Einführung in OpenPGP

Schlüssel, Vertrauen und das Web of Trust

Jens Erat
OpenRheinRuhr 2014

9. November 2014

Outline

- Was und warum?
- Schlüssel schleifen
- Kenne ich Dich? Das Web of Trust
- Fragen

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/.

Was und warum?

Was ist OpenPGP?

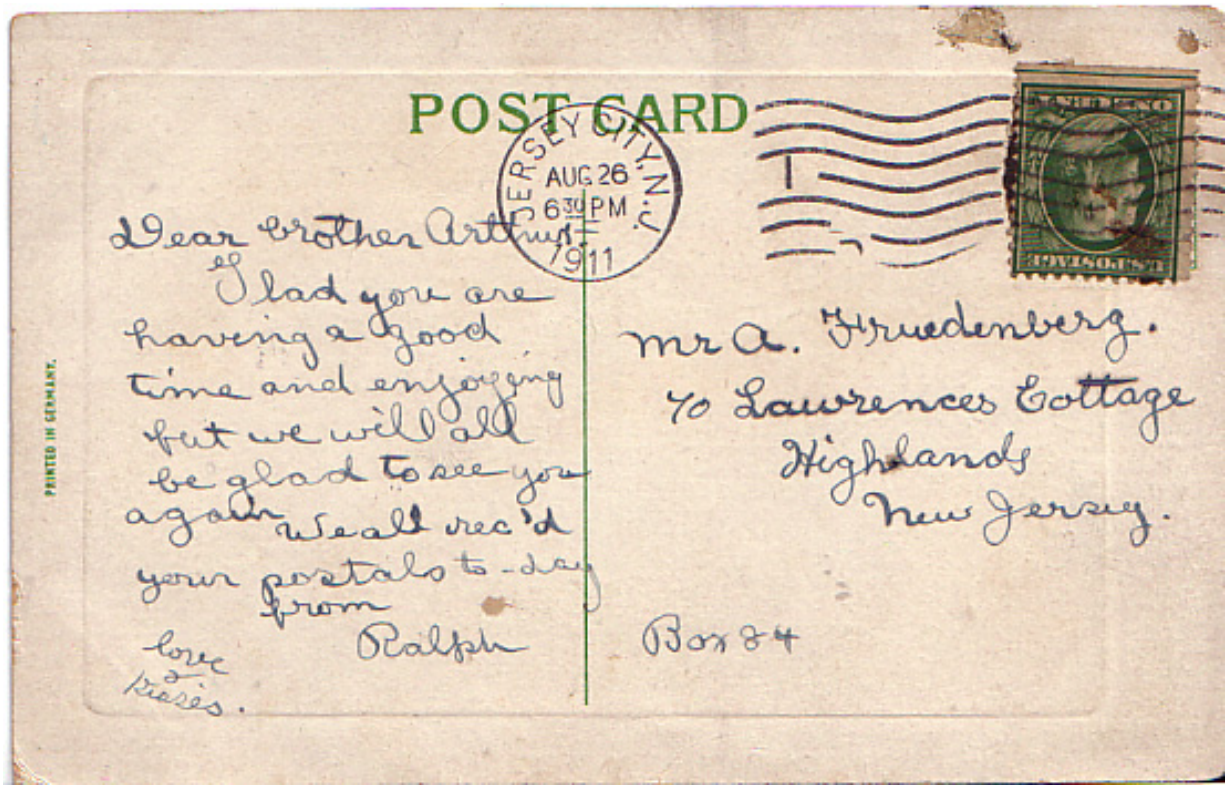
> *OpenPGP software uses a combination of strong* ***public-key and symmetric cryptography*** *to provide security services for electronic communications and data storage. These services include* ***confidentiality***, ***key management***, *authentication, and* ***digital signatures***.[1]

Produkte und Standards

- **PGP**: Phil Zimmermann entwickelt, proprietär
- **OpenPGP**: Standardisierung des Protokolls hinter PGP als RFC 4880
- **GnuPG**: "GNU Privacy Guard", freie Implementierung von OpenPGP

