PGP und das Web of Trust

Thomas Merkel

Frubar Network

14. Juni 2007

<drscream@frubar.net>
E509 273D 2107 23A6 AD86 1879 4C0E 6BFD E80B F2AB

Inhalt

Inhalt

Inhalt

Inhalt

Ziele kryptographischer Verfahren

- **Datenschutz**: Nachricht soll nur vom vorgesehenen Empfänger gelesen werden können.
- **Authentizität**: Feststellen ob die Nachricht tatsächlich von einem bestimmten Absender stammt.
- **Integrität**: Eine Nachricht auf ihrem Weg zu mir nicht verändert wurde.

Lösung: Verschlüsselen (Datenschutz) und Signieren (Authentizität, Integrität)

Allgemeines

Zum verschlüsselten Kommunizieren benötigen die Parteien:

- Ein Verschlüsselungsverfahren (z.B. PGP, SSL)
- Einen oder mehrere Schlüssel.

Warum verschlüsseln?

Ich habe doch nichts zu verbergen?

Nothing to hide, but something to protect!

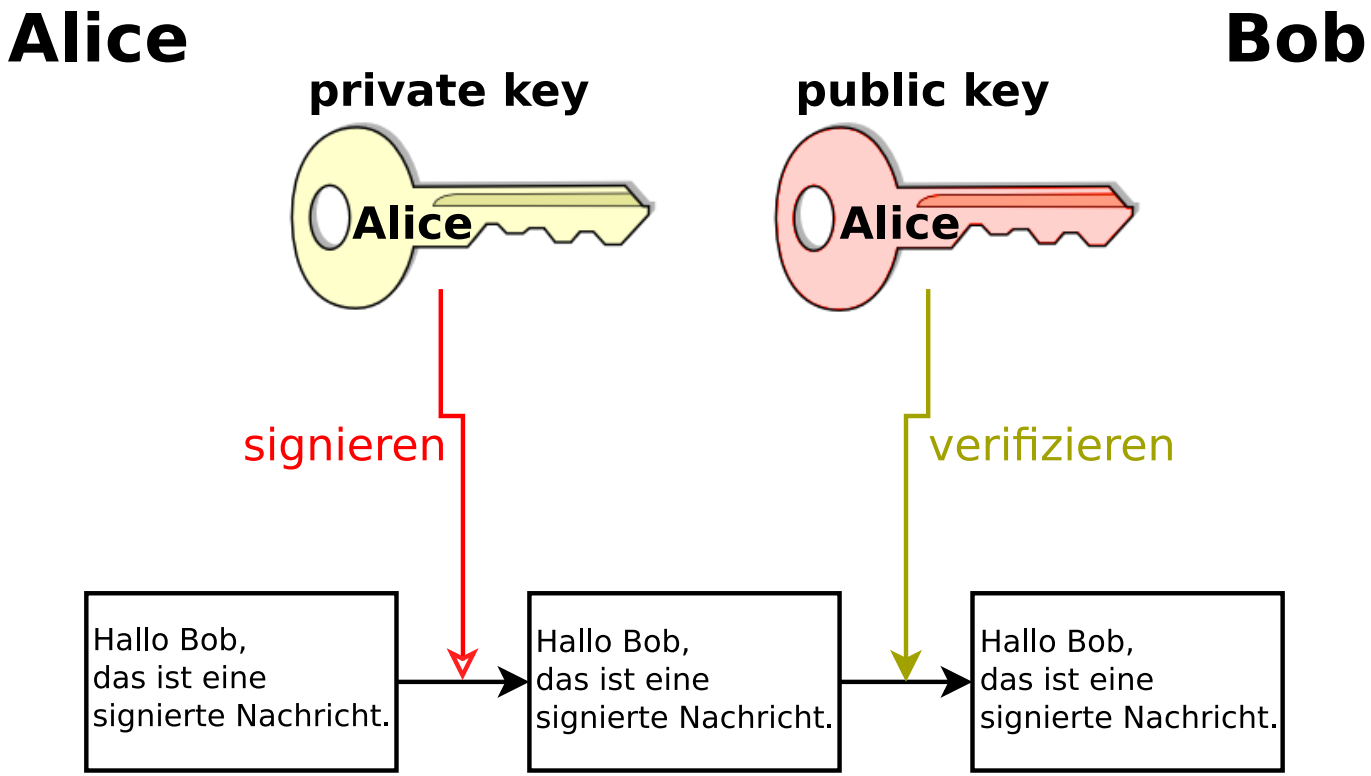
Asymmetrische Verfahren (1)

