

Die Nutzung von GnuPG

Werner Koch

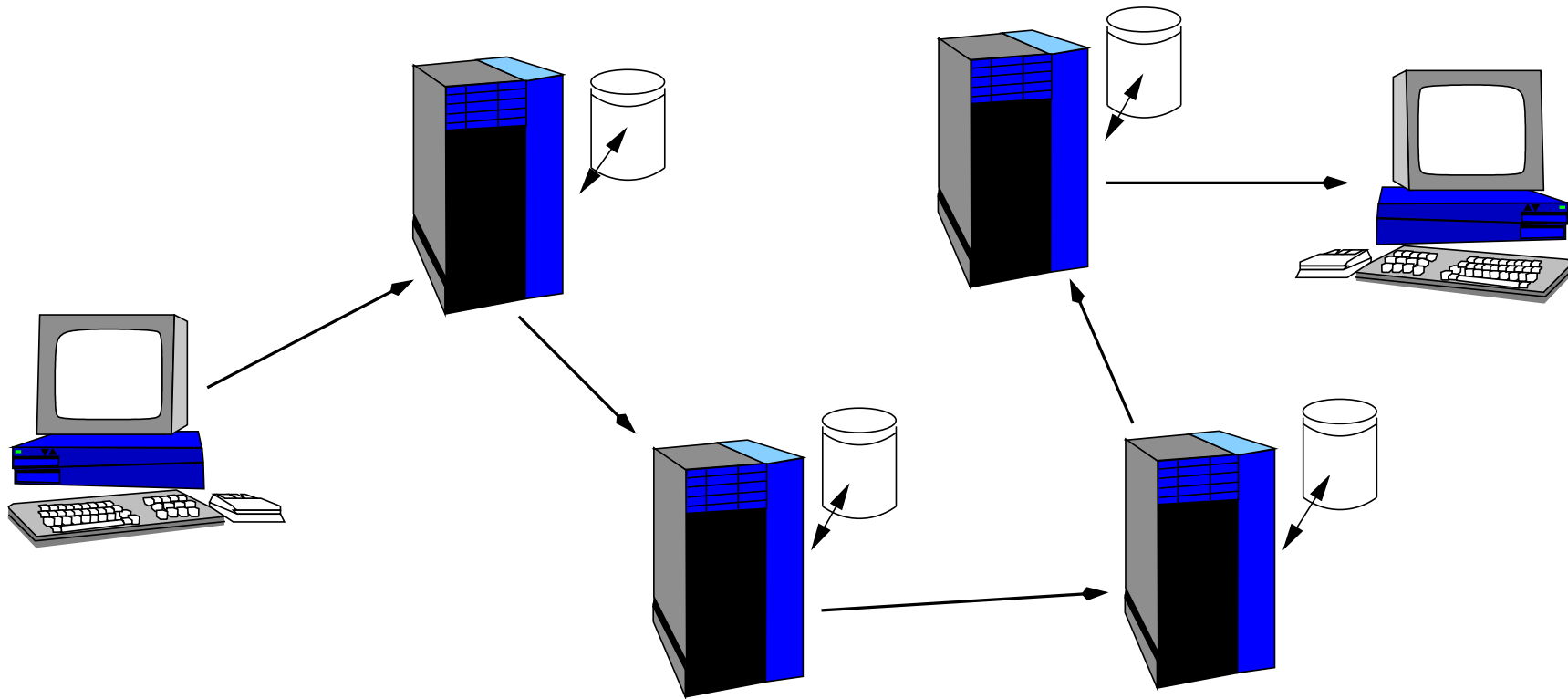
wk@gnupg.org

g10 Code GmbH

Warum Verschlüsseln

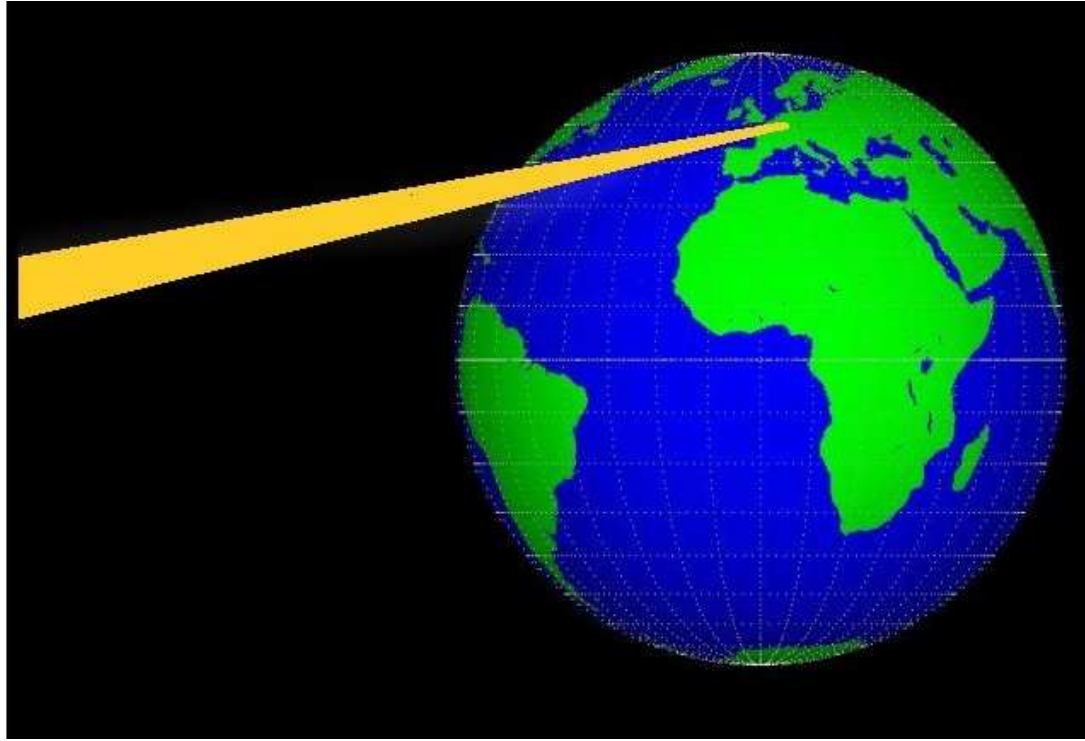
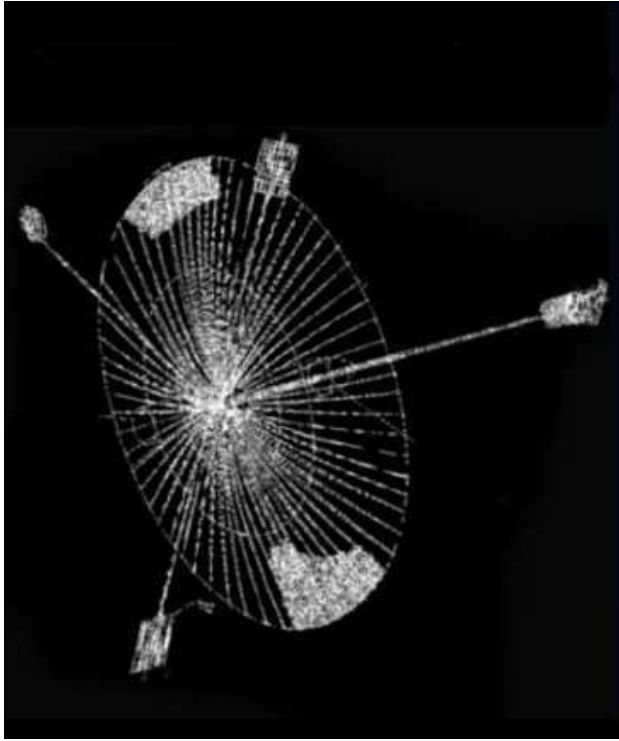
- Email wird im Klartext übertragen.
- Wird oft über viele Stationen transportiert.
- Dementsprechend viele Möglichkeiten zum Abhören.
- Einige Geheimdienste sind gehalten, der eigenen Wirtschaft Wettbewerbsvorteile zu schaffen.
- Abwendung von Schaden durch Wirtschaftsspionage.
- Aktienrecht
- Schutz der Privatsphäre.

Transportwege



Mail wird durch mehrere Rechner geleitet. An jeder Stelle ist ein Abhören möglich.

