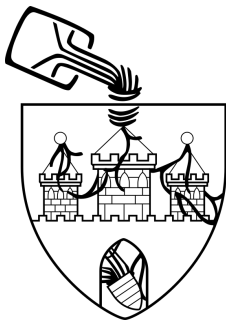


Sichere E-Mails mit PGP

Peter Gewalt & Manuel Groß

2015-11-15

Wer sind wir?



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



<https://mainframe.io/>

Was ist eigentlich unser Problem?



CC-BY-SA 2.0 Markus Winkler

Was ist eigentlich unser Problem?



CC-BY-SA 2.0 Markus Winkler

- 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

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

[illegible]

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

Lösungsvorschlag: Verschlüsselung

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

hQIMA9yhUHWLFXLAQ/+07en6i6BQQh9LEwQ2Zu/kFKXHN09GNGeXiPVtdGKMhUv
FJwFz7L6hrpWZJM8+YwWd7ny+nP4CHV5UjG0BLMLMQPVM40TnR9APa1Bf
qUrUs8UzQzsyQCC5ZEUjGnwKjRqfQV5sJj9970WxT0M20ZAdV1MQH54HqFRFB
OuQ9Q437KpW8nBX226VW/LSc600m2wHmKnmRn+Qb+SnmVaWn+Cp5Y0o8k7vv
qAdbUyRddVVUQzHnH+OPKiNo1AzXU0eqB0nVWML50qYiZiCyat51HfRu+LURUQz
nzdLIZyssDiss2rN/iVj8RcDrpTmikuKQATKXtLzIYrQh6+2NF37IztLjbIRkJ
GkTuZSMawJuRlpDzWriK3t+sXqF9dR0xHLKSE9sM3EPZCYomdTdBVt67iQ0mI
5K8Tx0dAck8Vh0Vrd5IBJy8LVHnSBuYCdV5jyxx+KMxfZxUP1X9ymOmB/jh06p
d/eLUGiWomWZtk7I8hVpVpnm8ByT+fbwLi33aUCCDa8jC7vC6FsA3kjgg7aDDl
r3nOCRL9LPibdyAMLiHDDdx0ae4wbU2Hn5aBFEiHBC0JPceI9euPrZnfZIM03ou3
IP6cZBLTavpHpz7NsZyBCAgKAXLQqXG2ptvVVL1Z96KhSxhZ/LQc0MdySoYUOF
AgwDywGdOQBy5bYBD/Ob5R7JBGt6aFNGMymPrQXt1gEU4n9n+FRcvXVp03eLcW1y
ZGJ8QjAaOSP0Qm0ECq+Q5CsQJMPzmXg08Kn7aKjzYQnDy6ZAAD2VEbSfZkue0kQ3
bIbfbYeANOALG80JTWsJSA2nPLXvD+E856j6xVZp0AmrJjFfiSu2/awzvw8giygH
