen verschiedenen Instanzen (z.B. zwischen <b>Client</b> und <b>Server</b> ) festlegen. Nur dadurch, dass beide dieselben Nachrichten „verstehen“, können sie sinnvoll miteinander kommunizieren. Beispiele: <ul style="list-style-type: none"> <li>• Das HTTP (Hypertext Transfer Protocol) regelt das Format der Nachrichten zwischen dem HTTP-Client (Browser) und dem HTTP-Server (Web-Server).</li> <li>• Das FTP (File Transfer Protocol) regelt die Nachrichten zur</li> </ul>

Begriff oder Abkürzung	Erläuterung
	<p>Übertragung von Dateien zwischen dem FTP-Client und dem FTP-Server.</p> <ul style="list-style-type: none"> <li>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.</li> </ul> <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. Denn 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 Schlüsselpaares 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 <b>asymmetrische Verschlüsselung</b> ).
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 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 Rivest, Shamir und Adleman
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 <b>Client</b> nutzt den angebotenen Dienst. Ein <b>SMTP</b> -Server z.B. steht in der Regel beim Mail-Provider und bietet dem <b>SMTP</b> -Client den Dienst an, Mails zu versenden. Das <b>Protokoll</b> definiert die Nachrichten, die der Client und der Server miteinander austauschen. Das <b>SMTP</b> -Protokoll beschreibt die Nachrichten zwischen <b>SMTP</b> -Client und <b>SMTP</b> -Server. (Als Server wird nicht nur das Server-Programm, sondern häufig auch der Rechner, auf dem das Server-Programm läuft, bezeichnet.)

Begriff oder Abkürzung	Erläuterung
S/MIME	<b>Secure MIME:</b> ein Verfahren zur Übertragung Ende-zu-Ende verschlüsselter Mails mit asymmetrischer Verschlüsselung. Dabei wird das <b>MIME</b> -Format (siehe dort) eingehalten. Anders als bei <b>PGP</b> werden bei S/MIME die öffentlichen Schlüssel von bestimmten Zertifizierungsstellen, den <b>CAs</b> beglaubigt.
Software-Repository	(engl. repository = Lager, Depot, Aufbewahrungsort) Ein Software-Repository ist eine zentrales Software-Aufbewahrungsstelle, an der viele Software-Pakete zur Installation bereitgestellt werden. Bei den Linux-Distributionen gibt es ein solches zentrales Software-Repository im Internet, an dem praktisch alle Software-Pakete einer Linux-Distribution aufbewahrt werden, z.B. das Ubuntu Linux Repository oder das SUSE Linux Repository. Dies macht die Aktualisierung des Systems sehr einfach. Diese ist mit ein paar Mausklicks erledigt. Bei Windows gibt es kein solches Repository für die Programme. Deshalb kann es recht aufwändig werden, alle Programme auf einem Windows-System aktuell zu halten. Bei iOS und Android kann man den Apple App Store bzw. den Google Play Store als Software-Repository betrachten.
SKS	<b>Synchronizing Key Server:</b> ein Key-Server, der sich automatisch mit anderen Key-Servers 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 es heute kaum noch zu kaufen gibt) genügt.
SMTP	<b>Simple Mail Transfer Protocol:</b> ein Protokoll zur Versenden von Mails.
SSL	<b>Secure Socket Layer:</b> veraltetes Protokoll zur Transport-Verschlüsselung. SSL ist der Vorläufer von <b>TLS</b> .
STARTTLS	Dies ein Verfahren, eine unverschlüsselte Verbindung in eine mit <b>SSL</b> oder <b>TLS</b> 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 unzweckmäßig, da der Absender zum Verschlüsseln und der Empfänger zum Entschlüsseln denselben Schlüssel benötigen. Dazu müsste man den Schlüssel aus der Hand geben. Dies würde eine sehr große Missbrauchsgefahr mit sich bringen.
TLS	<b>Transport Layer Security:</b> Nachfolge-Protokoll von SSL; ein Verfahren, um einen sicheren Übertragungskanal zwischen zwei Instanzen zum Transport von Daten aufzubauen. Das Verfahren wird bei verschiedenen Protokollen zur verschlüsselten Übertragung der Daten eingesetzt, z.B. für <b>HTTPS</b> (verschlüsselte Übertragung von HTTP-Nachrichten zwischen Browser und Web-Server). Das Verfahren kommt auch beim Abruf von Mails (mit POP oder IMAP), beim Versand von Mails (mit SMTP) oder bei der Übertragung von Mails zwischen den Providern zum Einsatz. Mit TLS kann bei der Mail-

Begriff oder Abkürzung	Erläuterung
	Übertragung keine <b>Ende-zu-Ende-Verschlüsselung</b> sichergestellt werden. Die Daten sind nur während der Übertragung zwischen zwei Instanzen verschlüsselt.
User-Tracking	„Verfolgung“ eines Benutzers im Internet. Dabei werden die Datenspuren, 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 anzuseigen, 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 <i>direktem Vertrauen</i> . 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 <i>transitivem Vertrauen</i> (siehe auch <b>WoT</b> , Web of Trust).
WebDAV	<b>Web Distributed Authoring and Versioning:</b> WebDAV basiert auf HTTP bzw. HTTPS. Es ist ein Protokoll zum Zugriff auf entfernte Ordner und Dateien. Spezialisierungen von WebDAV sind <b>CardDAV</b> und <b>CalDAV</b> .
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. <i>Thunderbird</i> , <i>Outlook</i> , <i>Apple Mail</i> und viele andere.
Windows	PC-Betriebssystem von Microsoft
WoT	<b>Web of Trust:</b> ein Vertrauensgeflecht, das sich aus direkten Vertrauensbeziehungen (siehe dort) und transitiven Vertrauensbeziehungen (siehe <b>Vertrauen</b> ) zusammensetzt.
Zero Knowledge	Dieser Begriff verwendet, wenn ein Provider die Daten, die er im Auftrag eines Benutzers speichert oder weiterleitet, nicht kennt; d.h. er kann sie nicht lesen oder verarbeiten. So hat z.B. die Post „kein Wissen“ über den Inhalt der Briefe, die sie transportiert. Zero Knowledge kann nur gewährleistet werden, wenn die Daten verschlüsselt sind und der Provider auch nicht den Schlüssel dazu hat. Zero Knowledge ist ein Synonym für <b>Ende-zu-Ende-Verschlüsselung</b> .