Verschlüsselung

21.1.2015

Hier möchte ich meine Erfahrungen berichten und dokumentieren:

Mein öffentlicher PGP-Schlüssel für eMail: Schlüssel-Datei

eMail und PGP

Standardmäßig ist die Kommunikation per eMail immer unverschlüsselt. D.h. jeder, der auf der Leitung vom Sender bis zum Empfänger mithorchen kann, kann den Inhalt auch lesen.
Hier ein nettes Video dazu; Der digitale Briefumschlag

Programme

Ich verwende folgende Programme:

- **gpg** (GNU privacy guard) als Hintergrund-Programm; ist auf *Kubuntu (Linux)* standardmäßig dabei
- **gpa** (GNU privacy assistant) zum Verwalten und Erzeugen der Schlüssel
- **Thunderbird**, mein eMail-Programm mit Adressbuch und Kalender
- **Enigmail**, ein Plugin für Thunderbird, das die Verschlüsselung innerhalb Thunderbird umsetzt.

Für *Windows* gibts anstatt gpg und gpa das Programm gpg4win

Installation

Die Installation ist fast selbsterklärend.

1. Schlüsselpaar mit gpa erzeugen (min RSA 2048, 1024 ist zu schwach)
2. Enigmail in Thunderbird installieren (über Tools → Add-Ons)
3. Enigmail einrichten
4. fertig.

Hinweise

Durch PGP wird nur der Inhalt der eMail verschlüsselt, also auch Anhänge. Offengelegt bleibt immer noch der Betreff, der Absender und Empfänger der eMail. Es fehlt also die Anonymisierung. Aber es ist schon viel besser, als gleich alles in Klartext zu versenden.

Jedes versendete eMail wird auch automatisch signiert. D.h. der Empfänger hat die Möglichkeit zu prüfen, ob wirklich wir die Sender der eMail sind, oder ob sich jemand als Karl Zeilhofer in meinem Fall ausgibt.

Das erzeugen des Schlüsselpaares benötigt ein Passwort (passphrase). Dieses wird beim Versenden und beim Entschlüsseln von eMails immer benötigt. Hat man kein Passwort angegeben, hat jeder, der die Datei mit dem privaten Schlüssel hat, die gleichen Möglichkeiten wie man selbst.

PGP auf Android

Nachdem die Verschlüsselung nun auf dem PC bereits funktioniert, macht es in heutiger Zeit auch Sinn, dies auch auf dem Smartphone einzurichten.

Programme

- **K-9 Mail** als eMail Programm (open source)
- **AGP** für die Schlüsselverwaltung

Installation

1. Zuerst AGP installieren.
2. Schlüssel-Backup vom PC auf SD-Karte im Handy übertragen. In gpa Schlüssel auswählen und im Menü „Keys → Backup“ verwenden. \\Wichtig: Nicht die Funktion „Export Key“ verwenden, denn hiermit würde nur der öffentliche Schlüssel abgespeichert werden.
3. Diese Datei lädt man dann in AGP mit „Import Key“. \\Hinweise: wenn man noch das USB-Kabel am Handy stecken hat, kann man auf die SD-Karte nicht zugreifen :)
4. K-9 Mail installieren und einrichten
5. Schlüsseldatei von der SD-Karte wieder löschen
6. fertig.

Hinweise

Das Passwort für den Schlüssel sollte Handy-tauglich sein. Viele Ziffern und Sonderzeichen würd ich eher vermeiden. Das Passwort kann man im Nachhinein auch verändern.

Passwort-Tresor mit KeePass

Für Verschlüsselung sind meist auch Passwörter notwendig. Überall das gleiche Passwort zu verwenden ist eine schlechte Idee. Denn wird einer der verwendeten Dienste gehackt, so bekommt der Angreifer Zugang zu allen Diensten, die man sonst noch so verwendet.

Vermutlich kann man sich aber zig verschiedene Passwörter nicht merken. Meiner Erfahrung kann man sich nicht einmal alle Seiten merken, bei denen man registriert ist.

Die Lösung ist ein Passwort-Tresor. Dieser ist nur zugänglich über ein Passwort und/oder eine Schlüssel-Datei. Eine mögliche Variante, die ich nun seit über einem Jahr in täglicher Verwendung habe, ist **KeePass**. KeePass gibt es für alle üblichen PC und Smartphone-Betriebssysteme. Weitere Infos gibt es auf Wikipedia: englisch oder deutsch

KeePass verwendet eine verschlüsselte Datenbank, die **in einer einzigen Datei** abgespeichert wird. D.h. überall, wo man diese Datei hat, und wo KeePass läuft, kann man all seine Passwörter abrufen. Weiters kann man darin auch gescannte sensible Dokumente wie Führerschein, Reisepass, Staatsbürgerschaftnachweis, TAN-Listen und ähnliches verwahren.

Meine Datenbank hat mittlerweile fast 130 Einträge. Das kann man sich beim besten Willen nicht alles merken. Vielen Dank hier an Christian M. Schmid, der mich in seinem Blog-Artikel darauf aufmerksam gemacht hatte