nVB+2wELuxC18o11UdaPFketjEfwE00nAmYo2hT8WEgviNNT7pWl1ONHUBKXZF3
j8ZyD377MN04sR0i13f557L++zDjw6tCuDVQZjmeVihUjJbLNNbwKZAC/+Dv8nS
nJ9uZ9A9i2P0635SmZjseYwspHnsI+2nqswrtvuv2LYr9raX+PUYyG5nIAtD7CYr
e40mWdHfFbIPrjdp6p+jq6Gde0KCF0YJy+MLtCjOINasWfLYdU0cmR0ok0Z
MXWqL/Fz8XGQMR44bQZsVgQK9CXgnrXWMMQh/dZgB7B0hKVGSG/Fe5PwLkz
gqYCd5nSj3tT9e5MR4LdGtBwaIJW8TUfXhLOKQsF5FTS2U0eL5aZz+crkJKkh
U5+0RzI471U0QcCbCehuy16sZ9nNLvrgqtLcXvGvAmZSMqH0LUXN6dNn+ASu9w
pH0rKvvh7Qa/AGbc3oLZfo700B460TtXZbQxzR9tEhxtC0ixCsLr6HdCqTNLT
AbNU0onLuuG+0i7px0BtCgSUD0nIY7pPv0vExXPymPAE0iRLt2dgEqQ3xnLTA
E982fPxAWcMSJyUtiK1Hx1NTYzDdXRoNgVQ14JA6YFmf/vL/4dQGbN0vkhIyNgsB
8RqAgArhwkdvd/+aD9dxQcGZQd2tUg740mIj/yP13/m5VaWjUUAUcDG00086kW
EL6YwWfFKi+FWLUGjSCRNrsphXapi5S5GzxrqSNoH30gY0ip3/Y8LE/MLdCKYU5
yHrJmhJa4Clz/Z+zG0kiQvVt1j0MeCjLkYj3/HJLanMqn1PncycDvppnj9/jpM
8BQZdWj1jysyGrRyga1+tAmEtpC6Q1+d/zrT867LzI2qbl+fX0Kd3j6nqn8aGgQX
6BF0CBmBar10tZ1Mqvt0kU0HGWZervZCZ1bYBrK/LMJbHskOppGS0UG6VIAkL
tG0CBGDW0/BE9nanEejGtAuJWE4TtB0iRLCnayV69WfCDuMNDhuEL7PNaLQixuU
beEXuehYwW8/+yTfrd29gyZCU6478DunupGjcLvhmAPKxQJ5AnLECSiQk7HYva
4DhpP/BgjkQLI2BpB85W0cpD132nz6jAzSfxrNbF57Ddx9Fhh49Hdx5tV0Ig+WA
D9xU0e9t3M0cLjEYEv970MXJrdns6FdwP2YQ6fSUjY1L/4ISxYaU6ETZsJgAzz6X
Hxq4bHh+9+QMtUJTX5SFOxyTWosdrIK3i/LHARDcPjHQI03MSAN3AUScbMmPGaVB
uGf+LrNbg3gVCTJwK9nrmuPjfw+1NtL6ic3g0L1N00z0kAby0tztQyZeEpf4F
H9JwvChBLTLBjEn3zgtf1f18esV797gK/dvS9EVP9sDzm+CCxRpGjMLymSiAcP
Qv7wID8a8M6GwWwv8u+uRb8eYwF7F9uJwC3Dn+T9F4aWuLd4uG4iLDTw7W8hK