- Alle Nutzen, Besitzen einen Öffentlichen sowie einen geheimen Schlüssel (public key, private key).
- z.B.: RSA, El-Gamal, elliptische Kurven

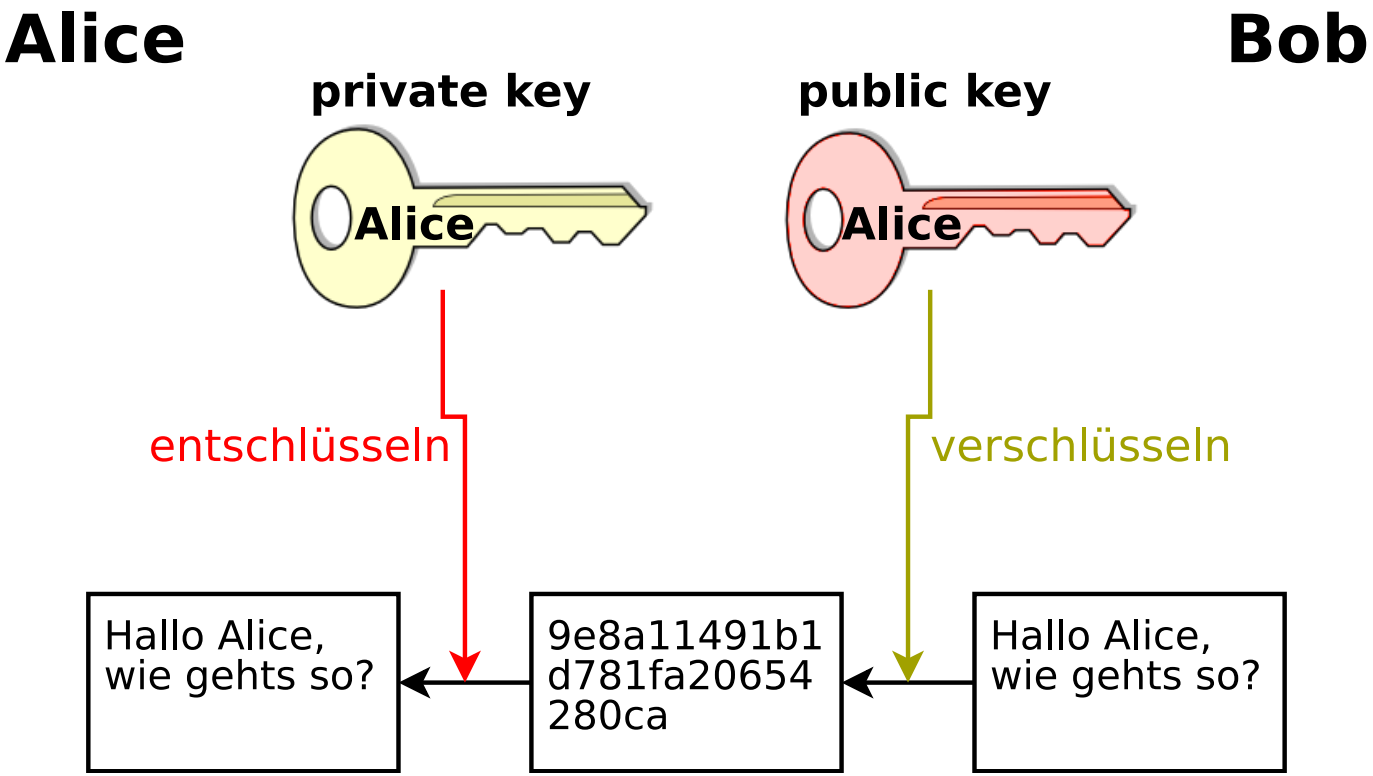
Asymmetrische Verfahren (2)

