

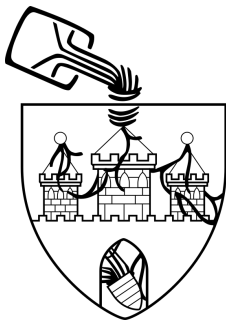
„Meine Mails gehören mir!“

Workshop zur E-Mail-Verschlüsselung

Peter Gewalt & Manuel Groß

2015-11-28

Wer sind wir?



<https://ccc-ol.de/>



<https://mainframe.io/>

Was ist eigentlich unser Problem?



CC-BY-SA 2.0 Markus Winkler

- Massenhafte, anlasslose Überwachung
- False Positives
- Chilling Effect

Sicherheitsziele

- ❶ Verschlüsselung
 - Text kann nicht von anderen gelesen werden
- ❷ Authentizität
 - Identität des Absenders sichergestellt
- ❸ Integrität
 - Nachricht nicht von dritten verändert

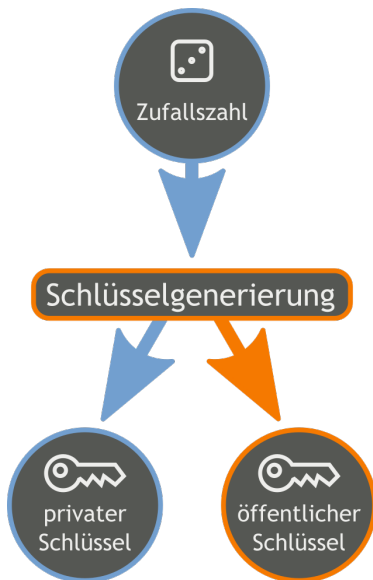
Ziel 1: Verschlüsselung (Abgrenzung)

Was ist damit nicht gemeint?

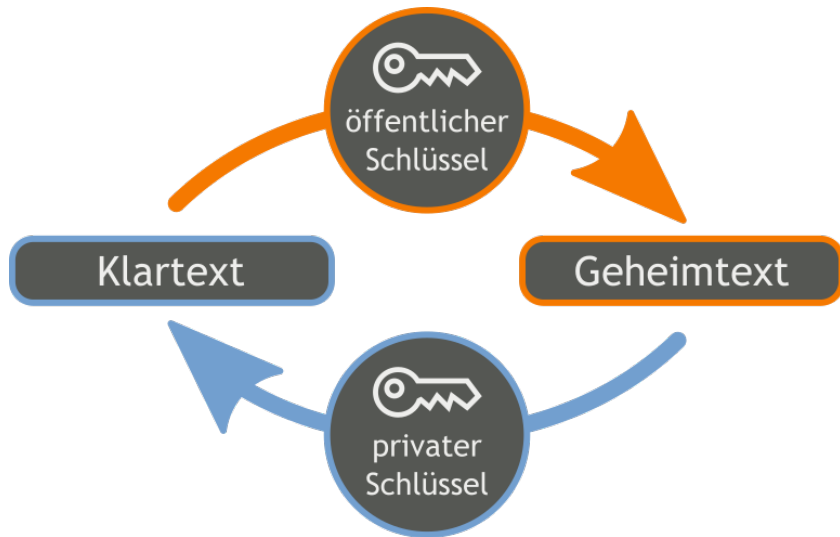
- Webseiten (http(s))
 - Ende-zu-Ende Verschlüsselung (Transportverschlüsselung)
- Messenger
 - Verschieden: Transportverschlüsselung, manchmal Ende-zu-Ende Verschlüsselung
- Mail
 - Per default unverschlüsselt
 - Typisch zum Server, nicht Ende-zu-Ende!
- DE-Mail
 - ▶ per default nur abschnittsweise verschlüsselt

⇒ Nachrichten am Speicherort ungeschützt

Asymmetrische Verschlüsselung



Asymmetrische Verschlüsselung



Ziel 2: Authentizität (Vertrauen)

Problem:

Wer garantiert uns, dass Bob wirklich der Absender ist?

