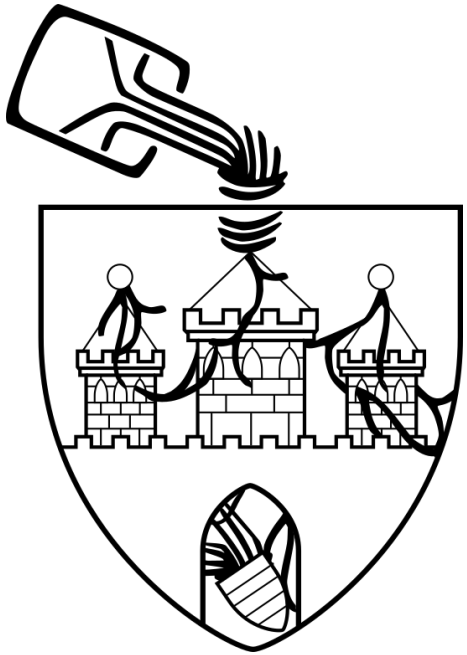
„Meine Mails gehören mir!“
Workshop zur E-Mail-Verschlüsselung

Peter Gewald & Manuel Groß

2015-11-28

https://github.com/ktt-ol/sichere-e-mails-mit-pgp

Wer sind wir?



https://ccc-ol.de/

https://mainframe.io/

Was ist eigentlich unser Problem?

- Massenhafte, anlasslose Überwachung
- False Positives
- Chilling Effect

Sicherheitsziele

1. Verschlüsselung
   - Text kann nicht von anderen gelesen werden
2. Authentizität
   - Identität des Absenders sichergestellt
3. Integrität
   - Nachricht nicht von dritten verändert