Richtfunk abhören



Weitere Abhörmöglichkeiten

- DNS Spoofing (MX Records)
- IP Spoofing
- Änderungen am Routing
 - BGP ist nicht geschützt
- Abhören direkt auf der Leitung
 - Zentrale Routingknoten
 - Echelon
 - Satellit
- Internes Abhören im LAN

Q entwickelt noch immer

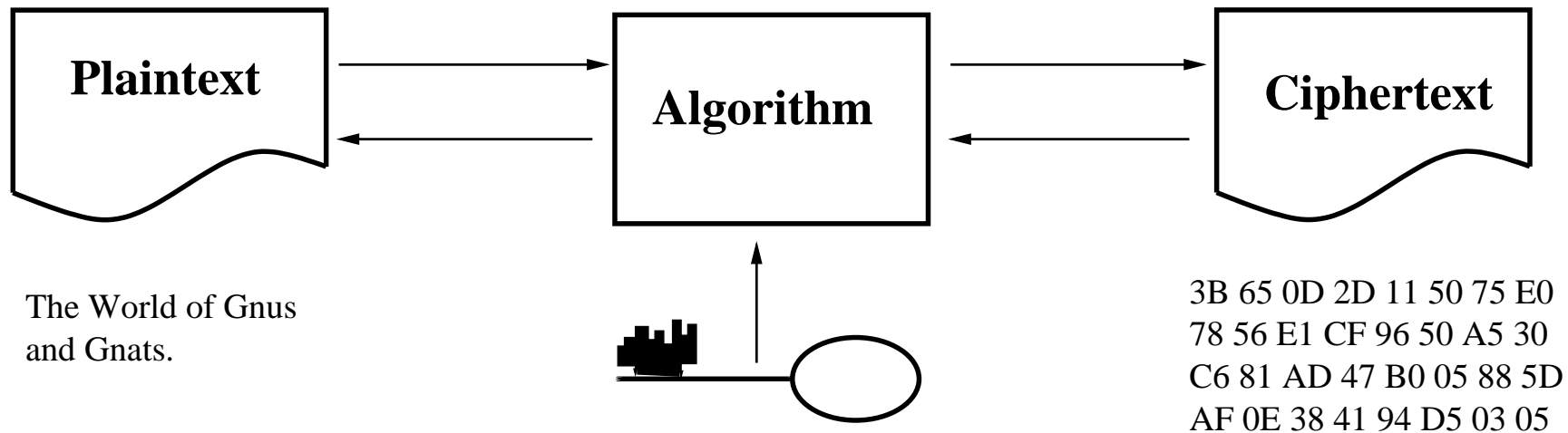
- Accepts up to 24 E3 inputs
- Improved flexibility and downloadability of algorithms and filters.
Allows users to customize algorithms
- Optional workstation software supports the identification and processing of signal variants, configuration of the Model 128 to process new signals, and displays for examining and analyzing signals in real time.
- TEMPEST design



Verschlüsselungsverfahren

Symmetrische Verschlüsselung

Die klassische Verschlüsselung basierend auf einem Schlüssel der beiden Parteien bekannt ist.



Beispiele: Triple-DES, AES

Symmetrische Verschlüsselung (2)

Vorteile:

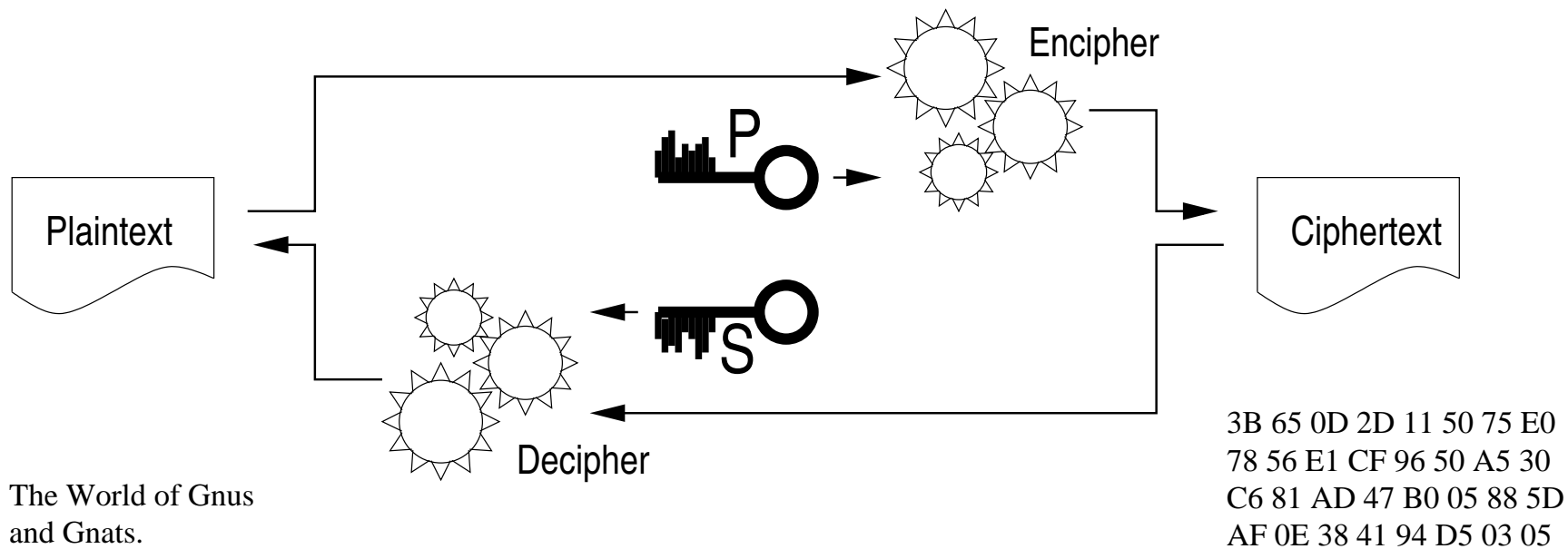
- Verfahren sind seit Jahrtausenden bekannt
- Heute sehr sicher und schnell.

