

Daten- und Emailverschlüsselung

Eine Einführung in GnuPG

10. Dezember 2007

Florian Streibelt <florian@freitagsrunde.org>

GG, Art. 10, Abs. 1:
Das Briefgeheimnis
sowie das Post- und
Fernmeldegeheimnis
sind unverletzlich (?)

Ziele der Verschlüsselung

- **Geheimhaltung**

Nachricht soll nicht durch Dritte entziffert werden können

- **Echtheit des Kommunikationspartners**

Stammt die Nachricht wirklich von dem, der er vorgibt zu sein?

- **Integrität der Nachricht**

Ist die Nachricht so verfasst worden, wie sie angekommen ist?

- ▶ **Vertrauen in die Kommunikation**

→ Ist erreicht, falls alle obigen Punkte eingehalten werden.

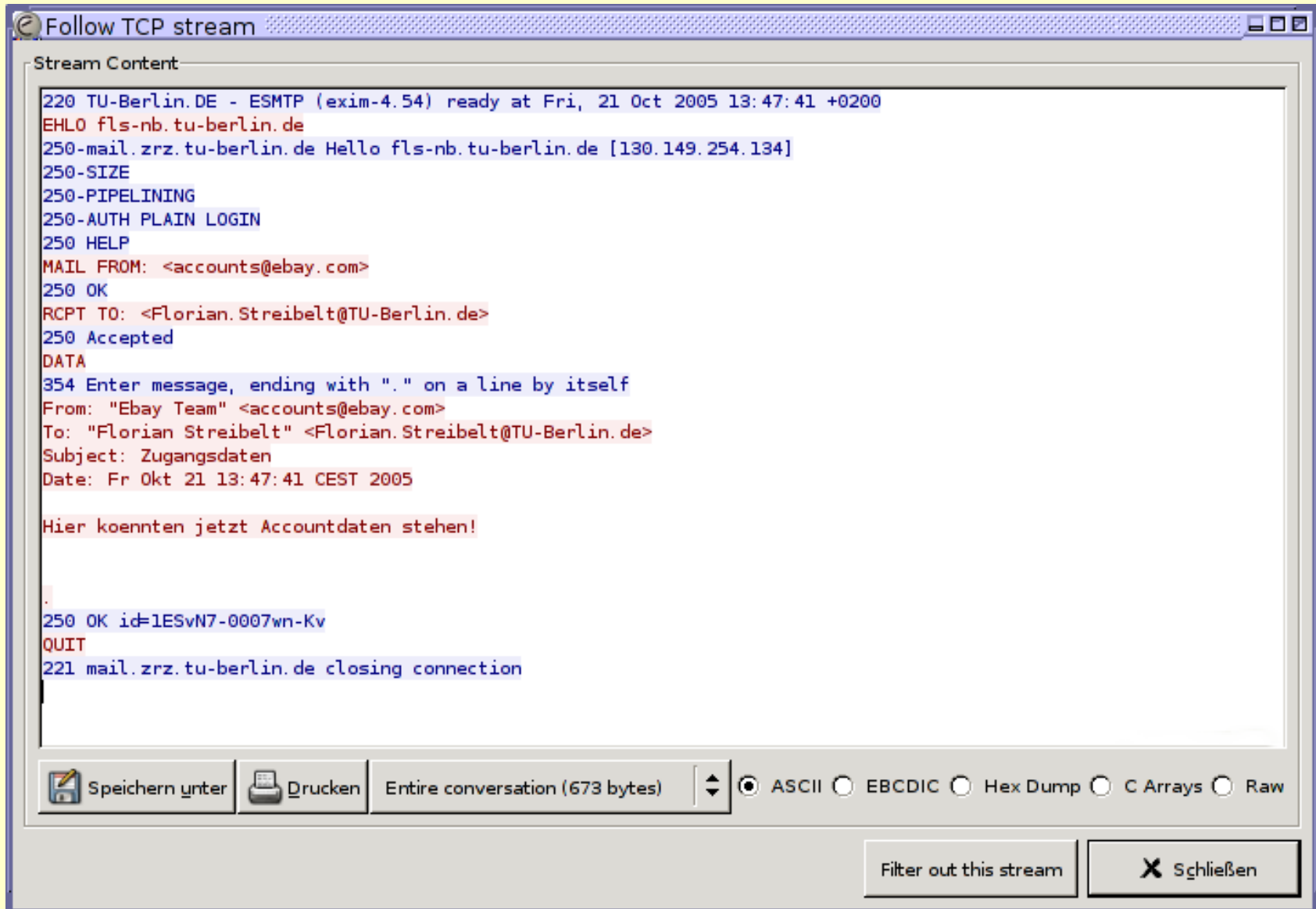
Bestandsaufnahme

- Email-Verkehr kann sehr einfach abgehört und manipuliert werden (ECHELON, TKÜV)
- ein sehr hoher Prozentsatz der Firmenkommunikation erfolgt dennoch per Email!
- interne Daten werden damit per „elektronischer Postkarte“ versandt
- im Rahmen der „Terrorismusbekämpfung“ ist die Vorratsdatenspeicherung eingeführt
- Emails werden zu Ermittlungszwecken gesichert / ausgelesen

Emails mitschneiden

- mailsnarf:
 - extrahiert Emails aus Netzwertraffic
 - speichert sie im mbox-Forma
- msgsnarf:
 - extrahiert Instant-Messenger-Nachrichten
- wireshark:
 - liest kompletten Netzwerkverkehr

Beispiel: wireshark



The screenshot shows the 'Follow TCP stream' window in Wireshark. The window title is 'Follow TCP stream'. The main content area displays the following text:

```
220 TU-Berlin.DE - ESMTP (exim-4.54) ready at Fri, 21 Oct 2005 13:47:41 +0200
EHLO fls-nb.tu-berlin.de
250-mail.zrz.tu-berlin.de Hello fls-nb.tu-berlin.de [130.149.254.134]
250-SIZE
250-PIPELINING
250-AUTH PLAIN LOGIN
250 HELP
MAIL FROM: <accounts@ebay.com>
250 OK
RCPT TO: <Florian.Streibelt@TU-Berlin.de>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
From: "Ebay Team" <accounts@ebay.com>
To: "Florian Streibelt" <Florian.Streibelt@TU-Berlin.de>
Subject: Zugangsdaten
Date: Fr Okt 21 13:47:41 CEST 2005

Hier koennten jetzt Accountdaten stehen!

.
250 OK id=1ESvN7-0007wn-Kv
QUIT
221 mail.zrz.tu-berlin.de closing connection
```

At the bottom of the window, there are several controls:

- Buttons: 'Speichern unter' (Save as), 'Drucken' (Print).
- Text: 'Entire conversation (673 bytes)'.
- Radio buttons for encoding: 'ASCII' (selected), 'EBCDIC', 'Hex Dump', 'C Arrays', 'Raw'.
- Buttons: 'Filter out this stream', 'Schließen' (Close).

Die Lösung?

Kryptographie

= Wissenschaft von der Verschlüsselung und Verschleierung

Kryptoanalyse

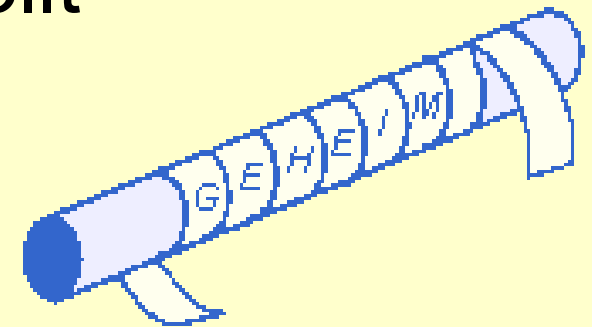
= Wissenschaft von der Entschlüsselung

Wie ist Kryptographie entstanden?