Lösungsvorschlag: Verschlüsselung

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2

hQIMA9yhUHWYLFXlAQ/+O7en6i6BQQh9lEwQ2Zu/kFKXHWO9GGNeXiPVtdGKMiUv
FJwfZb7lL6hrpwNZJM8+jYWwKD7ny+nP4CHV5UlGOBLhMLMqPVM440TmR9APaiBf
qUrUs8UzQ2syQCC52EUYGmwkRkjQofIVSsJj99TOwXtOM2OzAdvlMQUH54HqFRRB
OuQq9Q437kPW8nBX226vWl/Sc6OOm2nwHmmkNrn+QB+sNjmsVaWm+Cp5Y0o8k7vv
qAdbUyRDddVVUKQzhNN+0PXiNolAzXUOeqBGnVhWL5OqYZiCyat5lHIRu+lURUQz
nzdlIZyssDis2rN/iVj8RcDWrpTmiKuKQAtKXIstLziYrQh6+2Nf37IztLjbIRkJ
GkTuZSMAwJurlpDzWriK3t+sxXqF9drOxHlKSE9sM3EPZCYomdTdBVTep67iQOmI
5K8TxdO4Iack8VHbVRd5IBJy8lVVnSBUyCdV5yjjx+KMxfZxUP1X9YmOmB/jhO6p
d/eLU9IwomWNZtk7I8hVpVpmm8ByT+fbwIi33aUCcCDa8jc7vC6FsA3kjjg7aDDl
r3n0CR19LPibdyAMliHDddxOae4wbU2hN5aBFEiHBCOJPceI9euPrZmfzIMO3ou3
IP6cZBGLTawpHpz7NSzYBCAgKAXlQqXG2ptvVVL1Z96KhSxhZZ/LQc0MdySoYUOF
AgwDywGdOQBy5bYBD/0b5R7JBGt6aFN6MymPrQXt1gEU4n9n+FRCvXVpO3elCwly