[1] Aus: RFC 4880, OpenPGP Message Format

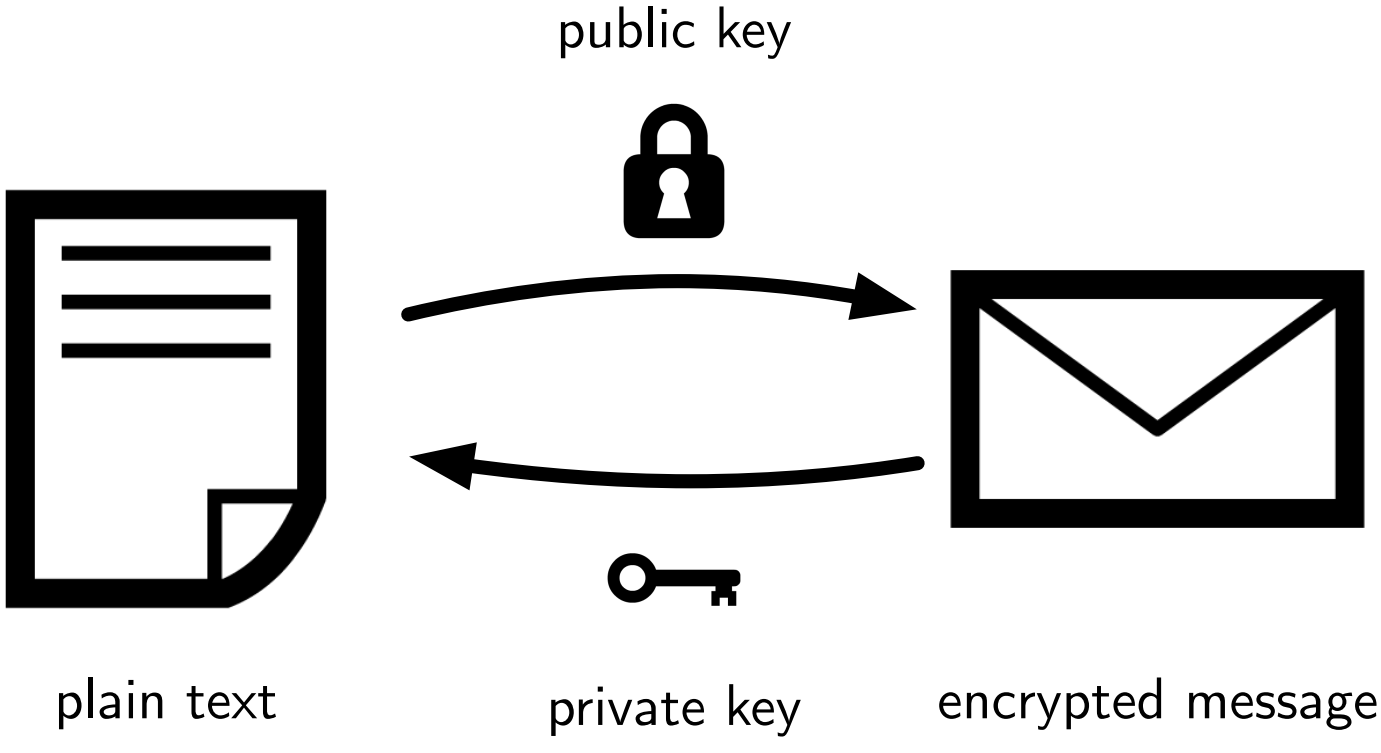
E-Mails sind Postkarten



http://commons.wikimedia.org/wiki/File:Freudenberg_ArthurOscar_02.jpg

Assymetrische Verschlüsselung

- Nutzung mathematischer Verfahren
- Zwei Schlüssel:
  - Privater Schlüssel bleibt geheim
  - Öffentlicher Schlüssel kann beliebig verteilt werden
- Mit öffentlichem verschlüsseln, mit privatem entschlüsselbar



Wie funktioniert digitale Signatur?