- die Ursprünge reichen sehr weit zurück
- zunächst wurden die Daten „nur“ versteckt
 - ein geheimer Bote überbringt die Nachricht
- später wurden die Daten verschlüsselt
 - Auch der Bote kann die Nachricht nun nicht mehr lesen oder manipulieren
 - Bei Gefangennahme eines Boten bleibt die Nachricht geheim

Skytala von Sparta

- älteste bekannte Verschlüsselungsmethode (ca. 2500 Jahre)
- von Plutarch überliefert
- ein Band wird um einen Zylinder gelegt, quer beschrieben und wieder abgerollt
- der Schlüssel besteht aus dem Zylinderdurchmesser



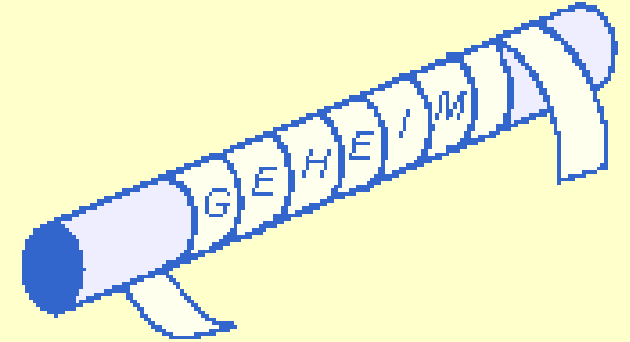
Transpositionschiffre:

Die Anordnung der Zeichen wird vertauscht

Beispiel Skytala

- Geheimtext:

FTRETRAUDUEGNEBISDRX



- Test mit „Durchmesser 5cm“:

FTDNSTRUEDRAEBREUGIX

- Test mit „Durchmesser 4cm“:

FREITAGSRUNDEDERTUBX

Cäsarchiffre

- von Julius Cäsar (100-44 v.Chr.) erfunden
- Buchstaben werden um einige Stellen verschoben
- Klasse der Verschiebechiffren
- ROT13 als „moderne“ Variante
- Problem:
Buchstabenhäufigkeiten bleiben erhalten
- Schlüssel kann aus Chiffre und Klartext wieder extrahiert werden

Beispiel Cäsarchiffre

- Klartextalphabet:

a b c d e f g h i j k l m n o p q r s t u v w x y z

- Geheimtextalphabet:

d e f g h i j k l m n o p q r s t u v w x y z a b c

- Klartext:

FREITAGSRUNDEDERTUB [3x 'E']

- Geheimtext:

IUHLWDJVUXQGHHUWXE [3x 'H']

One-Time-Pad

- einzige beweisbar sichere Verschlüsselung
- Voraussetzungen:
 - Schlüsselänge gleich Klartextlänge
 - Schlüsselerzeugung **streng** zufällig
 - Schlüssel wird nur **einmal** verwendet
- hoher Aufwand → militärische Umgebungen
- Probleme:
 - menschlicher Faktor
 - Beschaffung neuer OneTimePads => Henne-Ei

Symmetrische vs. Asymmetrische Verschlüsselung

- Nachteile aller bisherigen Methoden:
 - Zur Kommunikation über unsichere Kanäle (Internet) muss der Schlüssel zuerst über einen sicheren Kanal übertragen werden.
 - ein Schlüssel pro Kommunikationspartner
- Ausweg: asymmetrische Verschlüsselung
 - Es gibt einen geheimen und einen öffentlichen Schlüssel.
 - Der öffentliche Schlüssel dient zum Verschlüsseln von Daten, nur der Geheime kann die Originaldaten wieder herstellen. (idealerweise)

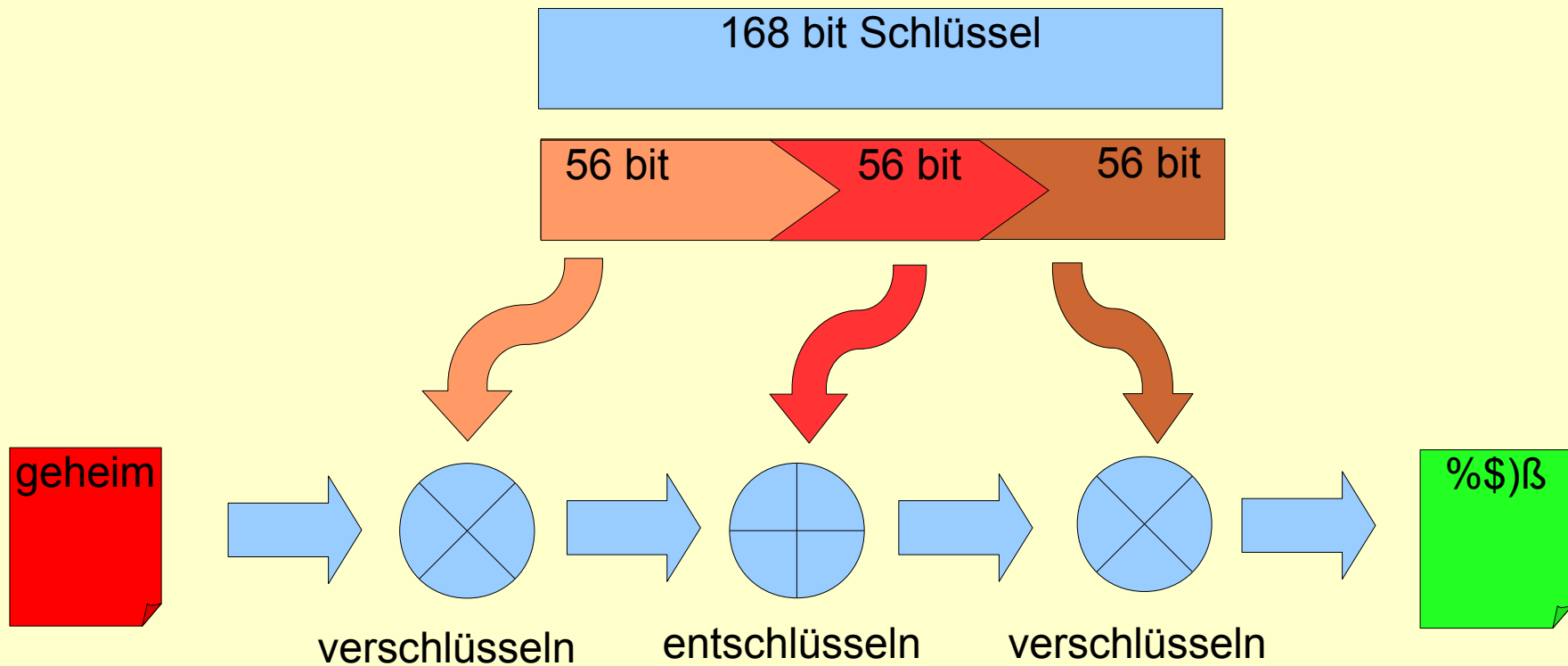
RSA

- Das Prinzip wurde von Whitfield Diffie und Martin Hellman 1976 entwickelt und publiziert
- Ron **R**ivest, Adi **S**hamir und Leonard **A**dleman wollten dessen Sicherheit widerlegen
- ... und erfanden dabei RSA
- ältester und angesehenster asymmetrischer Algorithmus, löste als erster das Schlüsselverteilungsproblem
- basiert auf der Faktorisierung großer Zahlen

DES und 3DES

- **D**ata **E**ncryption **S**tandard von 1977
- symmetrische Blockchiffre mit 56 Bit
- im kommerziellen Bereich am häufigsten eingesetzt
- u.a. 1999 in ca. 22 Stunden gebrochen
- Weiterentwicklung: 3DES, benutzt 3 unterschiedliche DES Schlüssel hintereinander: 168 Bit
- wird bei einigen Smartcards verwendet.

