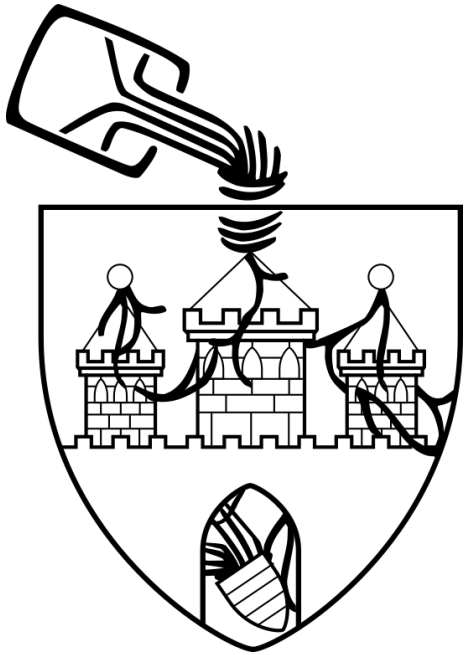
Sichere E-Mails mit PGP

Peter Gewald & Manuel Groß

2015-11-15

Wer sind wir?



https://ccc-ol.de/

MAIN FRAME
HACKSPACE OLDENBURG

https://mainframe.io/

Was ist eigentlich unser Problem?

CC-BY-SA 2.0 Markus Winkler

Was ist eigentlich unser Problem?

- Massenhafte, anlasslose Überwachung

Was ist eigentlich unser Problem?

CC-BY-SA 2.0 Markus Winkler

- Massenhafte, anlasslose Überwachung
- False Positives

Was ist eigentlich unser Problem?

CC-BY-SA 2.0 Markus Winkler

- Massenhafte, anlasslose Überwachung
- False Positives
- Chilling Effect

Lösungsvorschlag: Verschlüsselung

- wandelt Klartext in unlesbares Kauderwelsch um
- kann mit Kenntnis über Verschlüsselung und dem Schlüssel umgekehrt werden
- üblicherweise ein mathematisches Verfahren mit geheimem Eingabewert

Lösungsvorschlag: Verschlüsselung

- wandelt Klartext in unlesbares Kauderwelsch um
- kann mit Kenntnis über Verschlüsselung und dem Schlüssel umgekehrt werden
- üblicherweise ein mathematisches Verfahren mit geheimem Eingabewert

- symmetrisch
  - ver- und entschlüsseln mit selbem Schlüssel und selben Verfahren

Lösungsvorschlag: Verschlüsselung

- wandelt Klartext in unlesbares Kauderwelsch um
- kann mit Kenntnis über Verschlüsselung und dem Schlüssel umgekehrt werden
- üblicherweise ein mathematisches Verfahren mit geheimem Eingabewert

- symmetrisch
  - ver- und entschlüsseln mit selbem Schlüssel und selben Verfahren

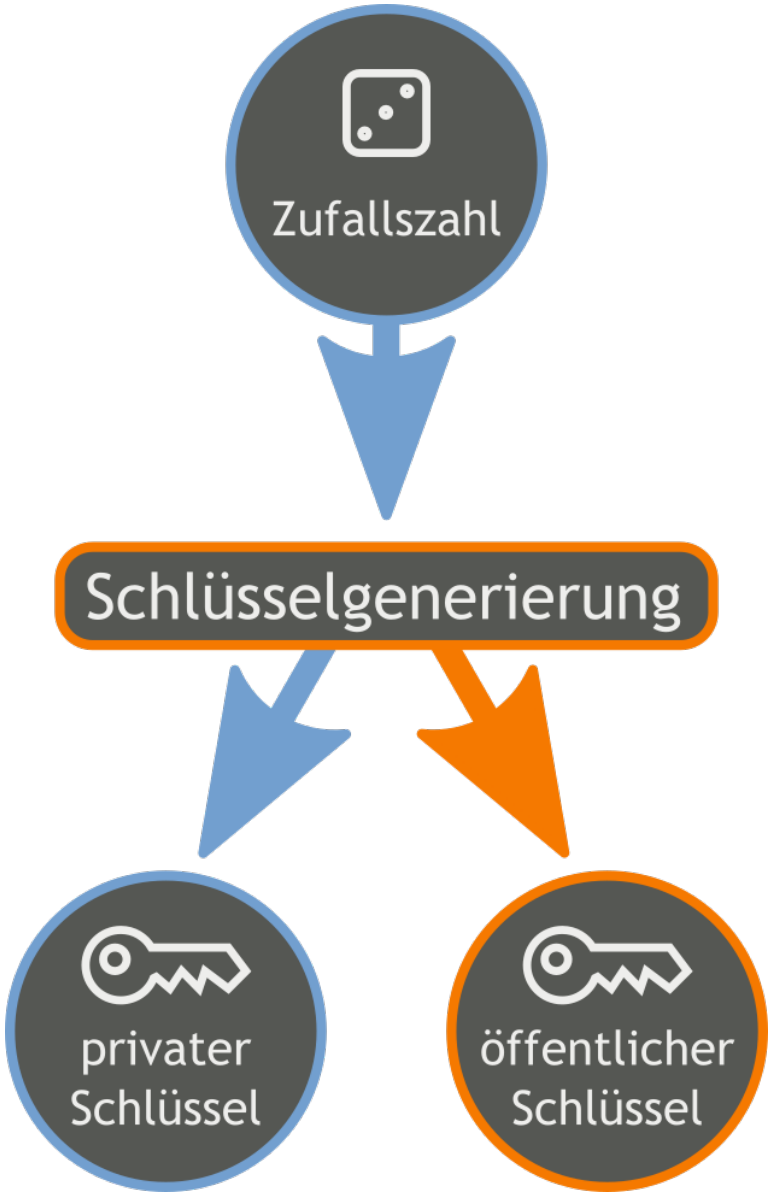
- asymmetrisch
  - ver- unt entschlüsseln mit unterschiedlichen Schlüsseln und selbem Verfahren