- Umgekehrte Verwendung von privaten und öffentlichen Schlüsseln
- Hashsumme mit privatem Schlüssel verschlüsseln, jeder kann entschlüsseln
- Ergibt sich Hashsumme, muss mit privatem Schlüssel verschlüsselt (signiert) worden sein

Hybride Verschlüsselung

- Assymetrische Verschlüsselungsverfahren sind langsam
  - Nur für kleine Datenmengen geeignet
- Symmetrische Verfahren sind schnell
- Symmetrischen Cipherblock mit assymetrischer Verschlüsselung verschlüsseln

OpenPGP benutzen

- (Symantec) PGP: Kaufsoftware
- GnuPG: Freie Software
  - Linux: Paketquellen
  - Windows: GPG4Win
  - OS X: GPGTools
- Plugins für E-Mail-Anwendungen
  - Thunderbird: Plugin “Enigmail”
  - Outlook: GpgOL
  - Apple Mail: Plugin “GPGMail”

Schlüssel schleifen

Was ist ein OpenPGP-Schlüssel?

- **Hauptschlüssel $\hat{=}$ eigenständige Identität**
- Empfohlene Verwendung: nur Beglaubigung
- Bestandteile
  - Unterschlüssel (*sub*), Verwendung: Signatur und Verschlüsselung
  - Benutzer-ID (*uid*): Wem gehört der Schlüssel?
- Gültigkeitsdatum
- *Eingehende* Signaturen (Beglaubigungen)

Beispiel eines OpenPGP-Schlüssels

```
$ gpg --list-keys A4FF2279

pub   8192R/A4FF2279 2012-12-25
uid          Jens Erat (born 1988-01-19 in
                                    Stuttgart, Germany)
uid          Jens Erat <email@jenserat.de>
uid          Jens Erat <jens.erat@uni-konstanz.de>
uid          Jens Erat <jens.erat@fsfe.org>
uid          Jens Erat <jabber@jenserat.de>
uid          [jpeg image of size 12899]
sub   4096R/759A536E 2012-12-26 [verfällt: 2016-12-26]
sub   4096R/CC85941A 2012-12-26 [verfällt: 2016-12-26]
sub   2048R/CB4BD3EE 2013-01-23 [verfällt: 2023-01-21]
sub   2048R/3E2554DF 2013-01-23 [verfällt: 2023-01-21]
```

Welchen Algorithmus wählen?

- OpenPGP erlaubt verschiedene Algorithmen - Verbeitet: RSA, DSA/Elgamal
  - Manche Implementierungen unterstützen weitere Algorithmen

- Debian empfiehlt RSA mit 4096 bit[2]
- Schlüsselgröße: Rechenaufwand vs. Sicherheit[3]
  - Größere Schlüssellänge für langlebigere Primärschlüssel

[2] https://wiki.debian.org/subkeys
[3] http://superuser.com/a/541162/102155

Benutzer-IDs: Wer bin ich?

- Mehrere Benutzer-IDs pro Schlüssel
- Schlüsselsuche: z.B. durch E-Mail-Adresse
- Beglaubigungen bestätigen Identität

- Benutzer-IDs können nicht gelöscht werden, nur widerrufen
- Benutzer-IDs werden mit Schlüsselservern synchronisiert (Spam)

Benutzer-IDs: Wer bin ich?

- Mehrere Benutzer-IDs pro Schlüssel
- Schlüsselsuche: z.B. durch E-Mail-Adresse
- Beglaubigungen bestätigen Identität

- Benutzer-IDs können nicht gelöscht werden, nur widerrufen
- Benutzer-IDs werden mit Schlüsselservern synchronisiert (Spam)

Kommentare

- 99,5% aller Kommentare sind überflüssig[4]
- Wiederhole Dich nicht.

```
John Q. Public (Debian) <johnqpublic@debian.org>
```

- Wir wissen, dass das Deine Uni-E-Mail-Adresse ist.

```
John Q. Public (University) <johnqpublic@example.edu>
```