Nachteile:

- Ein Schlüssel für jeden Kommunikationskanal.
- Schlüsselverwaltung ist umständlich.

Asymmetrische Verschlüsselung

Public-Key Verfahren basierend auf geteilten Schlüsseln.



Beispiele: RSA, ElGamal

Exkurs: RSA

Wähle 2 geheime und große Primzahlen: p, q .

Wähle eine beliebige Zahl e , teilerfremd zu $(p - 1)(q - 1)$.

Berechne $n = p * q$ sowie $d = e^{-1} \bmod (p - 1)(q - 1)$.

n, e ist der öffentliche Schlüssel,
 d der private Schlüssel.

Verschlüsseln: $(ciphertext) = (klartext)^e \bmod n$

Entschlüsseln: $(klartext) = (ciphertext)^d \bmod n$

Asymmetrische Verschlüsselung (2)

Vorteile:

- Nur eine Schlüsselpaar pro Kommunikationspartner.
- Nur ein Teil des Schlüssels muß geheim gehalten werden.
- Schlüsselverwaltung wird einfacher.

Nachteil:

- Bestimmung der Authentizität eines Schlüssels.

Digitale Signaturen

- Asymmetrische Verfahren können auch zur Erzeugung einer digitalen Signatur benutzt werden.
- Hier wird auf einen Text der *private Schlüssel* angewandt
- und mit dem *öffentlichen Schlüssel* kann dann jederzeit bewiesen werden, daß nur der Besitzer des privaten Schlüssels diese Signatur erzeugt haben kann.
- Bekannte Algorithmen: RSA, DSA

Digitale Signaturen dienen auch zur Bestimmung der Authentizität von öffentlichen Schlüsseln (Zertifikate).

GNU Privacy Guard

Was ist GnuPG

- Verschlüsselt und signiert Email und andere Daten.
- Vollständige Implementation des OpenPGP Standards.
- Verbindet digitale Signaturen, Verschlüsselung und Schlüsselverwaltung in einer Anwendung.
- Läuft auf allen POSIX Plattformen sowie auf Windows und Mac.
- Flexibel und lange im praktischen Einsatz.
- Verfügbar unter der GNU General Public License (GPL).

1991 PGP wird veröffentlicht — Patentprobleme.

1997 DH Patent verfällt.

Herbst 1997 Start der Entwicklung einer patentfreien PGP ähnlichen Software. Etwa zeitgleich gründet die IETF die OpenPGP WG.

Dezember 1997 Erste GnuPG Version unter dem Arbeitstitel “g10” veröffentlicht.

November 1998 RFC2440 als Spezifikation von OpenPGP erschienen.

September 1999 : GnuPG 1.0 veröffentlicht.

Rückblick (2)

1999/2000 Förderung der Portierung von GnuPG auf Windows durch das BMWi. Weltweit erste Förderung eines solchen Projekts.

Dezember 2000 Aufhebung der U.S. Exportrestriktionen.