Unverschlüsselung

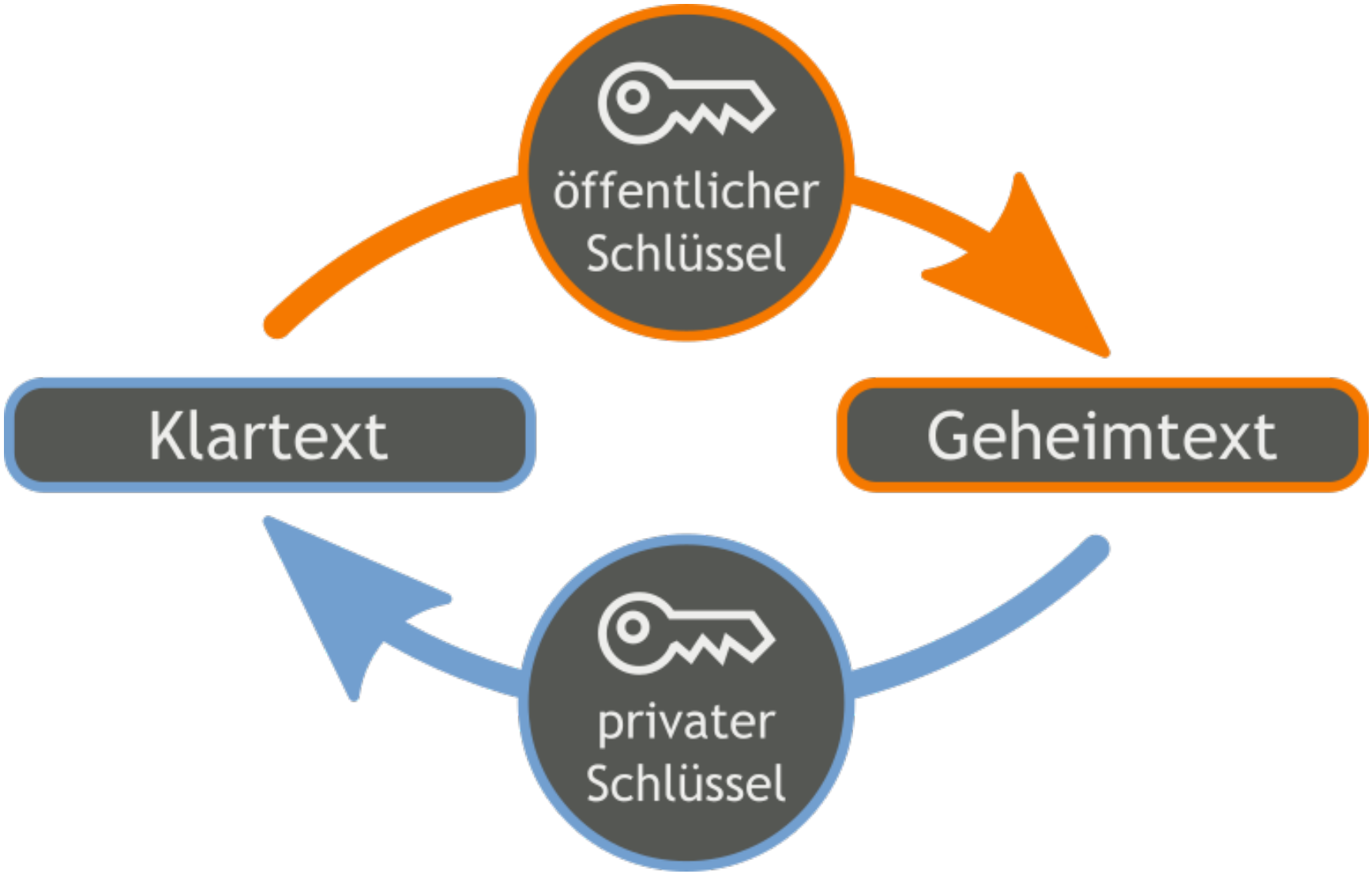
Was ist damit nicht gemeint?