ZGJ8QjAaOSP0QmOECq+Q5CsQJMPzmXg08Km7aKjzYQnDy6ZAAD2VEbSfZkue0kQ3
bIbfbyeANOA1G80JTWsJSA2mPLXvD+E856jb6xVZpoAmrJjFfiSu2/awzw8giygH
mVB+2wEUxc18ollUdaPFketjEfWE0OnAmYo2hT8WEgvviNNYT7pWll0NHU8kXZF3
j8ZyD377MNO4sR0il3f557L+e+zDJw6tCuDVQZjmeVihUijBlNNbwkZAC/+Dv8nS
mJ9u9ZA9i2PO635SmZjseYWsphGhsI+2nqswtmvw2lYr9raX+PUYyG5nIAtD7CYr
ge4OnwHDFmFbIPrjd6pzj+q6GdeOKCf0YJy+MlLtCjOINAsWFlyudTOcrnROok0Z
MXHwql/Fz8XGQMqt4bqZosVgQCL9CXgmrx0WMGQh/dZgB7BOhkVGSXi/Fe5Rwlkz
gqYCd5n5qJtT9e5RRELDGIBwaIJWV8TUXfhlOKBQsf5FT5zUr0ELSAzz+crkJkkh
U+sloRzI47luNQoCbCehuyl6sZ9niNlvrgqtlcXvGVamZSMqH0lUxN6dNh+ASu9w
pHorKveh7Qa/AGBc3oLZfo7OOB46oTtTXZbQxzR9tEHzxtC0IxCsLr6HDcCqTNLt
AbNUOonluuG+Oi7pxoBtvCg5UDnIfyBpPv0vExXPYnPAEOlRlt2dgEqQN3xnlxTA
E982fPxAWcMSJyUIkiHxlNTYZdDXRoNgVQl4JA6YFmjf/vl/4dQGbN0vkhIyNgsB
8RgAqArhwkdvd/+aD9XduQzGZQd2tUg7p4OmIj/yPi3/m5VaWjuUAUcdGOOO8bkW
EL6YwWFfKi+FWlUGj5CRNrsphpXapiS5Gzxrq5Noh30gGY0ip3/Y81E/MldCKYU5