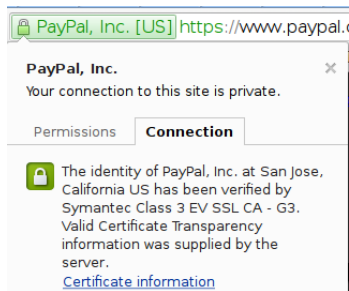
Ansätze:

- a) Zentral
- b) Dezentral

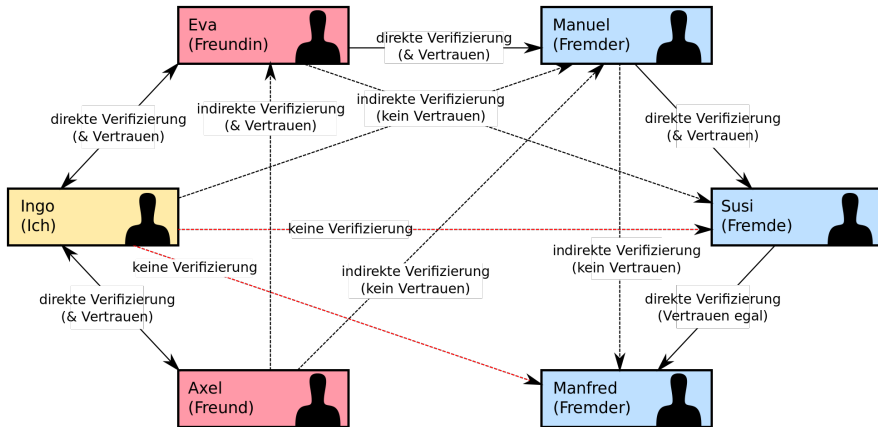
Ansatz a) Zentral (Server)

- Idee: zentraler Ansatz

- ▶ Indirekter Austausch über Dritte (z.B. Browser oder Mailprogramm)
- ▶ Hierarchie/Zertifikatsliste (schonmal reingeschaut?)



Ansatz b) Dezentral: Vertrauensnetzwerk (Web of trust)



CC-BY-SA 3.0 Hauke Laging

Cryptoparties

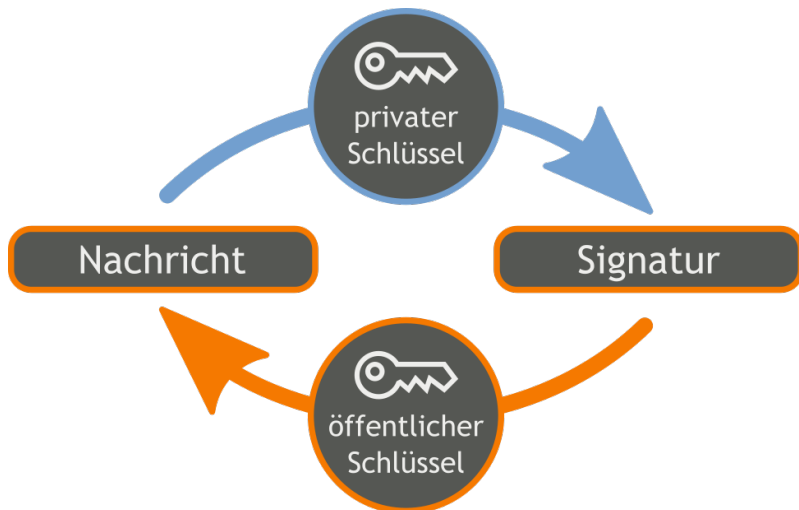
```
pub      rsa4096/63B7B190 2014-10-18
        Key fingerprint = FF9A C23C EBC5 CD0D 3998  CB4A CB77 6720 63B7 B190
uid      [ultimate] Manuel Groß <manuel.gross@freifeld-festival.de>
uid      [ultimate] Manuel Groß <mgr@nordkrater.de>
```



CC-BY 2.0 wbritzl

- Neue Leute kennenlernen
- Spaß an lustigen, alten, Ausweisbilder anderer haben
- Selbst Cryptoparties veranstalten ⇒ Web of trust stärken

Ziel 2 und 3: Authentizität und Integrität mittels Signatur



GPG

Was ist jetzt eigentlich GPG?