- Transportwegeverschlüsselung (TLS, ...)
- DE-Mail (jemand anderes hat auch einen Key)

Asymmetrische Verschlüsselung



Asymmetrische Verschlüsselung



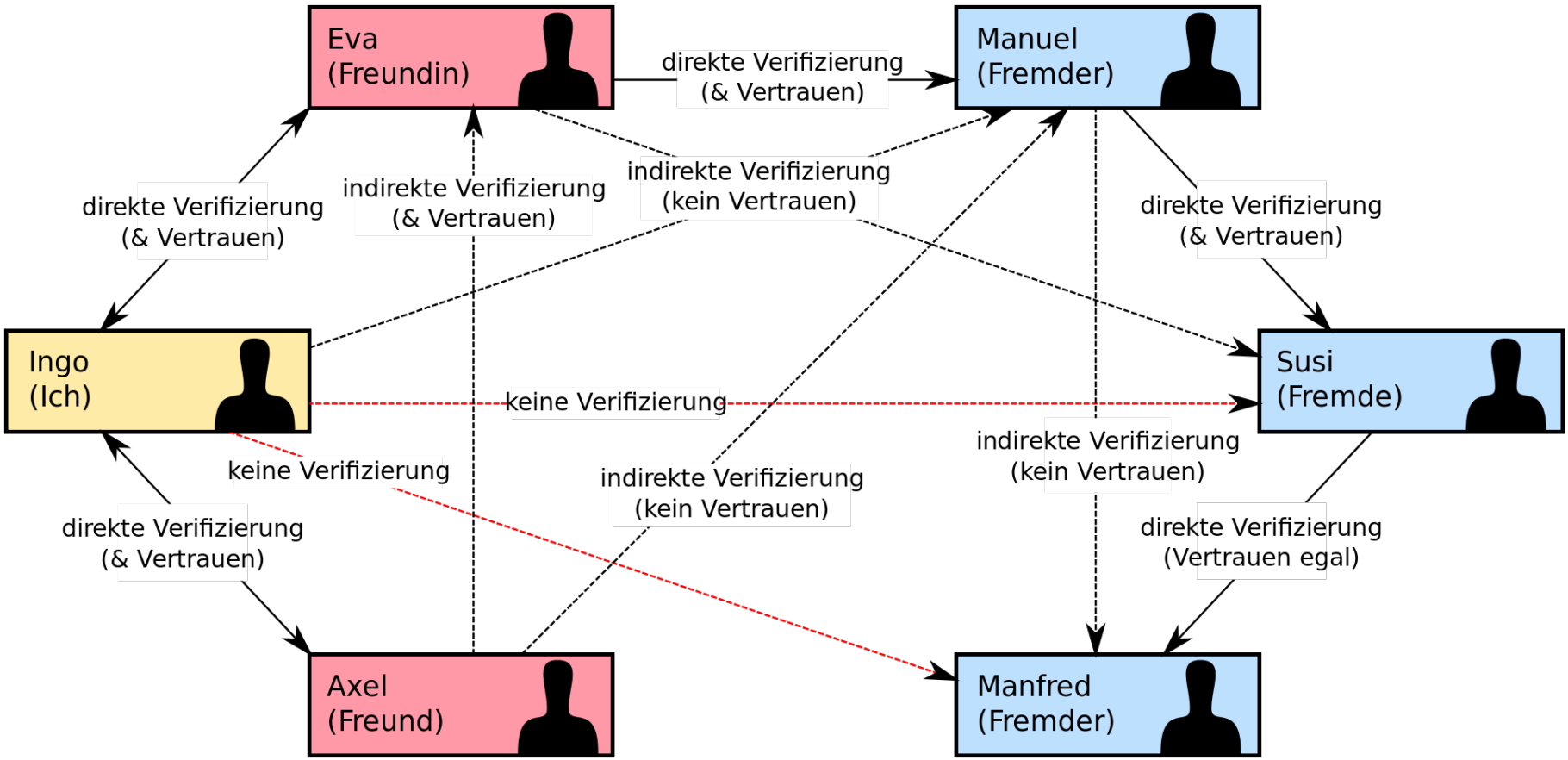
Problem: Vertrauenswürdigkeit

Problem:

Wer garantiert uns, dass Bob wirklich der Absender ist?

- Idee: zentraler Ansatz
  - indirekter Austausch über Dritte (z.B. Browser oder Mailprogramm)
  - Hierarchie/Zertifikatsliste (schonmal reingeschaut?)

Unterschiedliche Ansätze: Verteilt



Cryptoparties

CC-BY 2.0 wbritzl

Cryptoparties

CC-BY 2.0 wbritzl

- Neue Leute kennenlernen

Cryptoparties

CC-BY 2.0 wbritzl

- Neue Leute kennenlernen
- Spaß an lustigen, alten, Ausweisbilder anderer haben!

Cryptoparties

CC-BY 2.0 wbritzl

- Neue Leute kennenlernen
- Spaß an lustigen, alten, Ausweisbilder anderer haben!
- Irgendwas mit Sicherheit

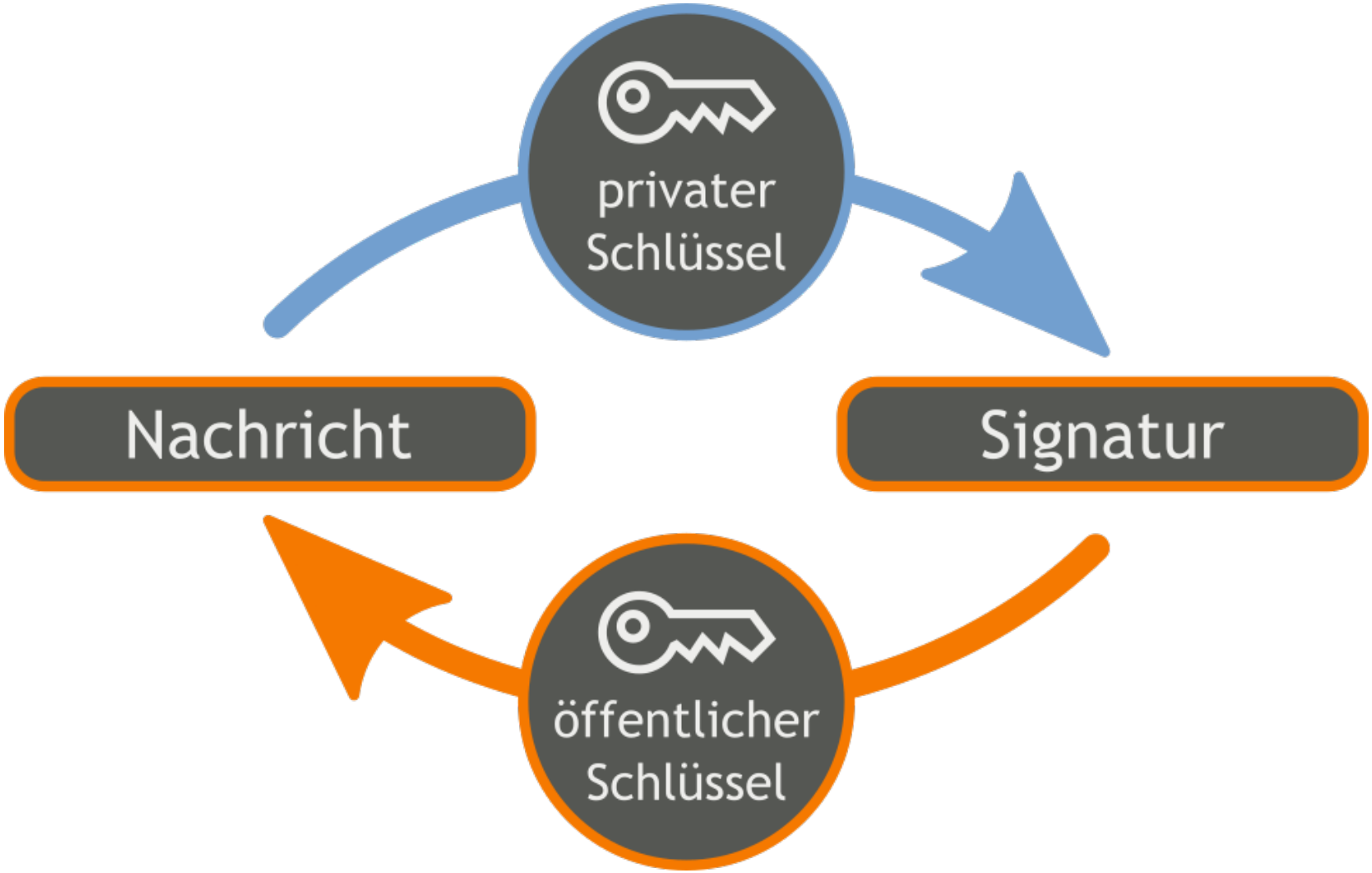
Cryptoparties

CC-BY 2.0 wbritzl

- Neue Leute kennenlernen
- Spaß an lustigen, alten, Ausweisbilder anderer haben!
- Irgendwas mit Sicherheit
- Selbst Cryptoparties veranstalten!

Signatur

„Nebeneffekt“ der Authentizitätsprüfung



GPG

Was ist jetzt eigentlich GPG?

- ,,Gnu Privacy Guard“, Open Source-Implementierung von OpenPGP
- ,,PGP“ (,,Pretty Good Privacy“), alternative, nun proprietäre Implementierung
- Sehr verbreitet im Open Source Umfeld

GPG

Was ist jetzt eigentlich GPG?

- ,,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

GPG

Was ist jetzt eigentlich GPG?

- ,,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!
- Teilweise Probleme mit Wrapping (z.B. Thunderbird)

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.]

***

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 unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`

GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
  - RSA oder DSA?

GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
  - RSA oder DSA?
  - Schlüssellänge?

GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
  - RSA oder DSA?
  - Schlüssellänge?
  - Gültigkeit?

GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
  - RSA oder DSA?
  - Schlüssellänge?
  - Gültigkeit?
  - Name (kein Pseudonym)

GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
  - RSA oder DSA?
  - Schlüssellänge?
  - Gültigkeit?
  - Name (kein Pseudonym)
  - Mailadresse

GPG unter Linux - Schlüsselpaar erstellen

- `$ gpg --gen-key`
  - RSA oder DSA?
  - Schlüssellänge?
  - Gültigkeit?