3DES-Verfahren



AES

- Advanced Encryption Standard
- Nachfolger von DES, 3DES
- Industriestandard zur symmetrischen Verschlüsselung
- Blocklänge: 128 Bit
- Schlüssellänge: 128, 192 oder 256 Bit
- frei verfügbar, keine Lizenzgebühren
- Standard bei Luks-Containern

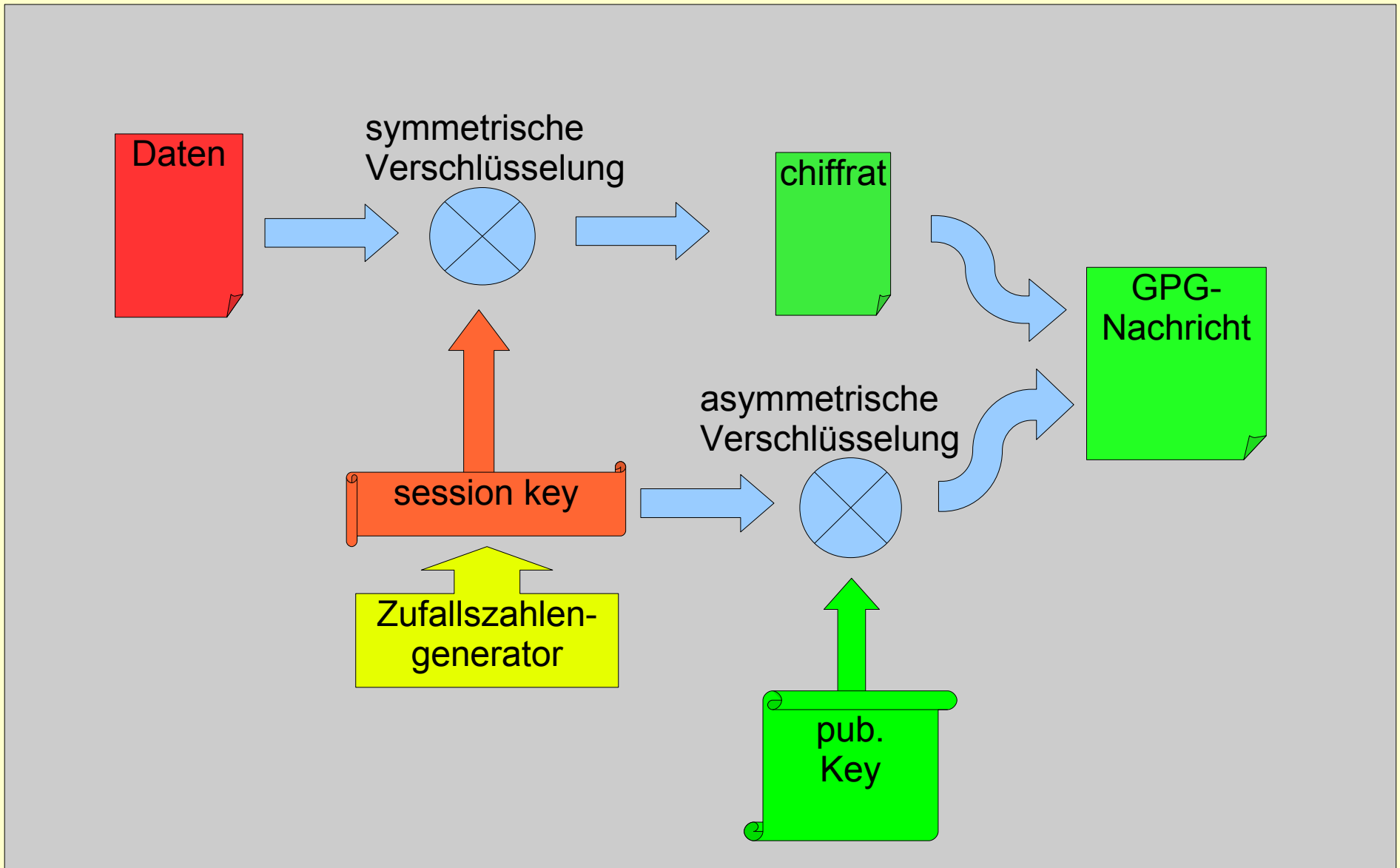
Symmetrische vs. Asymmetrische Verschlüsselung (2)

- Nachteile symmetrischer Verschlüsselung:
 - shared secret
 - Schlüsselverteilungsproblem
- Vorteil:
 - sehr schnell
- Nachteil asymmetrischer Verschlüsselung:
 - bei großen Datenmengen sehr langsam
- Vorteil:
 - kein Schlüsselverteilungsproblem

Funktionsweise von GnuPG

- hybrides Verfahren, kombiniert geschickt symmetrische und asymmetrische Algorithmen:
 - ein zufälliger symmetrischer Schlüssel wird zur Verschlüsselung benutzt („session key“)
 - dieser wird asymmetrisch verschlüsselt und mit der Nachricht verschickt
 - geheimer Schlüssel entschlüsselt den „session key“, damit kann der Empfänger die Daten lesen
- man benötigt nur den öffentlichen Schlüssel des Kommunikationspartners
- der geheime Schlüssel verlässt nie den Rechner

Schema GnuPG



Geschichte (auch) von GnuPG

- PGP (Pretty Good Privacy) 1991 von Phil Zimmermann, zunächst Open Source, später (1997) Verkauf an NAI
- div. Inkompatibilitäten der PGP-Versionen führten 1998 zu RFC 2440
- seit 1998 Entwicklung von GnuPG (Gnu Privacy Guard), Förderung durch BMBF
- aktuelle Version von GnuPG: 1.4.6
- seit geraumer Zeit: Chipkartenunterstützung
- Open Source, diverse Portierungen

Praktische Einführung in GnuPG

Auf den folgenden Folien wird gezeigt:

- erzeugen eines eigenen Schlüssels
- erzeugen eines Widerrufszeugnisses
- Verschlüsseln einer Datei
- Entschlüsseln einer Datei

Alle Beispiele beziehen sich auf die Kommandozeile, da hier die größte Kontrolle über GnuPG ausgeübt werden kann, am Ende werden noch einige GUI kurz vorgestellt.

Schlüssel erzeugen

Wahl des Signatur- und/oder Verschlüsselungsalgorithmus

```
fls@fls-nb:~$ gpg --gen-key
```

```
Bitte wählen Sie, welche Art von Schlüssel Sie  
möchten:
```

- (1) DSA and Elgamal (default)
- (2) DSA (nur signieren/beglaubigen)
- (5) RSA (nur signieren/beglaubigen)

DSA: Digital Signature Algorithm,
zum Signieren von Dateien (nicht DES!)

ElGamal: Verschlüsselungsalgorithmus von El Gamal
(Israel), ähnlich RSA

RSA: Rivest, Shamir, Adleman,
wie bereits vorgestellt

Schlüssel erzeugen

Wahl der Schlüssellänge

...

DSA keypair will have 1024 bits.