yHRJmhJa4Clz/Z+zgOkiQvVtljOOMeCjLK1Yf3/HHUanMqn1PNcydcVppmj9/jpM
8bQ2eIWjiysyGrRyga1+tAmEmpC6Qi+d/zrT8671zI2qbl+fX0Kd3j6nqm8aGgQX
6BFCMBarviOtZlMqwtOkUWoHGWzZervZcZlbpYBrK/LMJbHSkOgpGs5CU6gVIaKl
tGOCBGDWO/BE9mamEejGtAujWE4tT8OiRlCNayV69WCfDuMNIDhuEl7PNaLQiuxU
beEExuehYMw8/+ytfrdz9gyzCU6478DmupGjclvhemAPkxqJsANLeC5IqLk7HYva
4OhpP/BgjkQlI2RbP85W1cpDl32Wz6jAzSfxrNbf57Ddx9Fhh49HDxc5Tv0Ig+wA
D9xUo9oT3WOcljEyEV970MXJRdns6FdwP2YQ6fSUjYll/4I5xYaU6ETZsJgAzz6X
Hxq4bM+m9+QMtuJTX5SRXeyTWOsdrIK3i/lHARDcPjHQIo3MSAW3AUScbMmPGaVB
uGF+lrbNgg3gVCtJaW3WrnbuPJfw+iNtL6Ic3g0iiNO0zkoAbyOtztQyYzEeFp4F
M9JwvCnBLTLbJEnr3zGtfIf18esV797gX/dvS9EV9MRzDEm+CcxRpGlM1ymSIacP
```

- Klartext $\rightarrow$ Geheimtext
- mathematisches Verfahren
- geheime Zahl als „Passwort"

- symmetrisch
  - selbes Verfahren
  - selber Schlüssel

- asymmetrisch
  - selbes Verfahren
  - unterschiedliche Schlüssel

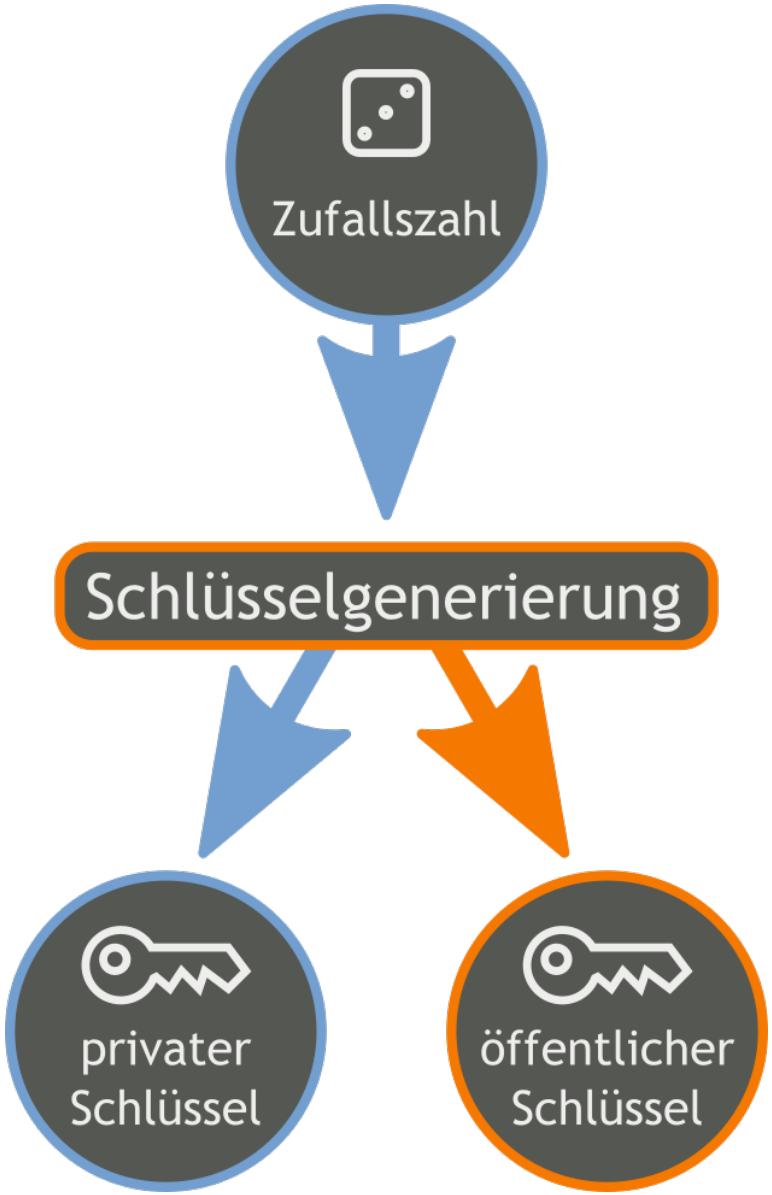