- Widerrufe Deine alten Schlüssel!

```
J. Public (This is my second key) <john@example.com>
J. Public (This is my primary key) <john@example.com>
J. Public (No wait really use this one) <john@example.com>
```

- (Kein Kommentar)

```
John Q. Public (none) <johnqpublic@example.com>
```

[4] http://www.debian-administration.org/users/dkg/weblog/97

Kommentare

- 99,5% aller Kommentare sind überflüssig[4]
- Wiederhole Dich nicht.

  ```
  John Q. Public (Debian) <johnqpublic@debian.org>
  ```

- Wir wissen, dass das Deine Uni-E-Mail-Adresse ist.

  ```
  John Q. Public (University) <johnqpublic@example.edu>
  ```

- Widerrufe Deine alten Schlüssel!

  ```
  J. Public (This is my second key) <john@example.com>
  J. Public (This is my primary key) <john@example.com>
  J. Public (No wait really use this one) <john@example.com>
  ```

- (Kein Kommentar)

  ```
  John Q. Public (none) <johnqpublic@example.com>
  ```

[4] http://www.debian-administration.org/users/dkg/weblog/97

Kommentare

- 99,5% aller Kommentare sind überflüssig[4]
- Wiederhole Dich nicht.

  ```
  John Q. Public (Debian) <johnqpublic@debian.org>
  ```

- Wir wissen, dass das Deine Uni-E-Mail-Adresse ist.

  ```
  John Q. Public (University) <johnqpublic@example.edu>
  ```

- Widerrufe Deine alten Schlüssel!

  ```
  J. Public (This is my second key) <john@example.com>
  J. Public (This is my primary key) <john@example.com>
  J. Public (No wait really use this one) <john@example.com>
  ```

- (Kein Kommentar)

  ```
  John Q. Public (none) <johnqpublic@example.com>
  ```

[4] http://www.debian-administration.org/users/dkg/weblog/97

Kommentare

- 99,5% aller Kommentare sind überflüssig[4]
- Wiederhole Dich nicht.

```
John Q. Public (Debian) <johnqpublic@debian.org>
```

- Wir wissen, dass das Deine Uni-E-Mail-Adresse ist.

```
John Q. Public (University) <johnqpublic@example.edu>
```

- Widerrufe Deine alten Schlüssel!

```
J. Public (This is my second key) <john@example.com>
J. Public (This is my primary key) <john@example.com>
J. Public (No wait really use this one) <john@example.com>
```

- (Kein Kommentar)

```
John Q. Public (none) <johnqpublic@example.com>
```

[4] http://www.debian-administration.org/users/dkg/weblog/97

Benutzer-IDs: Wer bin ich?

Die primäre Nutzer-ID

- E-Mail-Adressen ändern sich
- Verlust der Beglaubigungen beim Widerruf einer Benutzer-ID

- Vorschlag: Primäre Nutzer-ID ohne E-Mail-Adresse
  - Enthält nur unveränderliche Informationen
  - Name, Geburtsdatum, -ort (Abwägung Datenschutz!)

Benutzer-IDs: Wer bin ich?

Die primäre Nutzer-ID

- E-Mail-Adressen ändern sich
- Verlust der Beglaubigungen beim Widerruf einer Benutzer-ID

- Vorschlag: Primäre Nutzer-ID ohne E-Mail-Adresse
  - Enthält nur unveränderliche Informationen
  - Name, Geburtsdatum, -ort (Abwägung Datenschutz!)