ELG-E keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

Requested keysize is 2048 bits

- Es gibt Anzeichen, dass 1024 Bit nicht mehr ausreichend sicher sind. Deshalb sollte eine Schlüssellänge von mindestens 2048 bit gewählt werden.
- Das Arbeiten mit größeren Schlüsseln wird schnell sehr langsam...
- Hier ist wie immer zwischen Sicherheit und Komfort abzuwägen.

Schlüssel erzeugen

Wahl der Gültigkeitsdauer

...

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0)

- Gültigkeitsdauer hängt von eigenen Präferenzen ab.
- Oftmals wird ein Hauptschlüssel ohne feste Gültigkeit generiert und dann jeweils Unterschlüssel mit einjähriger Laufzeit.
- Vorteil: Bei Verlust der Passphrase und des Revocation Zertifikates verfällt der Schlüssel automatisch
- Partner werden gezwungen, „öfter“ auf Keyservern neu zu suchen

Schlüssel erzeugen

Name, Kommentar, E-Mail-Adresse und Passphrase

...

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:

```
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

```
Real name: Florian Streibelt
```

```
Email address: florian@freitagsrunde.org
```

```
Comment: Freitagsrunde der TUB
```

```
You selected this USER-ID:
```

```
"Florian Streibelt (Freitagsrunde der TUB) <florian@freitagsrunde.org>"
```

- Eingabe von Namen und E-Mail-Adresse und Bestätigung der Angaben
- Eingabe der Passphrase
- [Passphrase] kein „Wort“ aus dem Wörterbuch, möglichst ein verfremdeter Satz, z.B. „Ich habe den Linuxtag Chemnitz besucht.“
→ „1(h 4abe d3n L!nv)(tag Ch0mn:tz b?such+.“

Schlüssel

...

```
gpg: key F435050F marked as ultimately trusted
public and secret key created and signed.
```

```
gpg: checking the trustdb
```

```
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
```

```
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
```

```
pub 1024D/F435050F 2005-11-06
```

```
    Key fingerprint = B625 3451 AF4E F755 E407 FD88 A92B 359F F435 050F
```

```
uid                               Florian Streibelt (Freitagsrunde der TUB)
```

```
<florian@freitagsrunde.org>
```

```
sub 2048g/6BB5B4FA 2005-11-06
```

- Anzeige des Fingerprints am Ende der Schlüsselerzeugung
- Der Fingerprint ist eine Prüfsumme, mit der die Integrität des Schlüssels geprüft werden kann.
- Die letzten 8 Byte sind die Key-ID, wegen Kollisionen werden mittlerweile meist die letzten 16 Byte benutzt.

Widerrufszertifikat erzeugen

```
fls@fls-nb:~$ gpg --gen-revoke F435050F
```

```
sec 1024D/F435050F 2005-11-06 Florian Streibelt (Freitagsrunde der TUB)  
<florian@freitagsrunde.org>
```

```
Create a revocation certificate for this key? (y/N)
```

```
Please select the reason for the revocation:
```

- 0 = No reason specified
- 1 = Key has been compromised
- 2 = Key is superseded
- 3 = Key is no longer used
- Q = Cancel

```
(Probably you want to select 1 here)
```

```
Your decision?
```

- Vorbeugung vor kompromittierten Schlüsseln und vergessenen Passwörtern
- Widerrufszertifikat ausdrucken und an sicherem Ort aufbewahren, möglichst nicht mit Arial: II = iL, besser Times, Comic, etc.

Widerrufszertifikat

```
-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: GnuPG v1.4.1 (GNU/Linux)  
Comment: A revocation certificate should follow  
  
iFwEIBECABwFAkNuZYQVHQBQYXNzcGhyYXNlIHZlcmxvcmVuAAoJEKkrNZ/0NQUP  
mUsAniYuZuEp+Vrvhgrj/8KTTcBE0x4bAJ0YYyczU7zQlcJd7YsX00qGt12NSA==  
=F8X3  
-----END PGP PUBLIC KEY BLOCK-----
```

- Vorsicht beim Ausdruck: Das Drucksystem könnte die Datei speichern, eventuell sogar der (Netzwerk-) Drucker selbst
- Jeder mit Zugriff auf diese „Zeichenkette“ kann den Schlüssel unbrauchbar machen, indem er das Revocation-Zertifikat auf einem Keyserver veröffentlicht.

Beispiel: Verschlüsseln

```
$ echo "Geheime Mitteilung" > geheim.txt
$ gpg -aer F435050F geheim.txt
$ ls -l geheim*
-rw-r--r--  1 fls users  19 Nov  6 21:25 geheim.txt
-rw-r--r--  1 fls users 932 Nov  6 21:25 geheim.txt.asc

$ cat geheim.txt.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.1 (GNU/Linux)

hQIOA4C2E61rtbT6Eaf+PXrzs5CwHhRXsb3UK+5wgnV4eLYUDv0dVag5akMj+nvU
ekEBb0ucjZN20HKB9wgKAXo3+HmkiG4IYE1Qm5d2wp6He6wRUo2DoxZR1Gmfczoy
sQFC/fGWaq/wclgKiyEyYCECao0R6Y8PrNS0Zyybq+5XJOH4n8ybDJFTo6FqIYfb
uTZTCxtRqpB7T5MUMt0hUJTX3LNF46Qt1OxNXhmeYM9pd5AJMmXkPbwSw1CsHvdG
gbGajMmoewYswSD5WMeB66dxtIQQxzFQNRMDn/9i6Kpfj0rh/WBf/3NrPsoy04y6
l7lpR/eeeGyBooiY2ti0/rNimXipW3zh3bgejnpzAAgAmFwOyE5aJle1BVQ5EFxW
et1VutRo7hVjAimWNPxDK8pJUDxJ2P0h9Kbe/Tq6tV+yp5viwnI3Ec5F3YqWpVaC
ebdlgH7mxAvsz4266hGFZsCImvl9UqBeKd7flZxhYIpG/+omgTDWyve8gA5kUZg3
O/v7ic2xHeDsHoxOYvol9cGteeIgj+eBBCsZT79Nhy5ZoCvrouBaueLh3GNDInrs
nFvjGKAlg8BMPRofdD6BkGix+pp5pLKdOrVfVMgqdlIEHpoh0KXz09l4q5Bux+EC
bMltOcTX0IrlrxAVqqE0qqKCLLiHZ8TJlfh6polmOTryQAwimP3b8uNiB9NeT7sW
H9JVAVCnf3FUMmac1TR4/JmfPNjI41k4yQ84N5fJmcgbbmjgvLk75RhpEnLqaxrG
PnDx6jhgqQ+zUSznwrg67KfqIqCLUzPKnTTIxc6dlrzoWVhKAEiTEBw==
=xXPN
-----END PGP MESSAGE-----
```

Beispiel: Entschlüsseln

```
$ gpg < geheim.txt.asc
```

```
You need a passphrase to unlock the secret key for  
user: "Florian Streibelt (Freitagsrunde der TUB) <florian@freitagsrunde.org>"  
2048-bit ELG-E key, ID 6BB5B4FA, created 2005-11-06 (main key ID F435050F)
```

```
gpg: encrypted with 2048-bit ELG-E key, ID 6BB5B4FA, created 2005-11-06  
"Florian Streibelt (Freitagsrunde der TUB) <florian@freitagsrunde.org>"
```

Geheime Mitteilung

alternativ, direkt in eine Datei:

```
fls@fls-nb:/tmp$ gpg geheim.txt.asc
```

```
You need a passphrase to unlock the secret key for  
user: "Florian Streibelt (Freitagsrunde der TUB) <florian@freitagsrunde.org>"  
2048-bit ELG-E key, ID 6BB5B4FA, created 2005-11-06 (main key ID F435050F)
```

```
gpg: encrypted with 2048-bit ELG-E key, ID 6BB5B4FA, created 2005-11-06  
"Florian Streibelt (Freitagsrunde der TUB) <florian@freitagsrunde.org>"
```

Features in GnuPG

- Chipkartenunterstützung: OpenPGP-Card
- Foto-IDs:
 - eingebettete jpeg-Dateien
 - Vorsicht: bitte kleine Dateien benutzen
- mehrere uid's Pro Schlüssel
 - pro Emailadresse eine User-ID
- mehrere Schlüssel
 - werden alle zusammen signiert
 - z.B. für unsichere Umgebungen

Bis hier Fragen?

'Sicherheitshinweise'

- Emails auch lokal nur verschlüsselt speichern
- wirklich wichtige Dinge auf einem getrennten Rechner ohne Internet machen
 - Stichwort: (Quellen-TKÜV)
- GPG weiter promoten
- Festplattenverschlüsselung
- nur signierte Updates installieren
- Chipkarte nutzen – mit „Level 3“ Terminal

Aufbau des Web of Trust

- Das Problem:
Jeder kann Schlüssel für beliebige Namen und Emailadressen erzeugen.
- Viele falsche Schlüssel existieren.
- Suche nach Bill Gates ergibt knapp 70 keys
- Die Lösung: **Keysigning:**
Inhaber von PGP-Schlüsseln treffen sich und verifizieren gegenseitig ihre Identitäten.

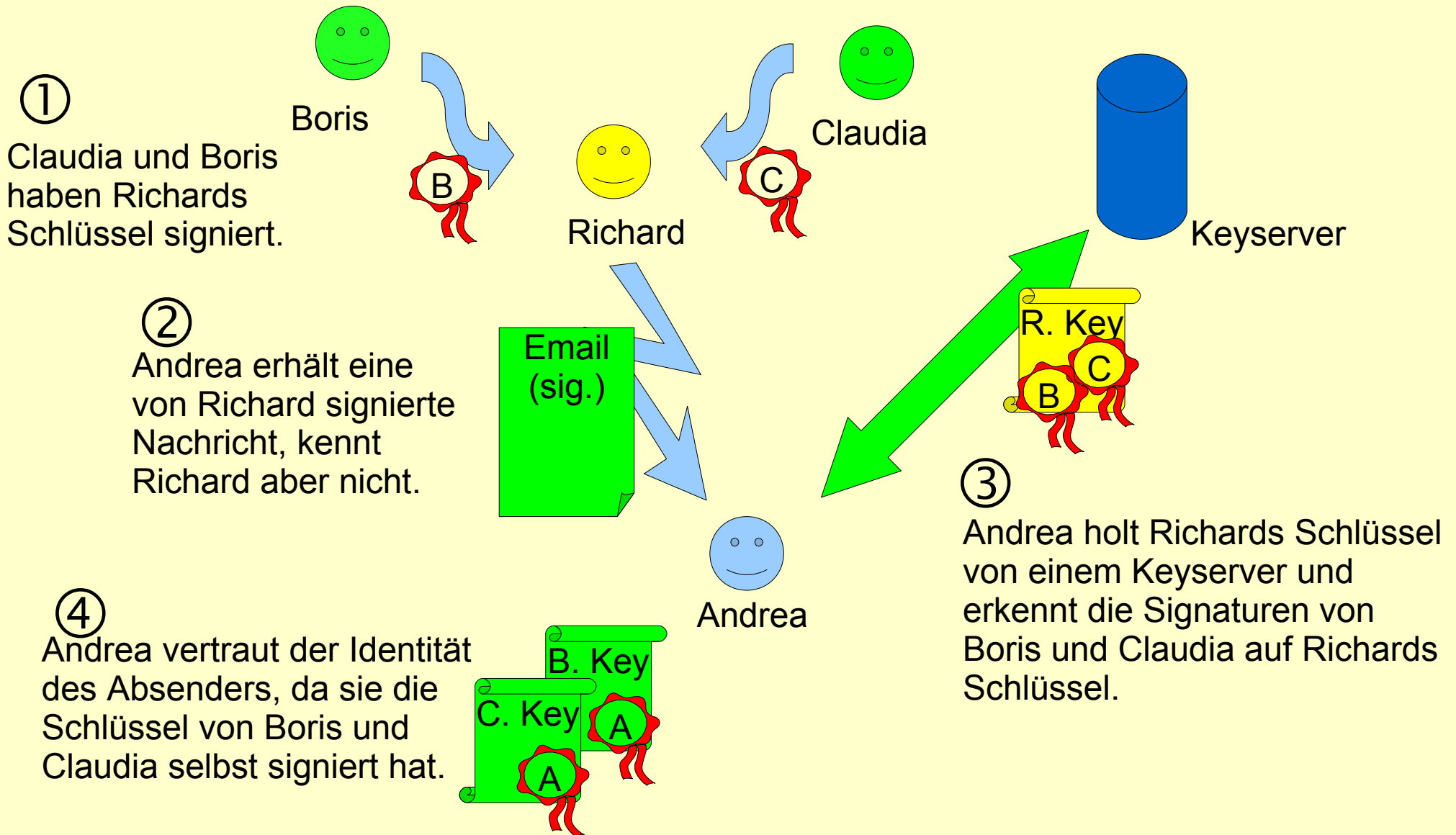
Keysigning

- Beim Keysigning überprüft man die user-id (den Namen) anhand eines Ausweises und unbedingt den Fingerprint des Schlüssels
- Danach signiert man den Schlüssel des anderen lokal auf dem eigenen Rechner
- Idealerweise signiert man jede BenutzerID auf dem Schlüssel einzeln
- Nun schickt man den signierten Schlüssel verschlüsselt an die angegebenen Emailadresse(n)
- Damit sind nun auch die Emailadressen verifiziert!
- Es gibt scripte dafür: caff, oder mycaff (Eigenwerbung):
<http://user.cs.tu-berlin.de/~mutax/ksp-scripts/mycaff/>
- das Web of Trust lebt, wie der Name sagt, vom Vertrauen:
bitte nicht blind signieren!

Web of Trust

- Hat man ein genügend großes Web of Trust, muss nicht mehr jeder jeden Schlüssel signieren!