2002 Beauftragung durch das BSI zur zusätzlichen Unterstützung von S/MIME.

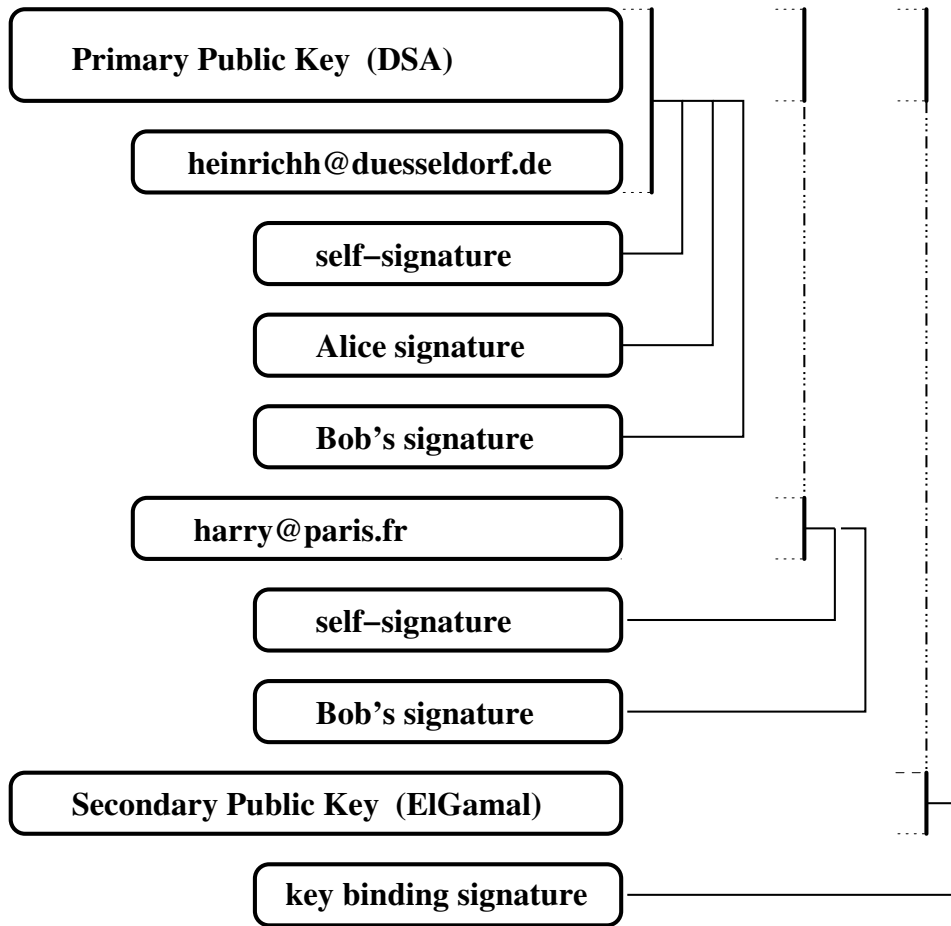
2004 GnuPG 1.4 mit Unterstützung von Smartcards.