  - Name (kein Pseudonym)
  - Mailadresse
  - Passphrase (Langes_Passwort $>$ S4l4t)

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

- Laden des fremden Keys:

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

GPG unter Linux - Signieren

- Laden des fremden Keys:

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

- Prüfung des Fingerprints:

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

GPG unter Linux - Signieren

- Laden des fremden Keys:

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

- Prüfung des Fingerprints:

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

- Identitätsprüfung (Personalausweis, Führerschein)

GPG unter Linux - Signieren

- 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>
  ```

GPG unter Linux - Signieren

- 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

GPG unter Linux - Signieren

- 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 signiergen Keys per Mail **an die signierten Mailadressen**

GPG unter Linux - Signieren mit caff

- Automatisch:

```
$ caff <key-id>
```

GPG unter Linux - Signieren mit caff

- Automatisch:

```
$ caff <key-id>
```

- Benötigt konfigurierten SMTP-client
- Beispiel msmtp

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

GPG unter Linux - Empfängerseite

- Datei *signature.asc* per Mail bekommen
- Signatur importieren:

```
gpg --import signatur.asc
```

GPG unter Linux - Empfängerseite

- Datei *signature.asc* per Mail bekommen
- Signatur importieren:

  ```
  gpg --import signatur.asc
  ```

- Signatur Uploaden:

  ```
  gpg --send-keys <meine key-id>
  ```

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

GPG Keys Revoken

- Entweder automatisch (Gültigkeit) oder manuell
- Wichtig: **Vorher** Cross-signieren

GPG Keys Revoken

- Entweder automatisch (Gültigkeit) oder manuell
- Wichtig: **Vorher** Cross-signieren
- Revoken $\neq$ Löschen

GPG Keys Revoken

- Entweder automatisch (Gültigkeit) oder manuell
- Wichtig: **Vorher** Cross-signieren
- Revoken $\neq$ Löschen
- Widerrufszertifikat:

```
gpg --gen-revoke <key-id>
```

- Importieren, Uploaden

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!

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>`