Ziel 1: Verschlüsselung (Abgrenzung)

Was ist damit nicht gemeint?

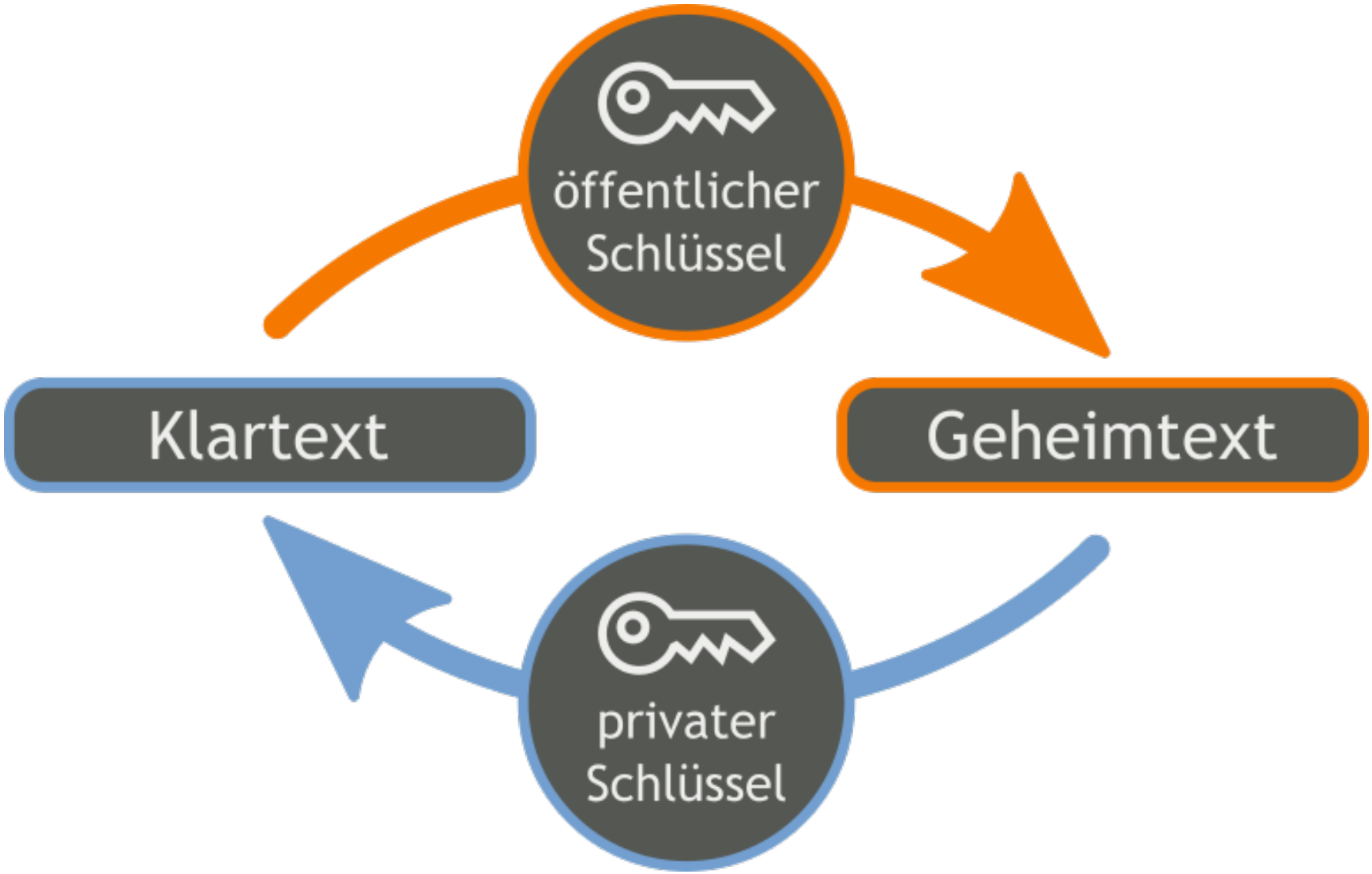
- Webseiten (http(s))
  → Ende-zu-Ende Verschlüsselung (Transportverschlüsselung)
- Messenger
  → Verschieden: Transportverschlüsselung, manchmal Ende-zu-Ende Verschlüsselung
- Mail
  → Per default unverschlüsselt
  → Typisch zum Server, nicht Ende-zu-Ende!
- DE-Mail
  - per default nur abschnittweise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

⇒ Nachrichten am Speicherort ungeschützt

Asymmetrische Verschlüsselung



Asymmetrische Verschlüsselung



Ziel 2: Authentizität (Vertrauen)

Problem:

Wer garantiert uns, dass Bob wirklich der Absender ist?

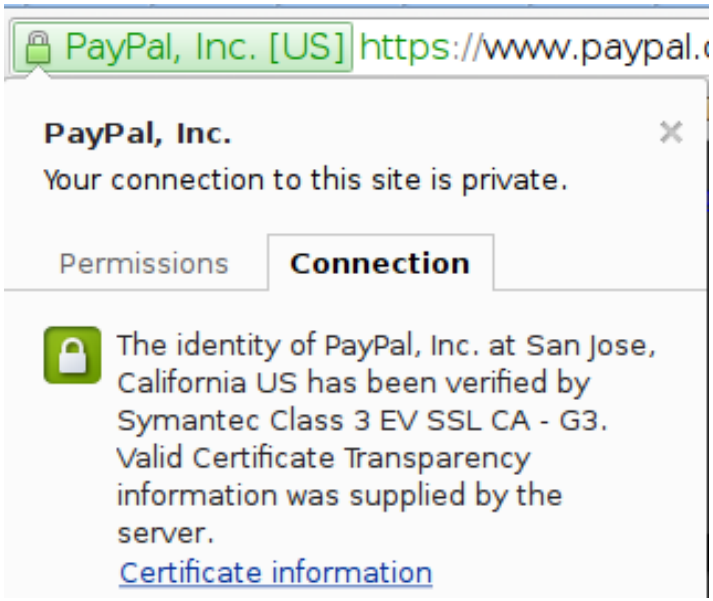
Ansätze:

a) Zentral

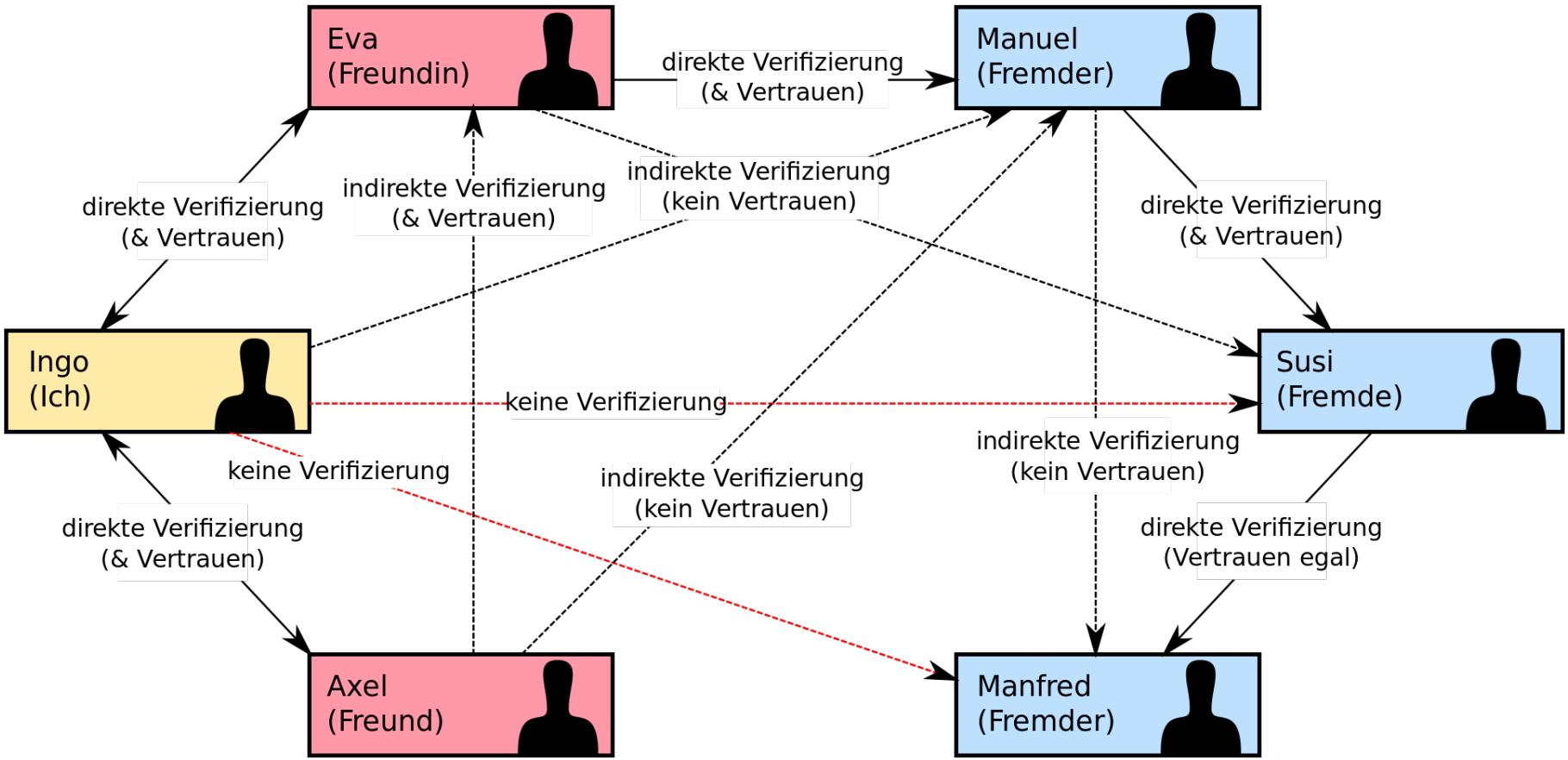
b) Dezentral

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)



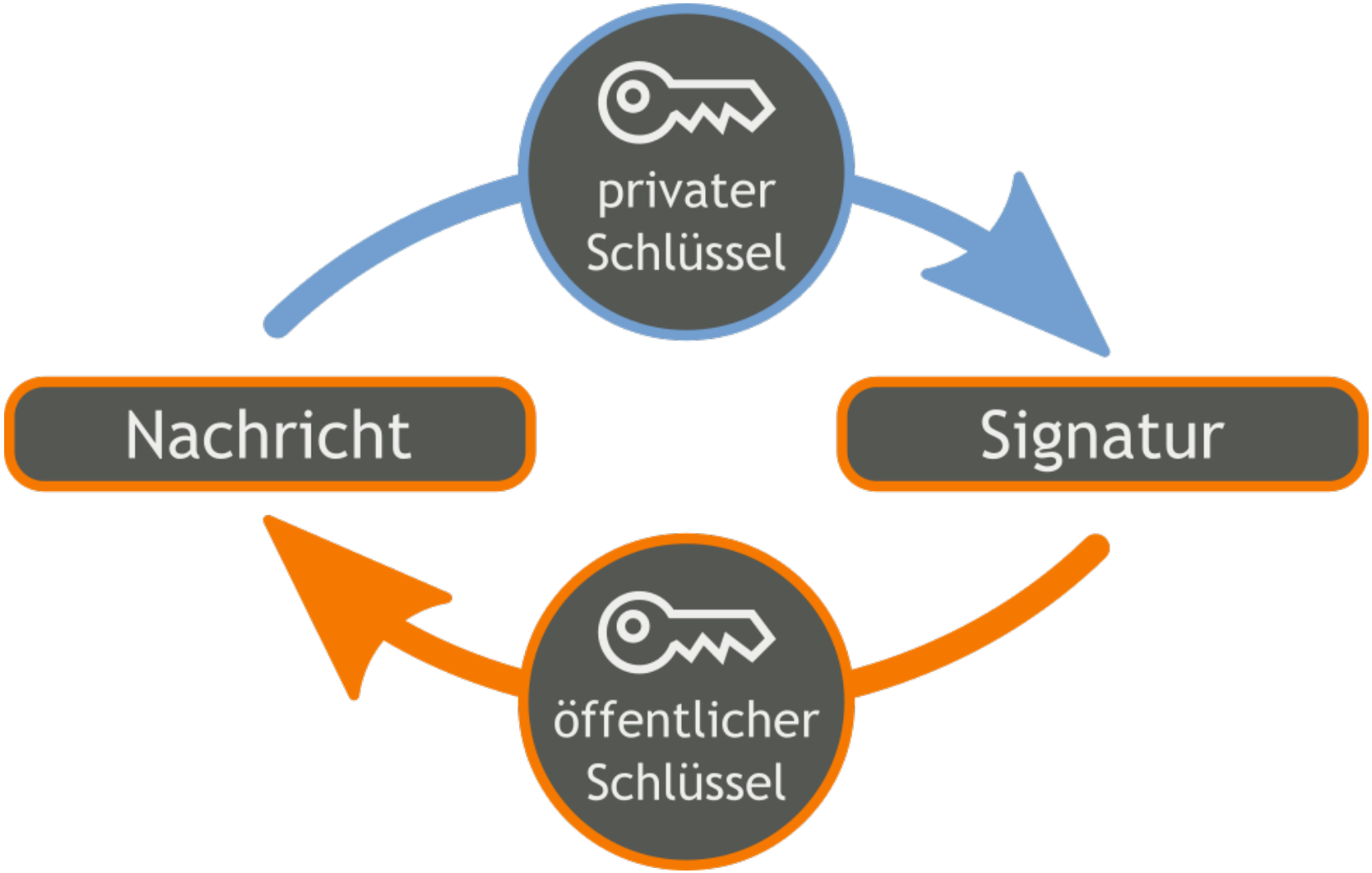
Cryptoparties