2005 Entwicklung von Freenigma; basierend auf GnuPG.

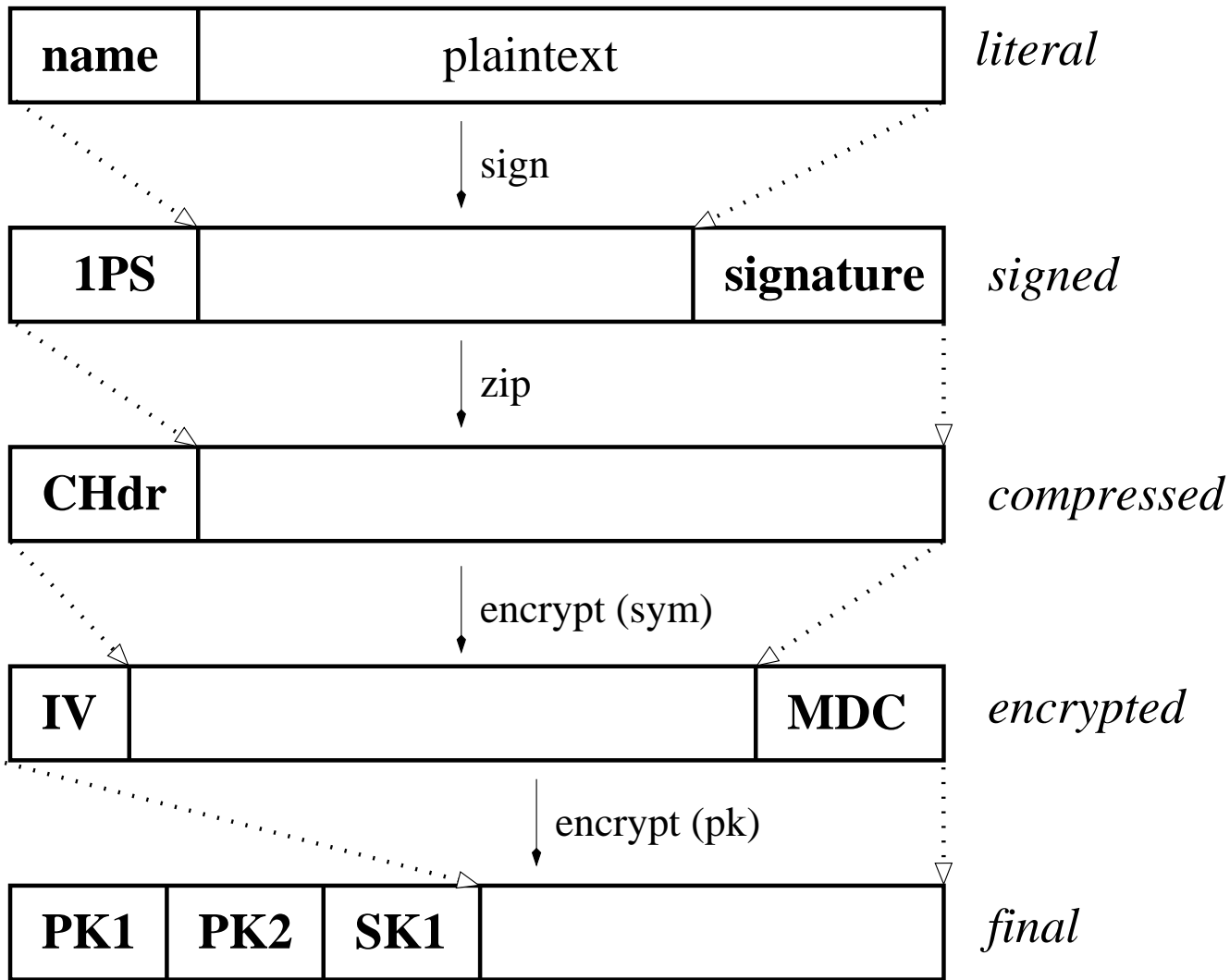
Einige Anwender und Anwendungsgebiete von GnuPG:

- Email
- DNS Root Zonenverwaltung
- RIPE und verschiedene NICs
- Industrie
- Deutsche Bahn
- DIMDI
- Radiologiedaten
- Transport von Kreditkartendaten