- Jedem öffentlichen Schlüssel anderer Personen kann man verschiedene Vertrauensstufen zuweisen
- Bekommt man eine Nachricht von einem bisher unbekanntem Schlüssel kann man prüfen, ob eine gewisse Anzahl vertrauenswürdiger Bekannter diesen unterschrieben haben
- ist dies der Fall, kann man davon ausgehen, dass die Identität genügend geprüft und sicher ist (was aber nicht heißt, dass man den Schlüssel signieren kann).
- wichtiges Hilfsmittel: Keyserver, die öffentliche Schlüssel und Widerrufszertifikate vorhalten

Web of Trust (Schema)



Werkzeuge

- auf der Kommandozeile hat man die größte Kontrolle
- es gibt einige grafische Tools für Linux und Windows™
- unter Linux: gpa - den Gnu Privacy Assistant
- Plugins für die meisten Emailprogramme sind verfügbar (Thunderbird, Evolution)
- GnuPG kann beliebige Daten verschlüsseln - wie gezeigt auch lokale Dateien, nicht nur Emails

GNU Privacy Assistant - Schlüsselverwaltung

Datei Bearbeiten Schlüssel Server Fenster Hilfe

Ändern Löschen Signieren Import Export Übersicht Details Dateien

Schlüsselverwaltung

▲	Schlüsselkennung	Verfallsdatum	Benutzervertrauen	Gültigkeit des Schlüssels	Benutzerkennung
🔑	05E281DE	19.09.2005	Ultimativ	Abgelaufen	Florian Streibelt (interActive-Systems) <florian.streibelt@interActive-Systems.de>
🔑	AC804673	kein Verfallsdatum	Ultimativ	Widerrufen	[Widerrufen] Florian Streibelt <mutax@ringworld.org>
🔑	731F33F6	31.10.2004	Ultimativ	Widerrufen	[Widerrufen] Florian Streibelt <florian.streibelt@brainMedia.de>
🔑	1073C4D9	21.09.2003	Ultimativ	Widerrufen	[Widerrufen] Florian Streibelt (key for business mail) <Florian.Streibelt@interActive-Systems.de>
🔑	C7C84CE1	18.09.2005	Ultimativ	Abgelaufen	Florian Streibelt (ringworld) <mutax@ringworld.org>
🔑	D34EE3D7	23.10.2004	Ultimativ	Abgelaufen	Florian Streibelt (Technische Universität Berlin, FB Informatik/Fak IV) <florian.streibelt@tu-berlin.de>
🔑	82F61240	21.10.2010	Ultimativ	voll gültig	Florian Streibelt
🔑	CE68DFEC	26.01.2010	Ultimativ	voll gültig	Florian Streibelt

Details | **Signaturen** | Untergeordnete Schlüssel