- Um verschlüsselte Nachrichten erhalten bzw. Nachrichten signieren zu können, generiert Alice einen Öffentlichen und einen geheimen Schlüssel.
- Alice muss ihren geheimen Schlüssel (private key) geheimhalten, den Öffentlichen (public key) jedoch veröffentlichen.
- **Signieren**: Alice signiert die Nachricht mit ihrem geheimen Schlüssel. Bob benutzt Alice Öffentlichen Schlüssel, um die Signatur zu verizieren.
- **Verschlüsseln**: Bob nutzt den Öffentlichen Schlüssel von Alice, um eine Nachricht an Alice zu verschlüsseln. Alice entschlüsselt die Nachricht mit ihrem geheimen Schlüssel.

Signieren


Alice
Bob
private key
public key
Alice
Alice
signieren
verifizieren
Hallo Bob,
das ist eine
signierte Nachricht.
Hallo Bob,
das ist eine
signierte Nachricht.
Hallo Bob,
das ist eine
signierte Nachricht.

Verschlüsseln



Problem bei Public-Key-Verfahren

Ist der Key echt?

- Wenn ich eine Person persönlich kenne, und sie mir ihren Öffentlichen Schlüssel selbst übergibt, kann ich dem Schlüssel vertrauen.
- **Problem**: Wie kann ich Öffentlichen Schlüsseln von Personen vertrauen, die mir nicht persönlich bekannt sind, oder von denen ich die öffentlichen Schlüssel per Mail oder via einem Keyserver erhalte?

Problem bei Public-Key-Verfahren

Lösungsmöglichkeiten

- Hierarchische Authentizierung durch Zertikate (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)
- Freie Implementation von OpenPGP: **GPG** (Gnu Privacy Guard)
  Graphische Frontends für GPG: 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.
- Auf Visitenkarten, Webseiten, bei Keysigningparties: **Fingerprint** $\Rightarrow$ Hash des öffentlichen Schlüssels.