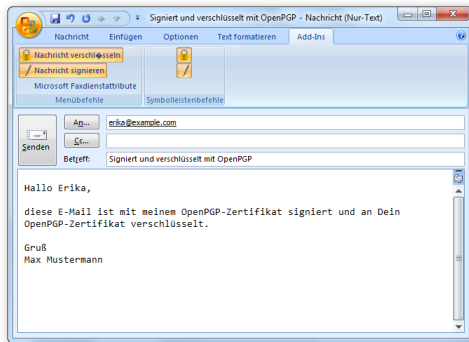
<https://gnupg.org>

- „Gnu Privacy Guard“, Open Source-Implementierung von OpenPGP
- „PGP“ („Pretty Good Privacy“), alternative, nun proprietäre Implementierung
- Sehr verbreitet im Open Source Umfeld
- Eher ungebräuchlich im geschäftlichen Umfeld
- Web of Trust Prinzip
- Plugins für viele Clients
- Metadaten **nicht** verschlüsselt!

- **Demo:** Mail (Source, Header)

Beispiel: GPG in MS Outlook

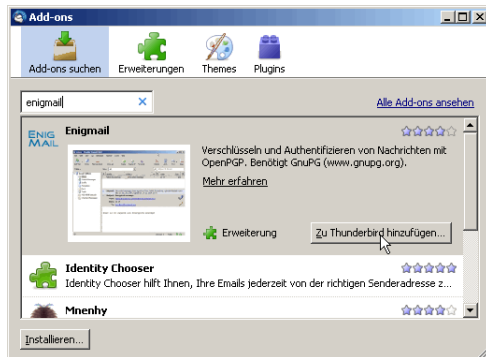
- Gpg4win: OpenPGP und S/MIME
- MS Outlook 2003/2007/2010 und 2013 (32 Bit)
- Windows XP, Vista, 7 and 8
- https://www.gpg4win.de/doc/de/gpg4win-compendium_11.html



http://www.gpg4win.org/img/sc-gpgol-sendSignEncryptedMail_de.png

Beispiel: GPG in Thunderbird

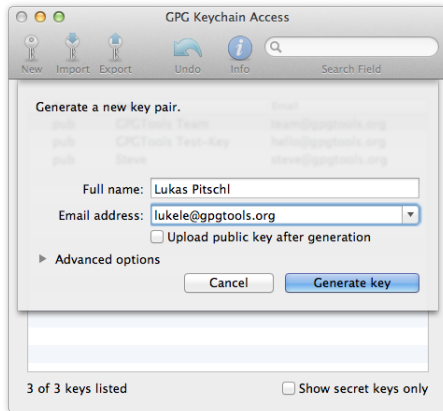
- Ebenfalls Gpg4win installieren
- Zusätzliches Plugin für Thunderbird nötig: Enigmail
- Anleitung <http://blog.ch-becker.de/2011/04/27/pgpgpg-unter-windows-mit-thunderbird-emails-verschlusseln/>



http://blog.ch-becker.de/wp-content/uploads/2011/04/enigmail_suche.png

Beispiel: GPG mit MAC OS X

- <https://gpgtools.org/>



<https://gpgtools.org/images/screenshots/gka-create-key.1375965203.png>

PGP funktioniert!

TOP SECRET//COMINT//REL TO USA, AUS

TOP SECRET//COMINT//REL TO USA, AUS//20320108

THIS INFORMATION IS DERIVED FROM FAA
COLLECTION UNDER FAA COUNTERTERRORISM CERT

THIS INFORMATION IS PROVIDED FOR INTELLIGENCE PURPOSES IN AN EFFORT
TO DEVELOP POTENTIAL LEADS. IT CANNOT BE USED IN AFFIDAVITS, COURT
PROCEEDINGS OR SUBPOENAS, OR FOR OTHER LEGAL OR JUDICIAL PURPOSES.