OpenPGP Schlüsselformat

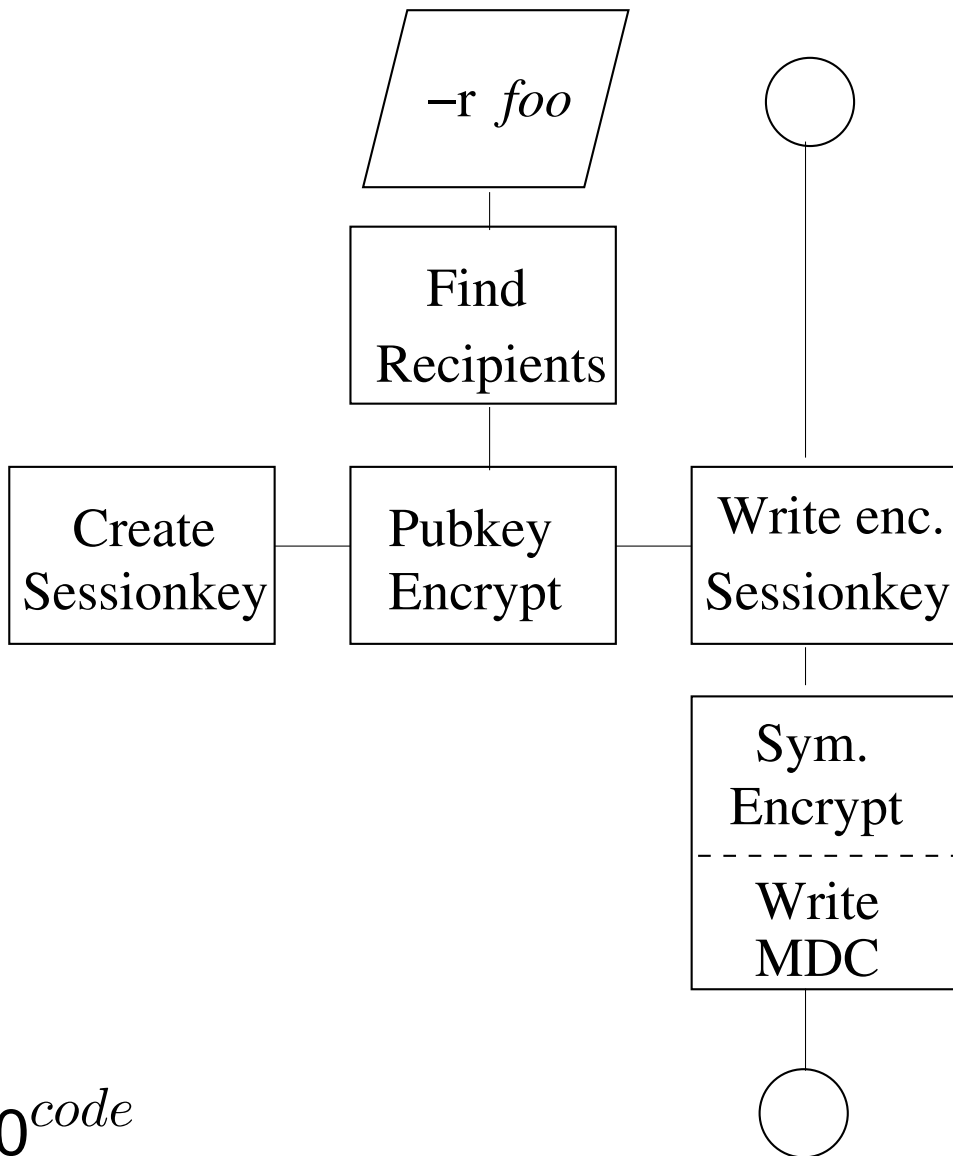


OpenPGP Nachrichtenformat



g10^{code}

Ablauf der Verschlüsselung



Anwendungsbeispiele

- Erzeugen eines Schlüsselpaars:

```
gpg --gen-key
```

```
gpg --gen-revoke your_user_id
```

- Verschlüsseln eine Datei `file` an `heine`

```
gpg -e -r heine file
```

- `gpg` ist ein Unix Werkzeug also geht auch:

```
awk -F: '{print $2,$1}' /etc/shadow | \  
gpg -ea -r Schily | mail schily@bmi.de
```

PKA

PKI Probleme

- PKI = Public Key Infrastructure??
 - Standard? Welcher X.509 Profile?
 - Inkompatible Implementationen.
 - Nur einsetzbar innerhalb einer Organisation
 - Flickwerk
- Web of Trust?
 - Andere Implementierung eines PKI
 - Zu kompliziert für Otto Müller

PKI Probleme (2)

- SSH Modell?
 - Einfach, funktioniert
 - Für Admins leicht zu verstehen
 - Zu kompliziert für Renate Mustermann

Fazit: Für alltägliche Aufgaben gibt es nichts ...

PKA anstatt PKI

PKA = Public Key Association

- DNS ist überall vorhanden
- DNS wird benutzt
- DNS wird *irgendwann* einmal sicher sein.

Also: **Schlüssel mit DNS verbinden**

PKA anstatt PKI (2)

- Vorteile:
 - Einfach
 - Flexibel
 - Kann zentral gemacht werden (MTA)
 - Kann für einzelne Schlüssel gemacht werden

Implementierung momentan via TXT RRs:

```
$ host -t txt werner._pka.fsfe.org
werner._pka.fsfe.org text \
    "v=pka1;fpr=A4 [...]58A2;uri=finger:wk@g10c
```

Vielen Dank für Ihre Aufmerksamkeit.

URLs

- <http://www.gnupg.org/>
- <ftp://ftp.gnupg.org/gcrypt/>
- <http://www.guug.de/>