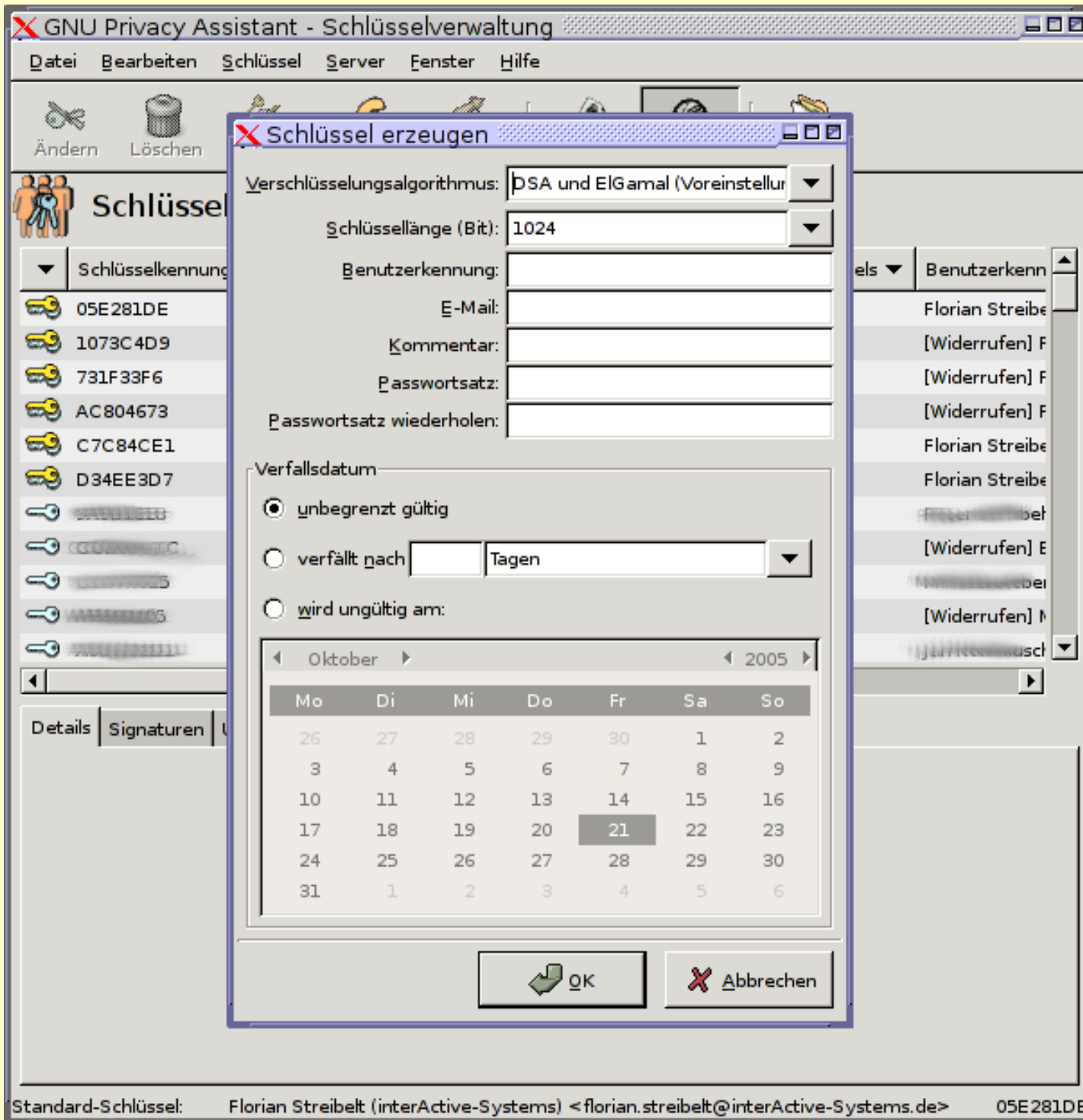
Dieser Schlüssel hat einen öffentlichen und einen geheimen Teil
 Der Schlüssel kann zur Zertifizierung, zum Signieren und zur Verschlüsselung verwendet werden.

Benutzerkennung: Florian Streibelt
 Florian Streibelt (general purpose key) <Florian.Streibelt@TU-Berlin.DE>
 Florian Streibelt (Usenet ONLY) <news@F-Streibelt.de>
 Florian Streibelt (some other account) <mutax@ringworld.org>
 Florian Streibelt (business key) <florian.streibelt@interActive-Systems.de>
 Florian Streibelt <mutax@cs.tu-berlin.de>

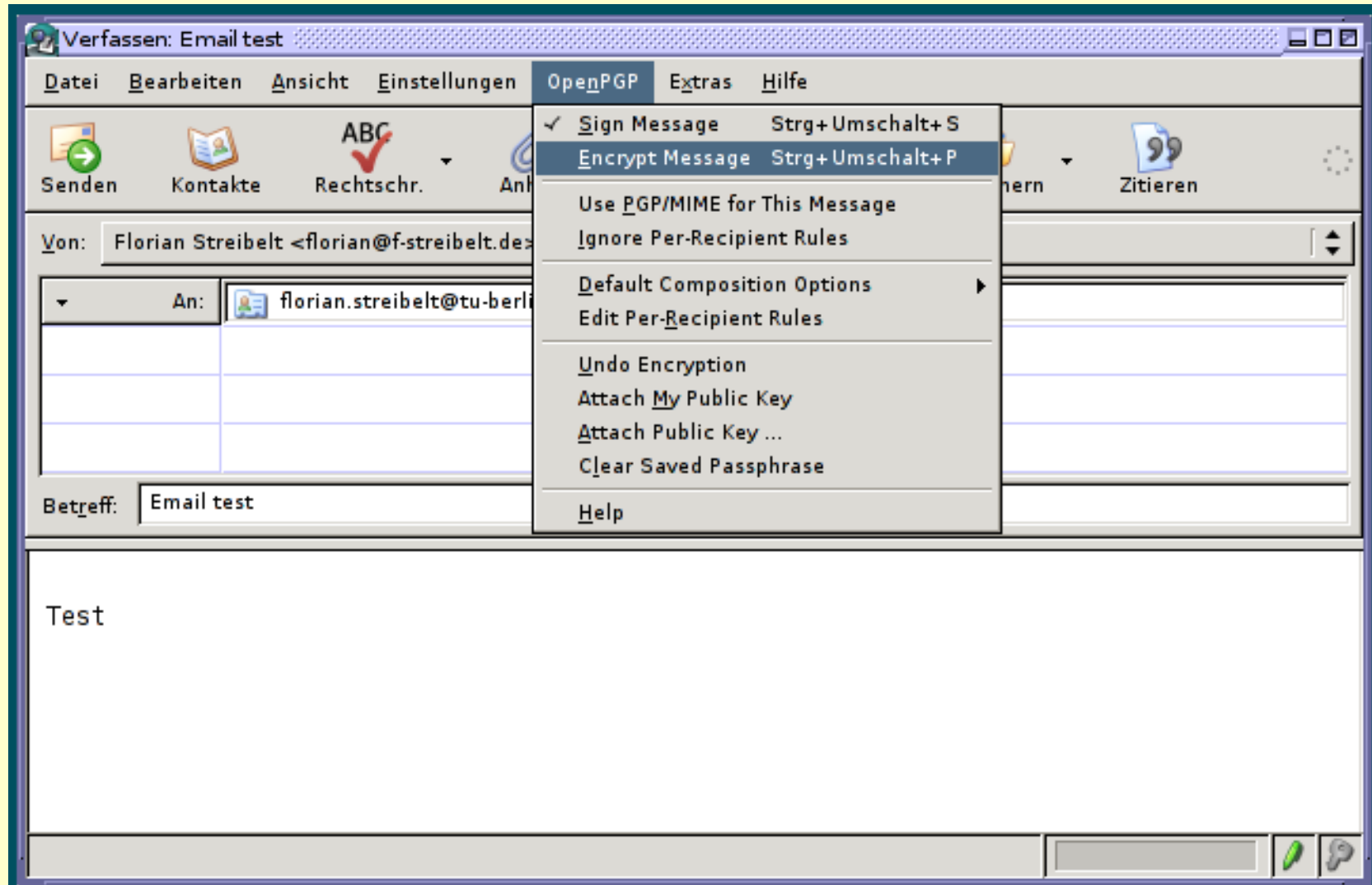
Schlüsselkennung: 82F61240
 Fingerabdruck: 5BE7 F008 8B83 9357 1108 984A 3B8E A41F 82F6 1240
 ungültig ab: 21.10.2010

Benutzervertrauen: Ultimativ
 Gültigkeit voll gültig
 Art: DSA 1024 bit
 erzeugt am: 22.10.2004

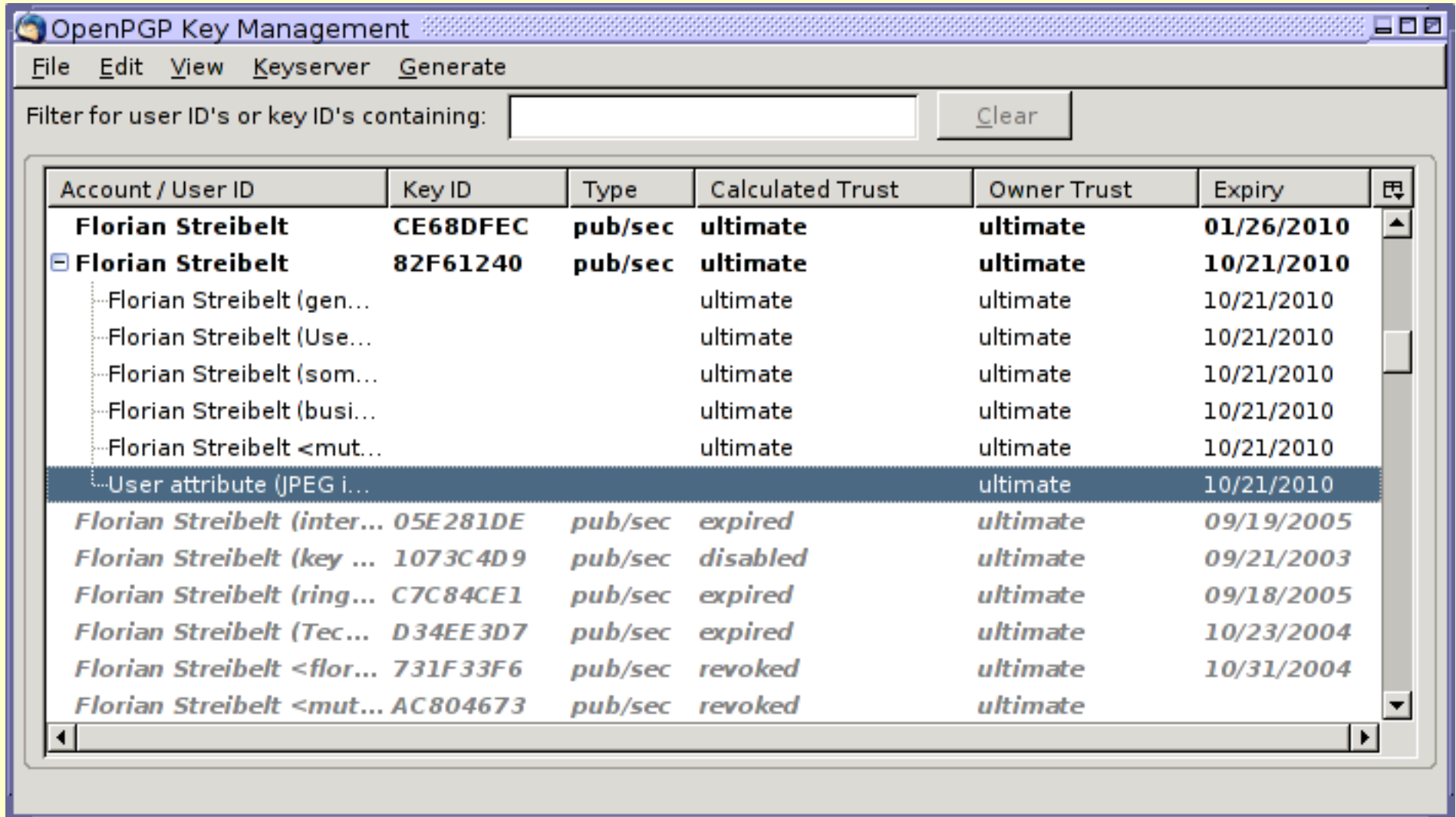
Standard-Schlüssel: Florian Streibelt (interActive-Systems) <florian.streibelt@interActive-Systems.de> 05E281DE



Eingmail – Thunderbird Plugin



Enigmail – Key Management



The screenshot shows the 'OpenPGP Key Management' application window. The title bar reads 'OpenPGP Key Management'. The menu bar includes 'File', 'Edit', 'View', 'Keyserver', and 'Generate'. Below the menu bar is a search filter: 'Filter for user ID's or key ID's containing:' followed by an empty text box and a 'Clear' button. The main area contains a table of keys.

Account / User ID	Key ID	Type	Calculated Trust	Owner Trust	Expiry	
Florian Streibelt	CE68DFEC	pub/sec	ultimate	ultimate	01/26/2010	▲
☐ Florian Streibelt	82F61240	pub/sec	ultimate	ultimate	10/21/2010	
Florian Streibelt (gen...)			ultimate	ultimate	10/21/2010	
Florian Streibelt (Use...)			ultimate	ultimate	10/21/2010	
Florian Streibelt (som...)			ultimate	ultimate	10/21/2010	
Florian Streibelt (busi...)			ultimate	ultimate	10/21/2010	
Florian Streibelt <mut...>			ultimate	ultimate	10/21/2010	
User attribute (JPEG i...)				ultimate	10/21/2010	
<i>Florian Streibelt (inter...)</i>	<i>05E281DE</i>	<i>pub/sec</i>	<i>expired</i>	<i>ultimate</i>	<i>09/19/2005</i>	
