Kryptographie

Rosa Freund <rosa@pool.math.tu-berlin.de>

D466 2926 104B 820E DD7F 6956 5E63 529D E219 AC59

30. September 2006

Problem bei Public-Key-Verfahren: Ist der Schlüssel echt?

Ich will Alice eine verschlüsselte Nachricht schicken, habe ihren öffentlichen Schlüssel aber nicht.

- Alice gibt mir ihren öffentlichen Schlüssel (oder den Fingerprint ihres Schlüssels) persönlich.
- Alice mailt mir ihren öffentlichen Schlüssel.
- Ich lade mir Alices öffentlichen Schlüssel aus dem Internet (von ihrer Website oder einem Keyserver) herunter.

Lösungsmöglichkeiten

- Hierarchische Authentifizierung durch Zertifikate (z.B. X.509, S/MIME)
- Vertrauensnetze
- Lösung für PGP: **Web of Trust** und **Keysigning**

PGP

- Pretty Good Privacy, Phil Zimmerman 1991, GPL-lizenzierte Weiterentwicklung: **OpenPGP** (RFC 2440)
- Hybrid-Verschlüsselungssystem: Verwendet asymmetrische Verfahren nur zur Verschlüsselung eines Schlüssels. Mit diesem wird die eigentliche Nachricht symmetrisch verschlüsselt.
- Freie Implementation von OpenPGP: **GPG** (Gnu Privacy Guard)
- Graphische Frontends für GPG für die meisten Plattformen erhältlich, z.B. KGPG, Enigmail (Thunderbird-Extension), Mac GPG

Schlüsselmanagement

- PGP-Schlüsselpaare werden lokal erzeugt und die öffentlichen Schlüssel auf Keyserver (z.B. `http://pgp.mit.edu/`) hochgeladen
- Schlüssel sind mit einer **User-ID** assoziiert, die aus Namen und Mailadresse besteht. Die User-ID ist variabel, man kann beliebige Namen (auch Pseudonyme) und mehrere Mailadressen mit einem Schlüssel assoziieren.
- Auf Visitenkarten, Webseiten, bei Keysigningparties: **Fingerprint** $\longrightarrow$ Hash des öffentlichen Schlüssels ermöglicht eindeutige Zuordnung

Schlüssel signieren

- Lokal gespeicherte öffentliche Schlüssel können mit dem eigenen Schlüssel **signiert** und wieder auf Keyserver hochgeladen werden.

- Wenn ich einen Schlüssel signiere, bewerte ich:

  1. Wie überzeugt ich davon bin, dass Identität, Mailadresse und Schlüssel tatsächlich zusammengehören. **Öffentlich!**
  2. Für wie vertrauenswürdig ich die Signatur dieser Person auf anderen Schlüsseln halte. **Privat!**

- Die beiden Aspekte können unabhängig voneinander bewertet werden.