Unterschlüssel

- Jeder (Haupt-)Schlüssel kann mehrere Unterschlüssel haben
- Technische Notwendigkeit bei DSA/Elgamal
- Einzeln widerrufbar

Offline-Hauptschlüssel

- Besonders gesichert (externer Datenträger)
- Nur selten (für Beglaubigungen) benötigt

OpenPGP-Chipkarte

- Offline-Schlüssel, verbleibt auf Karte
- Karte hat eigenen Cryptoprozessor

Unterschlüssel

- Jeder (Haupt-)Schlüssel kann mehrere Unterschlüssel haben
- Technische Notwendigkeit bei DSA/Elgamal
- Einzeln widerrufbar

Offline-Hauptschlüssel

- Besonders gesichert (externer Datenträger)
- Nur selten (für Beglaubigungen) benötigt

OpenPGP-Chipkarte

- Offline-Schlüssel, verbleibt auf Karte
- Karte hat eigenen Cryptoprozessor

Unterschlüssel

- Jeder (Haupt-)Schlüssel kann mehrere Unterschlüssel haben
- Technische Notwendigkeit bei DSA/Elgamal
- Einzeln widerrufbar

Offline-Hauptschlüssel

- Besonders gesichert (externer Datenträger)
- Nur selten (für Beglaubigungen) benötigt

OpenPGP-Chipkarte

- Offline-Schlüssel, verbleibt auf Karte
- Karte hat eigenen Cryptoprozessor

Unterschlüssel

Wie viele Hauptschlüssel soll ich erstellen?

- Normalerweise reicht ein Schlüssel!
- Hauptschlüssel $\hat{=}$ eigenständige Identität
  - Schlüssel kann mehrere Benutzer-IDs (uid) haben
- Weniger ist mehr: Mehr Schlüssel bedeuten mehr Verwaltungsaufwand (z.B. Key-Signing)[5]
- Mehrere Schlüssel können zur Trennung von privat/beruflich sinnvoll sein

[5] http://security.stackexchange.com/a/29858/19837

Unterschlüssel

Wie viele Hauptschlüssel soll ich erstellen?

- Normalerweise reicht ein Schlüssel!
- Hauptschlüssel $\hat{=}$ eigenständige Identität
  - Schlüssel kann mehrere Benutzer-IDs (uid) haben
- Weniger ist mehr: Mehr Schlüssel bedeuten mehr Verwaltungsaufwand (z.B. Key-Signing)[5]
- Mehrere Schlüssel können zur Trennung von privat/beruflich sinnvoll sein

[5] http://security.stackexchange.com/a/29858/19837

Verfallsdatum von Schlüsseln

- Letzter Ausweg wenn Kontrolle über Schlüssel verloren geht
- Verfallsdatum kann nachträglich verändert werden!

Schlüssel austauschen

- Regelmäßiger Austausch durch neue Schlüssel
- Höhere Rechenleistung benötigt größere Schlüssel

Vertrauensverlust durch Schlüsselwechsel

- Verfällt Primärschlüssel, geht aufgebautes Vertrauen verloren
- Unterschlüssel ohne Verlust neu erzeugt werden

Widerrufszertifikat

- Widerruf nur mit privatem Schlüssel möglich
- Bei Verlust öffentlicher Schlüssel immer noch auf Keyservern!
- **Widerrufszertifikat sofort nach Schlüsselerstellung anlegen!**

- Unbedingt auch Offline-Kopie erzeugen
- Tipp: Widerrufszertifikat als QR-Code drucken

```
$ gpg --armor --gen-revoke [Key-ID] | qrencode
```

Widerrufszertifikat

- Widerruf nur mit privatem Schlüssel möglich
- Bei Verlust öffentlicher Schlüssel immer noch auf Keyservern!
- **Widerrufszertifikat sofort nach Schlüsselerstellung anlegen!**

- Unbedingt auch Offline-Kopie erzeugen
- Tipp: Widerrufszertifikat als QR-Code drucken

```
$ gpg --armor --gen-revoke [Key-ID] | qrencode
```

Schlüsselrichtlinie

- Erklärt Regeln, nach denen beglaubigt wird
- URL wird an jede Beglaubigung angehängt
- Anhaltspunkt für andere, welche Beglaubigungen prüfen
- Mögliche Inhalte
  - Wo gibt es weitere Informationen zum Schlüssel?
  - Wie genau wird geprüft?
  - Welche Bedeutungen haben die einzelnen Beglaubigungsstufen (sig, sig0 - sig3)?