<i>Florian Streibelt (key ...)</i>	<i>1073C4D9</i>	<i>pub/sec</i>	<i>disabled</i>	<i>ultimate</i>	<i>09/21/2003</i>	
<i>Florian Streibelt (ring...)</i>	<i>C7C84CE1</i>	<i>pub/sec</i>	<i>expired</i>	<i>ultimate</i>	<i>09/18/2005</i>	
<i>Florian Streibelt (Tec...)</i>	<i>D34EE3D7</i>	<i>pub/sec</i>	<i>expired</i>	<i>ultimate</i>	<i>10/23/2004</i>	
<i>Florian Streibelt <flor...></i>	<i>731F33F6</i>	<i>pub/sec</i>	<i>revoked</i>	<i>ultimate</i>	<i>10/31/2004</i>	
<i>Florian Streibelt <mut...></i>	<i>AC804673</i>	<i>pub/sec</i>	<i>revoked</i>	<i>ultimate</i>		

Enigmail – Key erzeugen

Generate OpenPGP Key

Account / User ID Florian Streibelt <florian.streibelt@TU-Berlin.DE> - local IMAP

Use generated key for the selected identity

No passphrase

Passphrase Passphrase (repeat)

Comment

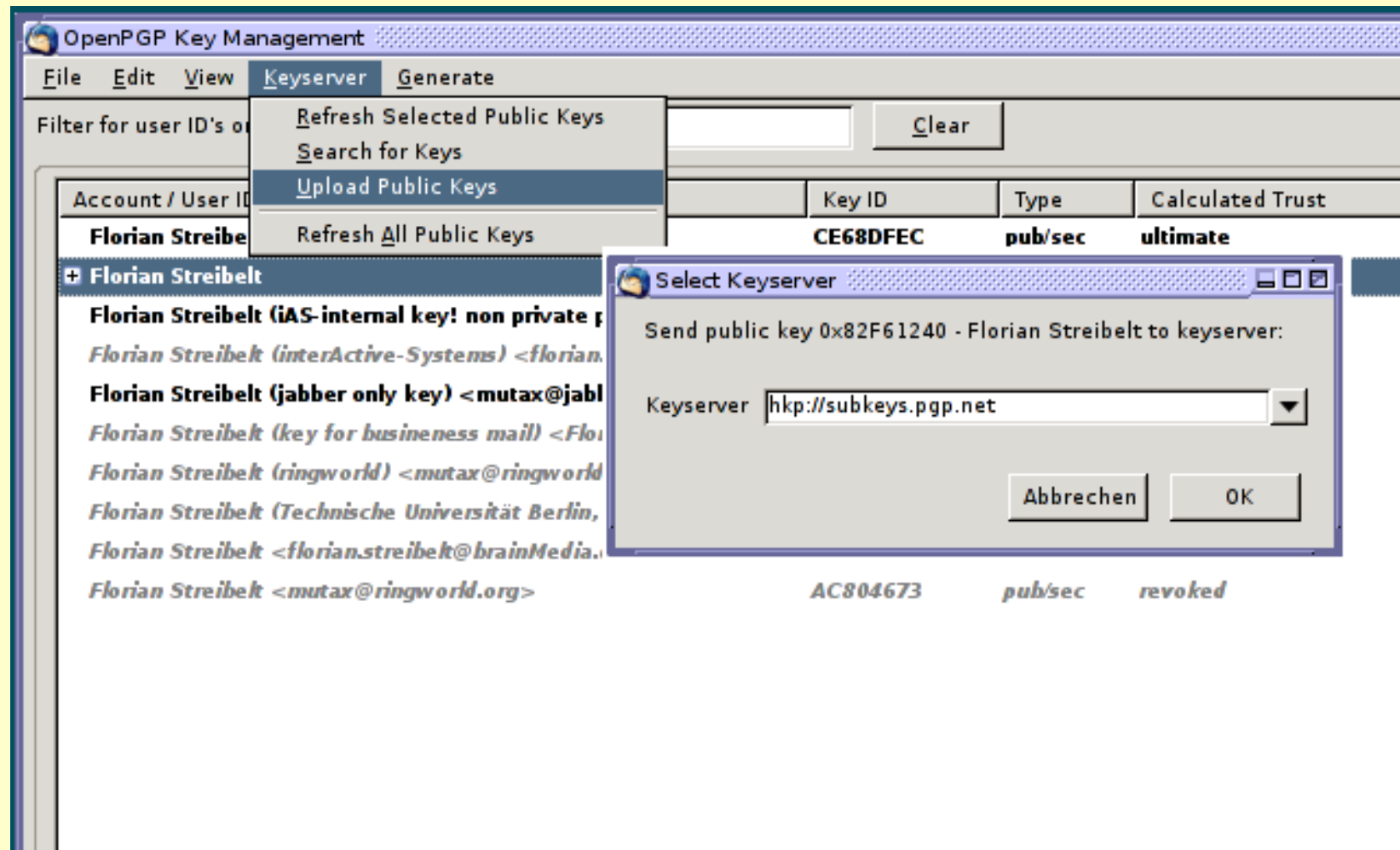
Basic | Advanced

Key expires in years Key does not expire

Key Generation Console

NOTE: Key generation may take up to several minutes to complete. Do not exit the application while key generation is in progress. Actively browsing or performing disk-intensive operations during key generation will replenish the 'randomness pool' and speed-up the process. You will be alerted when key generation is completed.

Enigmail – Key veröffentlichen



-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.1 (GNU/Linux)
hQE...M8RRL+EeuDEAP+MVU9PPQRXcQfQdq21g2L+3zSRD5qfrWhKwgTF5tI+ItU
J5X...
LTer...
drNI...
yTm...
qv/
HaJ...
rCm...
uQ/
n50...
3fY...
1t6...
Jje...
Dg...
q+...
+H...
CE...
qB...
J6dQ1L5ckkeYcXVrD...
hDPYwys8LMMckpGRn8OCMUKqlVXuXux4Ry4OUuJ...
JssQBCWrAjwC5YK0dAdSu3W25uXAZWs1eJGAi...
=Jc7W
-----END PGP MESSAGE-----

Vielen Dank für Eure Aufmerksamkeit!

Fragen? Fragen!

Keysigningparty:	am Freitag, 14.12.2005 ab 15:30 in FR5046
Anmeldung zur Party:	http://wiki.freitagrunde.org/Keysigning-Party_2007
Foliendownload unter:	http://docs.freitagrunde.org/keysigning/
Kontakt:	Florian Streibelt < Florian@freitagrunde.org >
Howtos:	http://www.gnupg.org/(de)/documentation/howtos.html

GG, Art. 10, Abs. 1:
Das Briefgeheimnis
sowie das Post- und
Fernmeldegeheimnis
sind unverletzlich (?)