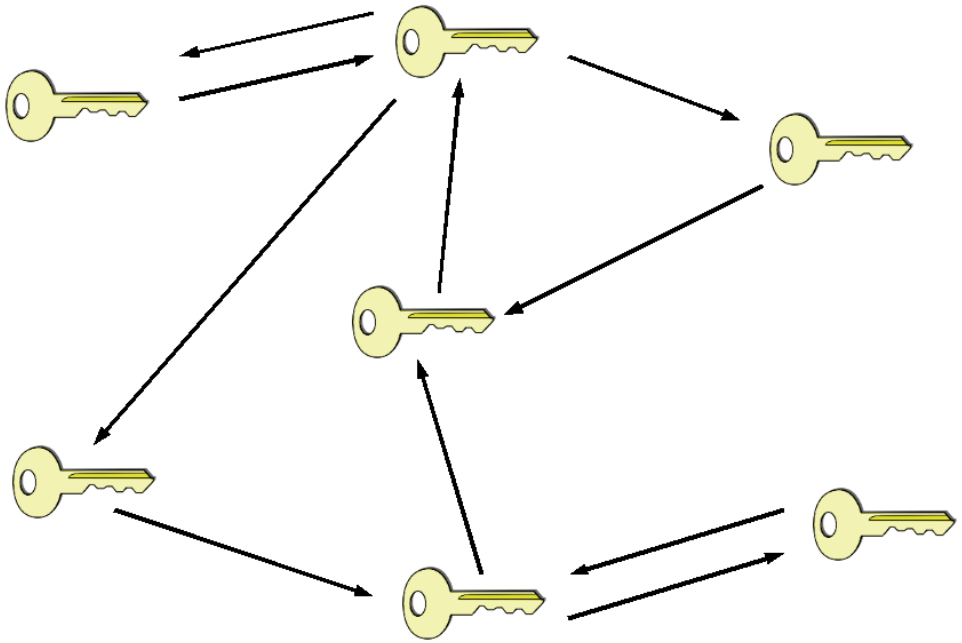
PGP Strong Set 1

- So entsteht über Signaturen ein Netzwerk des Vertrauens, das **Web of Trust** (ich vertraue A, A hat den Schlüssel von B signiert, also kann ich Bs Schlüssel vertrauen)

- Schlüssel und Signaturen sind als gerichteter Graph darstellbar

- Das **Strong Set** ist die grösste PGP-Schlüsselmenge, in der von jedem Schlüssel zu jedem anderen Schlüssel ein Signaturpfad besteht.
  Grösse derzeit: ca. 30 000 Schlüssel.

Strong Set



PGP Strong Set 2

- Vertrauensbarometer: **MSD**, **mean shortest distance**. In einem gerichteten Graph kann berechnet werden, wie weit ein bestimmter Key **im Durchschnitt** von anderen Schlüsseln über Signaturen entfernt ist. Je kleiner die MSD, desto vertrauenswürdiger der Schlüssel
  Meine MSD: 4,99

- Skripte verfügbar, die MSD und Pfade zwischen beliebigen Schlüsseln berechnen, z.B. http://www.cs.uu.nl/people/henkp/henkp/pgp

- Mein Schlüssel ist im strong set, sobald mein Schlüssel und ein beliebiger Schlüssel im strong set die gegenseitige Signatur tragen

Keysigning 1

Zwei Aspekte des Vertrauens in PGP:

- **Wie stark vertraue ich der Verbindung von Schlüssel und User-ID?**
  Wie sicher bin ich, dass Mailadresse und Person und öffentlicher Schlüssel wirklich zusammengehören?
  Eigentliche Signatur, wird über Keyserver veröffentlicht

- **Wie stark vertraue ich Signaturen dieses öffentlichen Schlüssels?**
  Für wie sorgfältig halte ich den Schlüsseleigentümer bei der Signatur anderer Schlüssel?
  'Trust', privat

Keysigning 2

Erster Aspekt:

```
rosa@horst:~> gpg --sign-key E4F7B641
How carefully have you verified the key you are about to sign
actually belongs to the person named above? If you don't know
what to answer, enter '0'.

(0) I will not answer. (default)
(1) I have not checked at all.
(2) I have done casual checking.
(3) I have done very careful checking.
```

Keysigning 3

Zweiter Aspekt:

```
rosa@horst: > gpg --edit-key 40E8BD10
Command> trust
Please decide how far you trust this user to correctly verify
other users' keys (by looking at passports, checking fingerprints
from different sources, etc.)

1 = I don't know or won't say
2 = I do NOT trust
3 = I trust marginally
4 = I trust fully
5 = I trust ultimately
```

Keysigning 4

- Beim ersten Aspekt kann (und sollte) ich die höchste Vertrauensstufe ('Level 3') nur vergeben, falls ich die Person getroffen, ihren Ausweis kontrolliert, sowie eine Challenge-Response-Mail geschickt habe.
- Beim zweiten Aspekt sind die Bewertungen weniger klar definierbar.
- Bei beiden Aspekten liegt die Entscheidung, welchen Wert ich wähle, letztendlich aber nur bei mir.

Keysigning 5: Praxis

- Bob will Alices Schlüssel signieren. Alice muss ihm dazu ihren Lichtbildausweis und Fingerprint des öffentlichen Schlüssels zeigen.
- Bob überprüft Ausweis und notiert Fingerprint. Zuhause lädt er sich den Schlüssel von Alice herunter und bestätigt, dass der Fingerprint stimmt.
- Um zu überprüfen, ob der Schlüssel wirklich zu Alice gehört und die Mailadresse funktioniert, kann er noch eine Challenge-Response-Mail schicken.