```
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] M[illegible]el 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 Intigrität mittels Signatur



GPG

Was ist jetzt eigentlich GPG?

https://gnupg.org

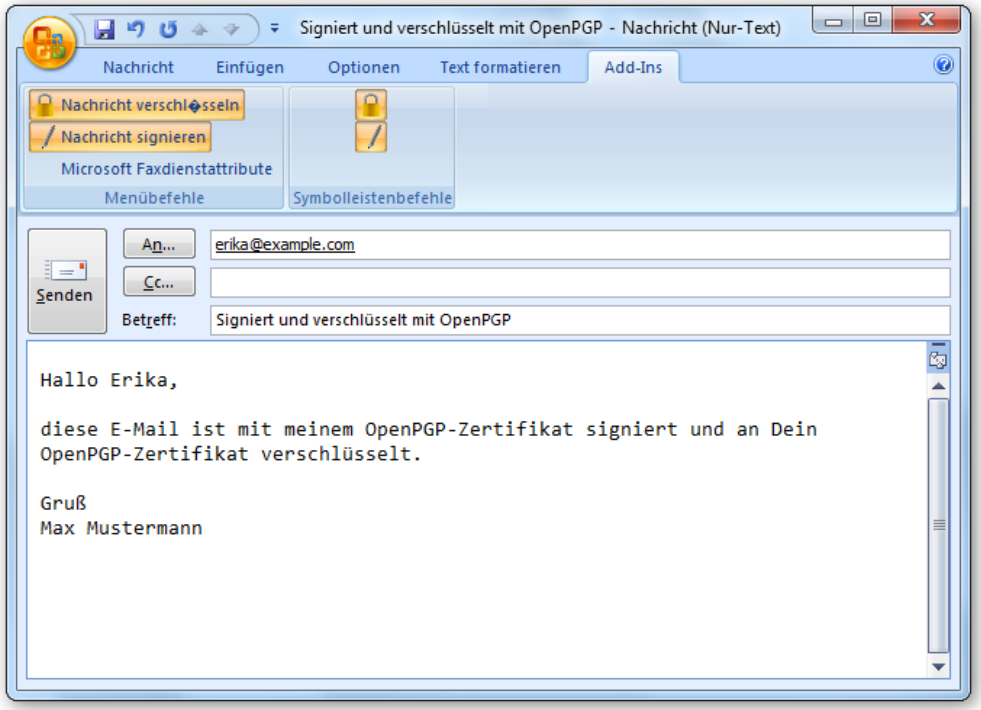
- „Gnu Privacy Guard“, Open Source-Implementierung von OpenPGP
- „PGP“ („Pretty Good Privacy“), alternative, nun proprietäre Implementierung
- Sehr verbreitet im Open Source Umfeld
- Eher ungebräuchlich im geschäftlichen Umfeld
- Web of Trust Prinzip
- Plugins für viele Clients
- Metadaten **nicht**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

GPG

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

Beispiel: GPG in MS Outlook

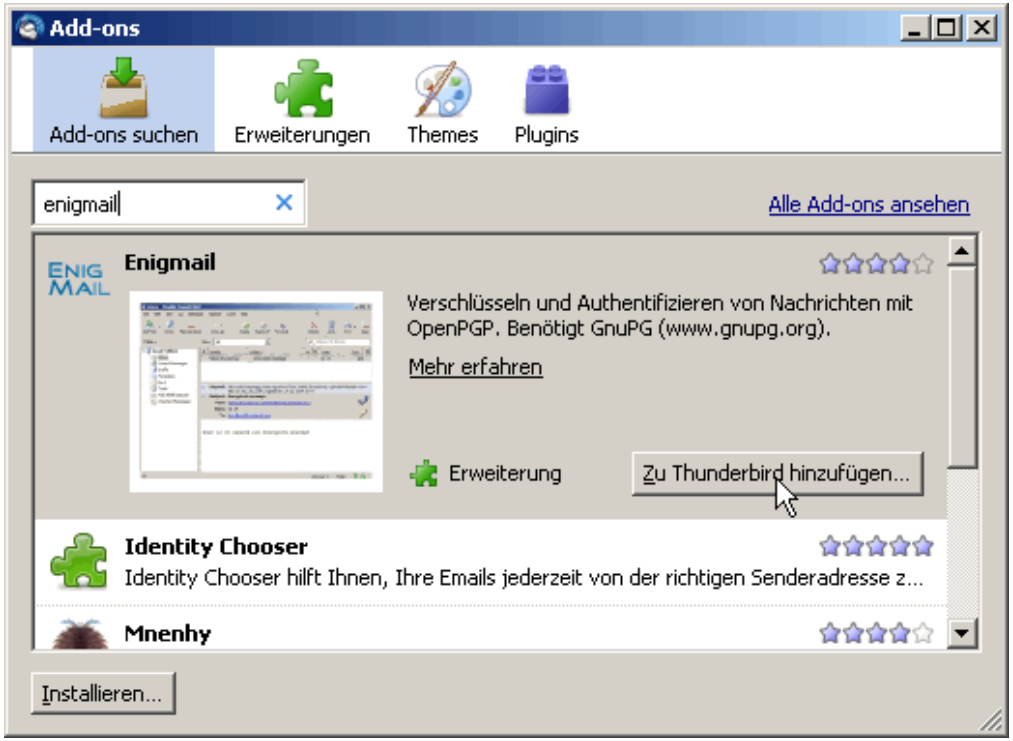
- Gpg4win: OpenPGP und S/MIME
- MS Outlook 2003/2007/2010 und 2013 (32 Bit)
- Windows XP, Vista, 7 and 8
- https://www.gpg4win.de/doc/de/gpg4win-compendium_11.html