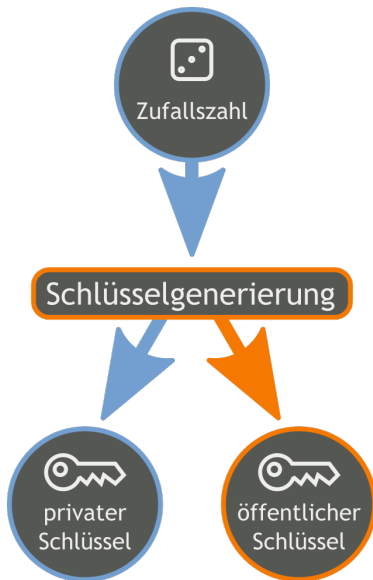
- Klartext → Geheimtext
- mathematische Verfahren
- geheime Zahl als „Passwort“
- symmetrisch
 - ▶ selbes Verfahren
 - ▶ selber Schlüssel

Abgrenzung: Verschlüsselung

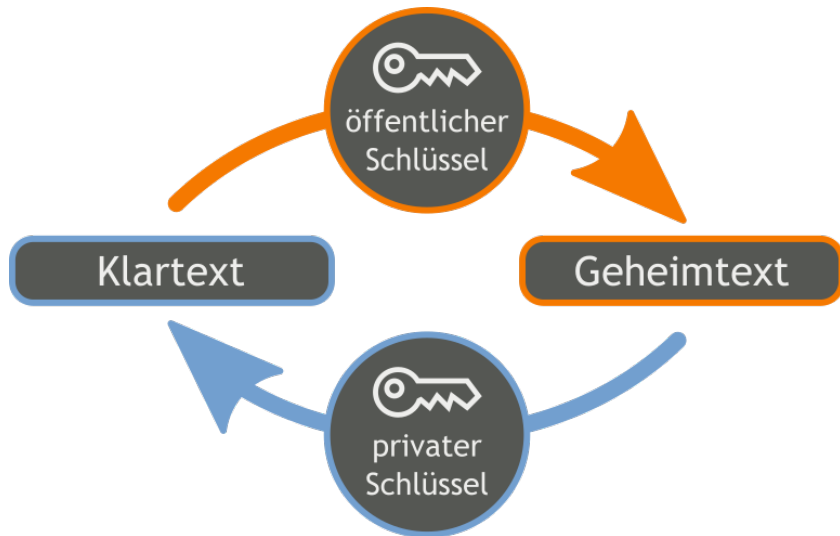
Was ist damit nicht gemeint?

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

Asymmetrische Verschlüsselung



Asymmetrische Verschlüsselung



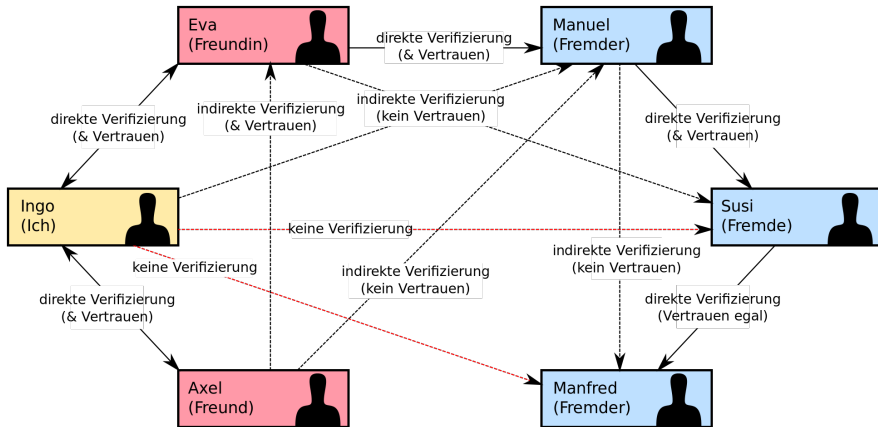
Problem: Vertrauenswürdigkeit

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



CC-BY-SA 3.0 Hauke Laging

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

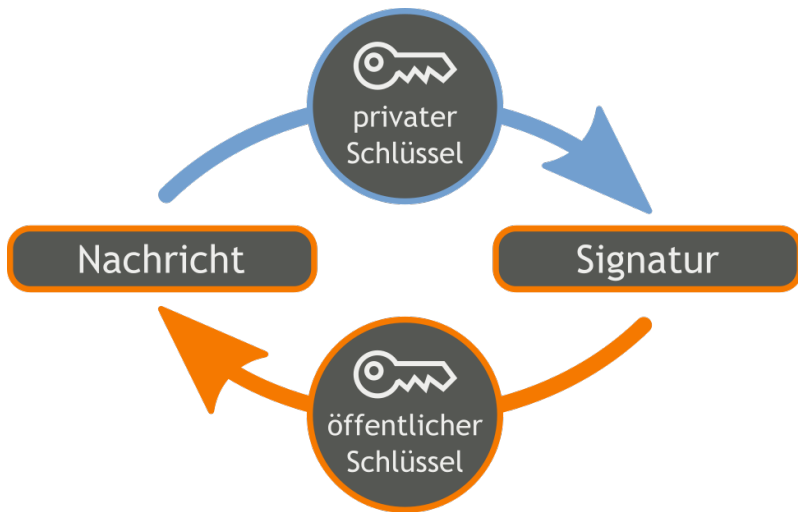


CC-BY 2.0 wbritzl

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

Signatur

„Nebeneffekt“ der Authentizitätsprüfung





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



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



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.

██████████@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.]

GPG unter Linux - Installation

Debian Pakete:

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

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 signierten 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 msmtpp

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

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
- Widerrufszeugnis:
`gpg --gen-revoke <key-id>`
- Importieren, Uploaden

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!

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>`