██████████@yahoo.com

████████████████████

SIGAD: US-984XN
PDDG: AX
CASE_NOTATION: ██████████
DTG: 31JA0101Z12

Received from: [MINIMIZED US IP ADDRESS]
Date: Mon, 30 Jan 2012 17:01:37 -0800 (PST)
From: ██████████ ██████████@yahoo.com>
Subject: Re: Untitled
To: ██████████@yahoo.com

[OC: No decrypt available for this PGP encrypted message.]

Vorgehen

Erstellen (einmalig):

- ➊ Schlüsselpaar erstellen
- ➋ Optional: Auf Schlüsselserver (keyserver) laden

Signieren:

- ➌ Fremden Schlüssel (public key) laden
- ➍ Fingerprint prüfen/vergleichen
- ➎ → Identität der Person prüfen (Ausweis) ←
- ➏ Schlüssel signieren
- ➐ Signieren Schlüssel auf Schlüsselserver laden oder per Mail schicken

GPG unter Linux - Installation

Debian Pakete:

- gnupg2
- signing-party (für caff)
- msmtptt

GPG unter Linux - Konfiguration

- Config: `~/.gnupg/gpg.conf`
- `keyserver pgp.mit.edu`
- Vorteil: Kein manuelles `--keyserver <keyserver>`

- **Demo:** Schlüssel erstellen, Signieren

1: GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
 - ▶ RSA oder DSA?
 - ▶ Schlüssellänge?
 - ▶ Gültigkeit?
 - ▶ Name (kein Pseudonym)
 - ▶ Mailadresse
 - ▶ Passphrase (Langes_Passwort > S4l4t)
- Upload (sofern öffentlich):
 - `$ gpg --send-keys <key-id>`

GPG unter Linux - Signieren

3

- Laden des fremden Keys:

```
$ gpg --recv-keys <key-id>
```

- Prüfung des Fingerprints:

```
$ gpg --fingerprint <key-id>
```

- Identitätsprüfung (Personalausweis, Führerschein)

- Signieren:

```
$ gpg --sign-key <key-id>
```

- Passphrase eingeben

- Upload-Varianten:

Unsicher Direkter Upload auf Keyserver

Sicher Verschicken des signierten Keys per Mail **an die signierten Mailadressen**

GPG unter Linux - Signieren mit caff

- Automatisch:
 \$ caff <key-id>
- Benötigt konfigurierten SMTP-client
- Beispiel msmtprc
- Config: ~/.msmtprc
 - ▶ *host, port, user, password*

GPG unter Linux - Empfängerseite

- Datei *signature.asc* per Mail bekommen
- Signatur importieren:
`gpg --import signatur.asc`
- Signatur Uploaden:
`gpg --send-keys <meine key-id>`
- Done! Ready for GPG-Mails!

GPG Keys Revoken

- Entweder automatisch (Gültigkeit) oder manuell
- Wichtig: **Vorher** Cross-signieren
- Revoken \neq Löschen
- Widerrufszeugnis:
`gpg --gen-revoke <key-id>`
- Importieren, Uploaden
- ggf. direkt nach Generierung erzeugen
- Besser: Backup private key!

Fazit

- Ende-zu-Ende Verschlüsselung als sicherer Kommunikationsweg
- Metadaten nicht verschlüsselt!
- Je mehr Nutzer, desto funktionaler!

Hands-on - Let's go!

❶ Schlüsselpaar erstellen

- ▶ `gpg --gen-key`
- ▶ (keyserver definieren)
- ▶ `gpg --send-keys <key-id>`

❷ Signieren mit Identitätsprüfung (z.B. ohne caff)

- ▶ `gpg --recv-keys <key-id>`
- ▶ `gpg --fingerprint <key-id>`
- ▶ `gpg --sign-key <key-id>`

❸ Signaturen per Mail verschicken

- ▶ `caff <key-id>`

❹ Uploaden auf den Keyserver (Empfänger)

- ▶ `gpg --import signatur.asc`
- ▶ `gpg --send-keys <meine key-id>`