Auf der Keysigningparty

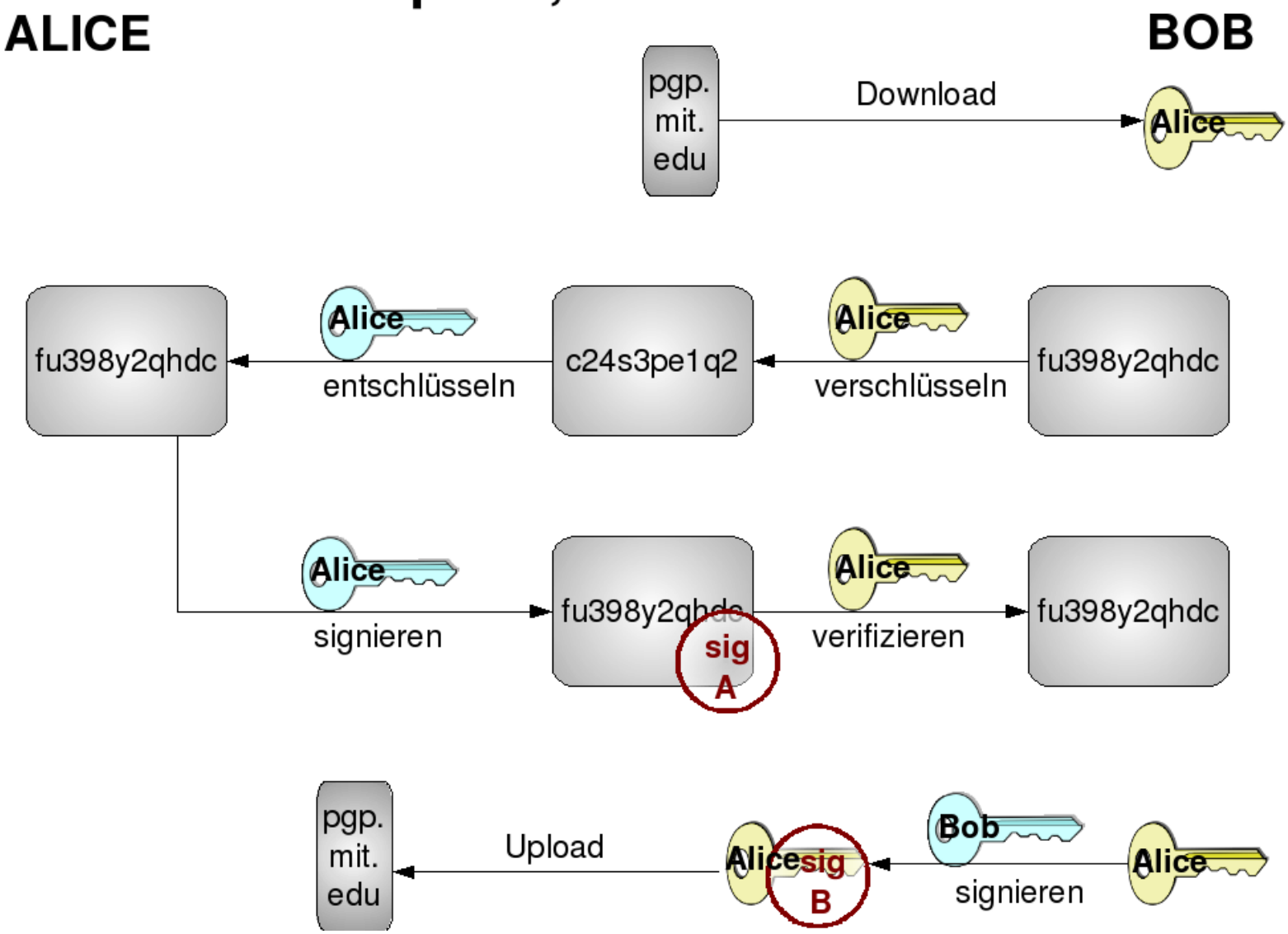
ALICE

möchte viele Signaturen sammeln

BOB

möchte Alices Schlüssel signieren

Später, zuhause



Literatur

Bruce Schneier: *Applied Cryptography*

GnuPG Keysigning Party HOWTO
`http://www.cryptnet.net/fdp/crypto/gpg-party.html`

Enigmail
`http://enigmail.mozdev.org/`

RSA Security Cryptography FAQ
`www.rsasecurity.com/rsalabs/faq/`

Wikiprojekt Kryptologie
`http://de.wikipedia.org/wiki/Wikipedia:WikiProjekt_Kryptologie`

Und jetzt...

Noch Fragen?