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

Beispiel: GPG in Thunderbird

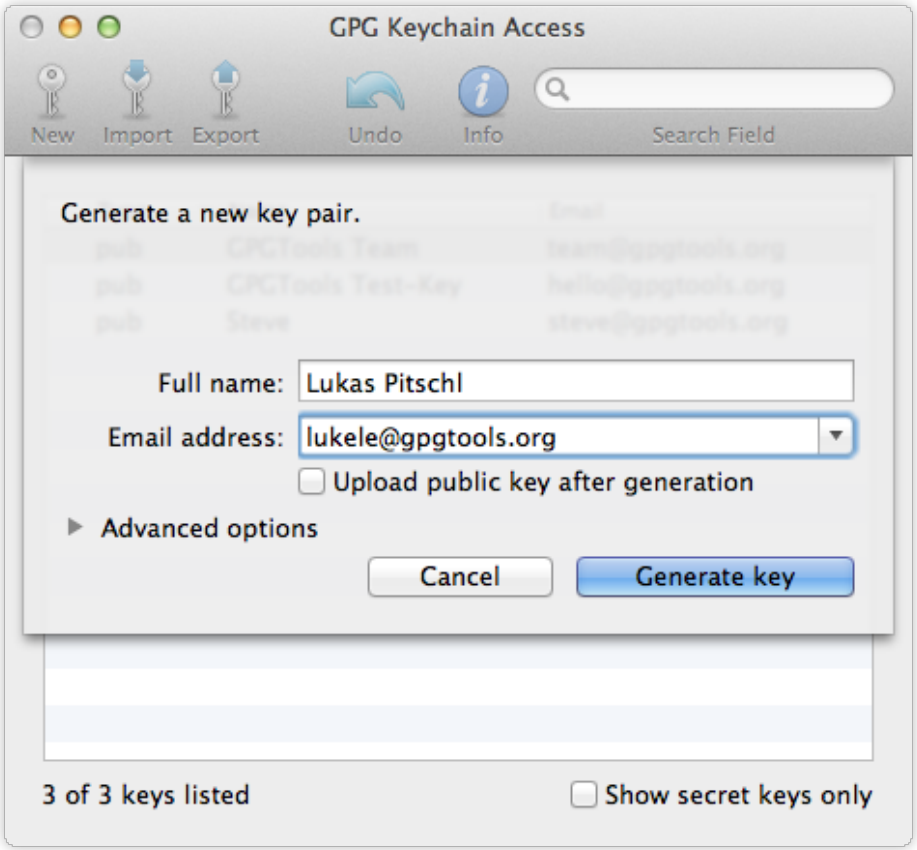
- Ebenfalls Gpg4win installlieren
- Zusätzliches Plugin für Thunderbird nötig: Enigmail
- Anleitung http://blog.ch-becker.de/2011/04/27/pgpgpg-unter-windows-mit-thunderbird-emails-verschlusseln/