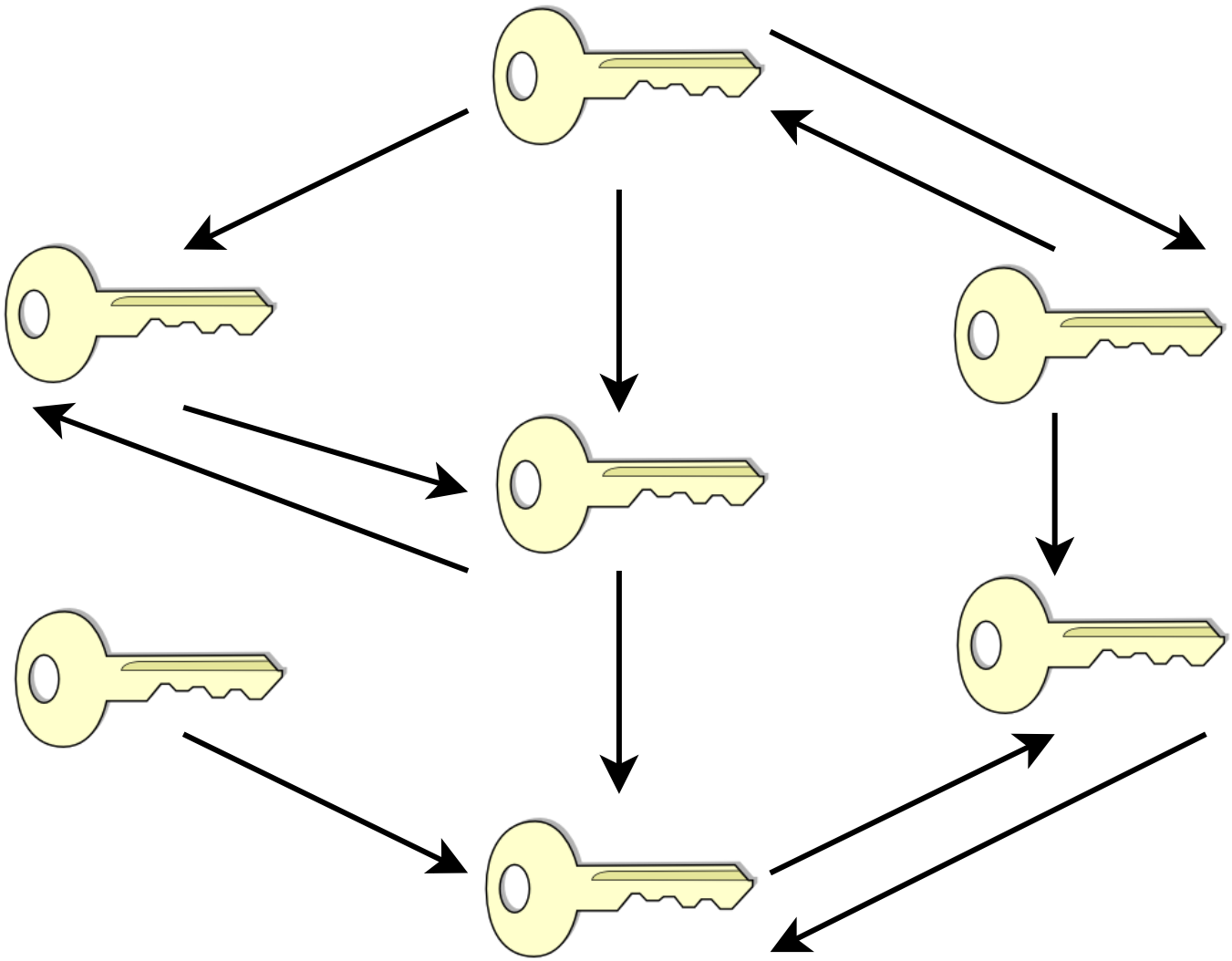
Schlüssel signieren

- 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!

PGP Strongset (1)

- So entsteht über Signaturen ein Netzwerk des Vertrauens, das **Web of Trust**
- Schlüssel und Signaturen sind als gerichteter Graph darstellbar.
- Das **Strongset** ist die grösste PGP-Schlüsselmenge, in der von jedem Schlüssel zu jedem anderen Schlüssel ein Signaturpfad besteht. Grösse derzeit: ca. 45 000 Schlüssel.

Strongset



PGP Strongset (2)

- Vertrauensbarometer: **MSD, mean shortest distance**. Je kleiner die MSD, desto vertrauenswürdiger der Schlüssel.
- Skripte verfügbar, die MSD und Pfade zwischen beliebigen Schlüsseln berechnen, z.B. http://www.cs.uu.nl/people/henkp/henkp/
- Mein Schlüssel ist im Strongset, sobald mein Schlüssel und ein beliebiger Schlüssel im Strongset die gegenseitige Signatur tragen.

Keysigning (1)

Zwei Aspekte des Vertrauens in PGP:

- **Wie stark vertraue ich der Verbindung von Schlüssel und User-ID?**
- **Wie stark vertraue ich Signaturen dieses öffentlichen Schlüssels?**

Keysigning (2) - Erster Aspekt

```
drscream@havanna:~> gpg --sign-key E80BF2AB
How carefully have you verified the key you are about to
sign actually belongs to the person named above? If you
dont 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 Aspekte

```
drscream@havanna:~> 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 dont know or wont say
 2 = I do NOT trust
 3 = I trust marginally
 4 = I trust fully
 5 = I trust ultimately
```

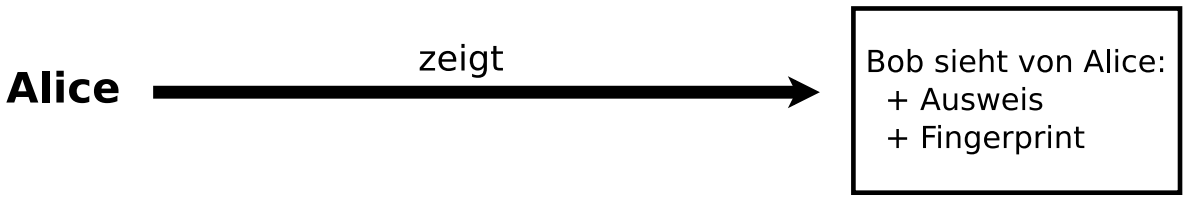
Auf der Keysigning-Party

Alice

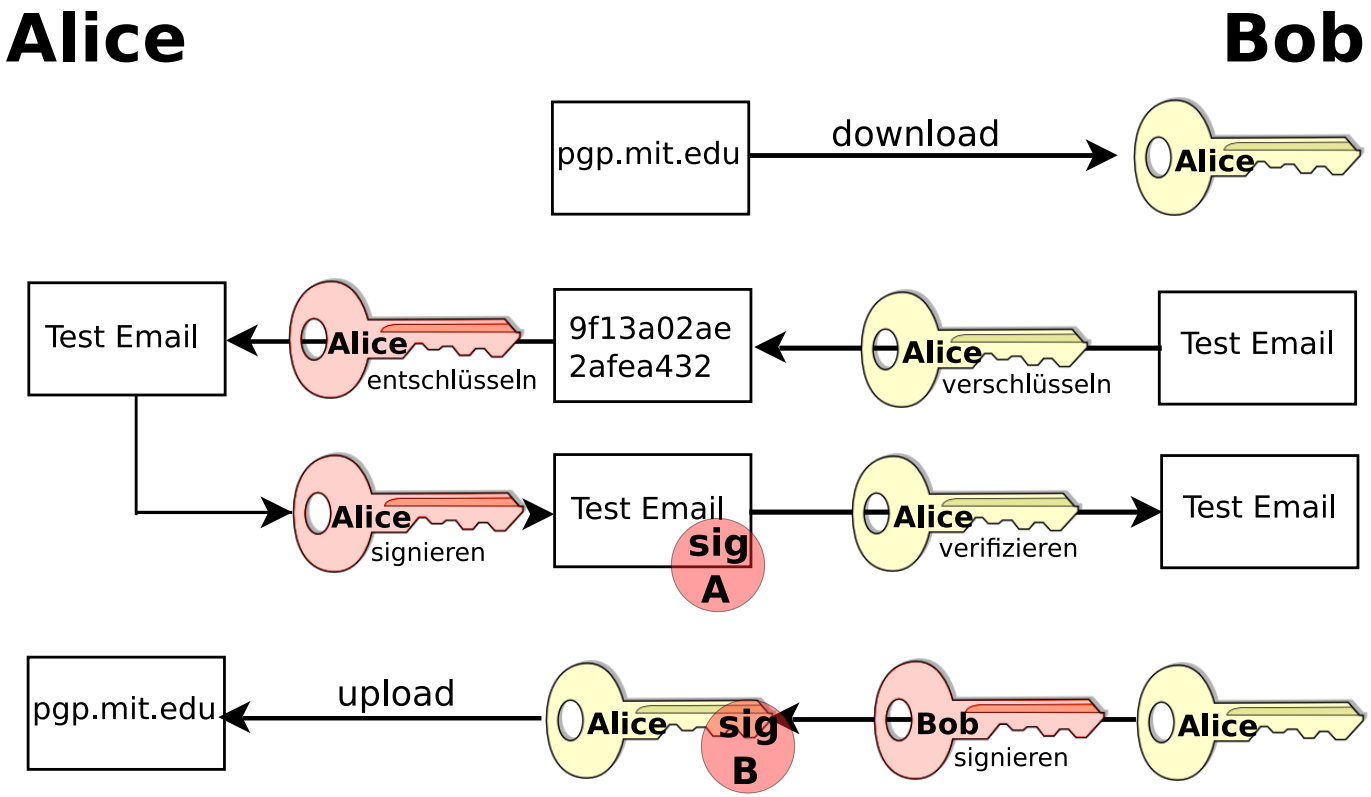
möchte viele Signaturen sammeln

Bob

möchte Alice 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
  http://www.rsasecurity.com/rsalabs/faq/
- Wikiprojekt Kryptologie
  http://de.wikipedia.org/wiki/Wikipedia:WikiProjekt Kryptologie

Diskussion

Vielen Dank für Ihre Aufmerksamkeit.