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

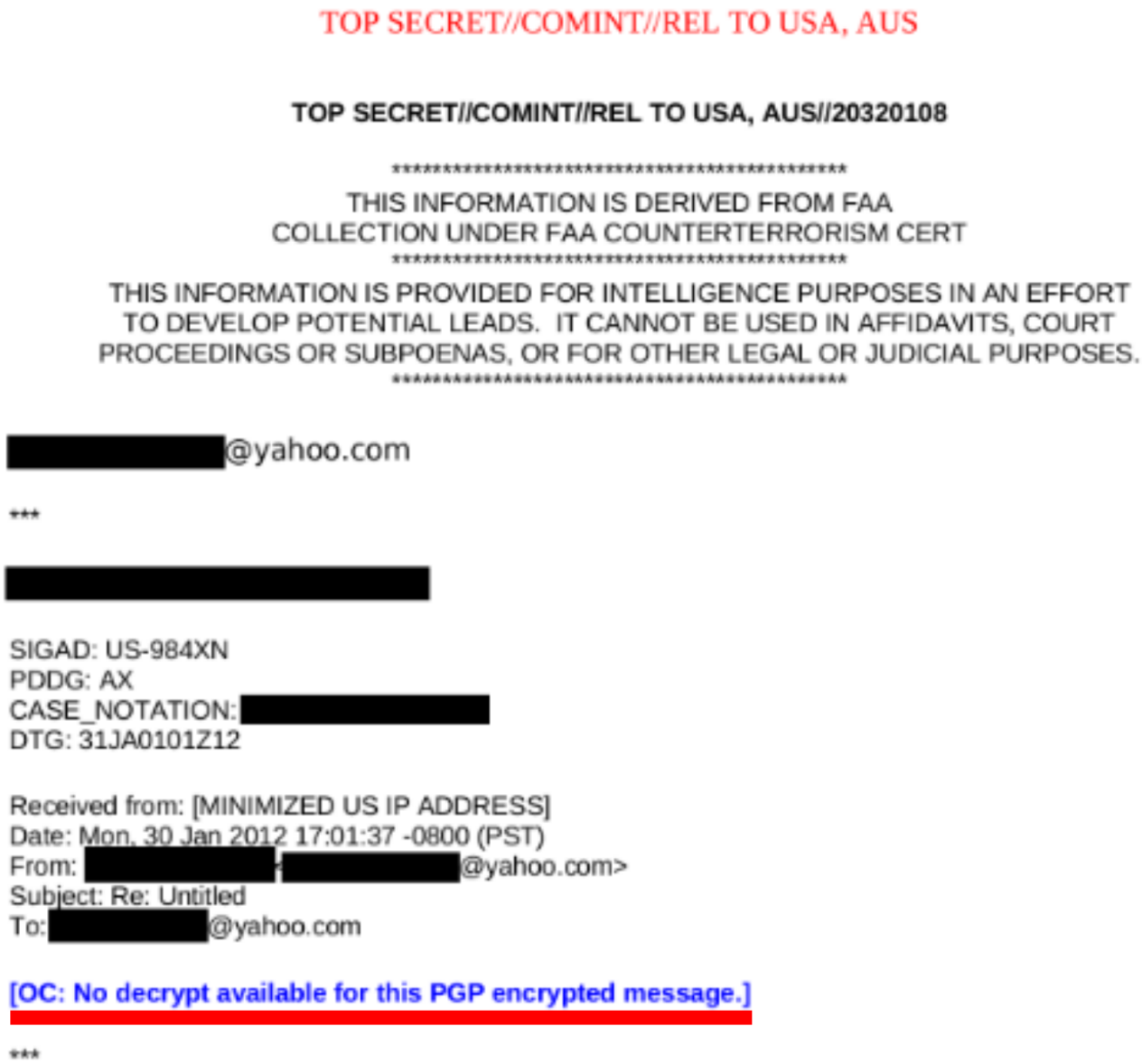
Beispiel: GPG mit MAC OS X

- 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.
****

[redacted]@yahoo.com

***

[redacted]

SIGAD: US-984XN
PDDG: AX
CASE_NOTATION:[redacted]
DTG: 31JA0101Z12

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

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

***

Vorgehen

Erstellen (einmalig):

1. Schlüsselpaar erstellen
2. Optional: Auf Schlüsselserver (keyserver) laden

Signieren:

3. Fremden Schlüssel (public key) laden
4. Fingerprint prüfen/vergleichen
5. $\rightarrow$ Identität der Person prüfen (Ausweis) $\leftarrow$
6. Schlüssel signieren
7. Signieren Schlüssel auf Schlüsselserver laden oder per Mail schicken

GPG unter Linux - Installation

Debian Pakete:

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

GPG unter Linux - Konfiguration

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

GPG - Live

- **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 msmtp
- Config: ∼/.msmtprc
  - *host*, *port*, *user*, *passwort*

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
- Widerrufszertifikat:

```
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!

1. Schlüsselpaar erstellen
   - `gpg --gen-key`
   - (keyserver definieren)
   - `gpg --send-keys <key-id>`
2. Signieren mit Identitätsprüfung (z.B. ohne caff)
   - `gpg --recv-keys <key-id>`
   - `gpg --fingerprint <key-id>`
   - `gpg --sign-key <key-id>`
3. Signaturen per Mail verschicken
   - `caff <key-id>`
4. Uploaden auf den Keyserver (Empfänger)
   - `gpg --import signatur.asc`
   - `gpg --send-keys <meine key-id>`