Kenne ich Dich? Das Web of Trust

Was ist Vertrauen eigentlich?

Abb. 2: `http://xkcd.com/1181/`

If you want to be extra safe, check that there's a big block of jumbled characters at the bottom.

Was ist Vertrauen eigentlich?

Abb. 2: `http://xkcd.com/1181/`

If you want to be extra safe, check that there's a big block of jumbled characters at the bottom.

Was ist Vertrauen eigentlich?

Ver|trau|en, das

> *festes Überzeugtsein von der Verlässlichkeit, Zuverlässigkeit einer Person, Sache*[6]

[6] http://www.duden.de/node/676212/revisions/1228784/view

Vertrauen, in was überhaupt?

- **Identität**: Wirklich die vorgegebene Person?
  - Keyserver-Suche nach president@whitehouse.gov[7]
- **Zuverlässigkeit**: Traue ich fremden Beglaubigungen?
  - Grundlage des "Web of Trust"
  - Auch Schlüssel, die man nicht selbst beglaubigt hat, können gültig sein
- ~~**Handeln**: Welche Ziele hat die Person?~~
  - Niemand im Web of Trust bürgt für das Handeln anderer.

[7] `http://pool.sks-keyservers.net:11371/pks/lookup?op=vindex&search=president%40whitehouse.gov`

Vertrauen, in was überhaupt?

- **Identität**: Wirklich die vorgegebene Person?
  - Keyserver-Suche nach president@whitehouse.gov[7]
- **Zuverlässigkeit**: Traue ich fremden Beglaubigungen?
  - Grundlage des "Web of Trust"
  - Auch Schlüssel, die man nicht selbst beglaubigt hat, können gültig sein
- ~~**Handeln**: Welche Ziele hat die Person?~~
  - Niemand im Web of Trust bürgt für das Handeln anderer.

[7] `http://pool.sks-keyservers.net:11371/pks/lookup?op=vindex&search=president%40whitehouse.gov`

Vertrauen, in was überhaupt?

- **Identität**: Wirklich die vorgegebene Person?
  - Keyserver-Suche nach president@whitehouse.gov[7]
- **Zuverlässigkeit**: Traue ich fremden Beglaubigungen?
  - Grundlage des "Web of Trust"
  - Auch Schlüssel, die man nicht selbst beglaubigt hat, können gültig sein
- ~~**Handeln**: Welche Ziele hat die Person?~~
  - Niemand im Web of Trust bürgt für das Handeln anderer.

[7] `http://pool.sks-keyservers.net:11371/pks/lookup?op=vindex&search=president%40whitehouse.gov`

Direktes Vertrauen

Vertraue nur dem, den Du direkt kennst.

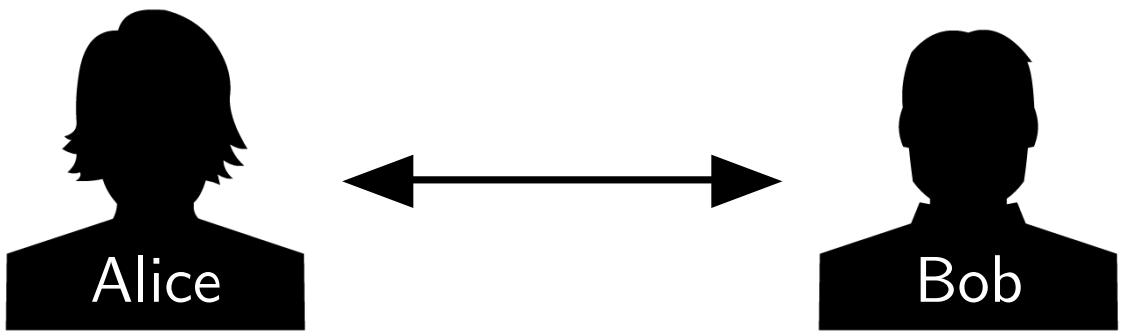


Abb. 3: Direktes Vertrauen

Direktes Vertrauen

Leider explodiert der Aufwand bei Gruppen.

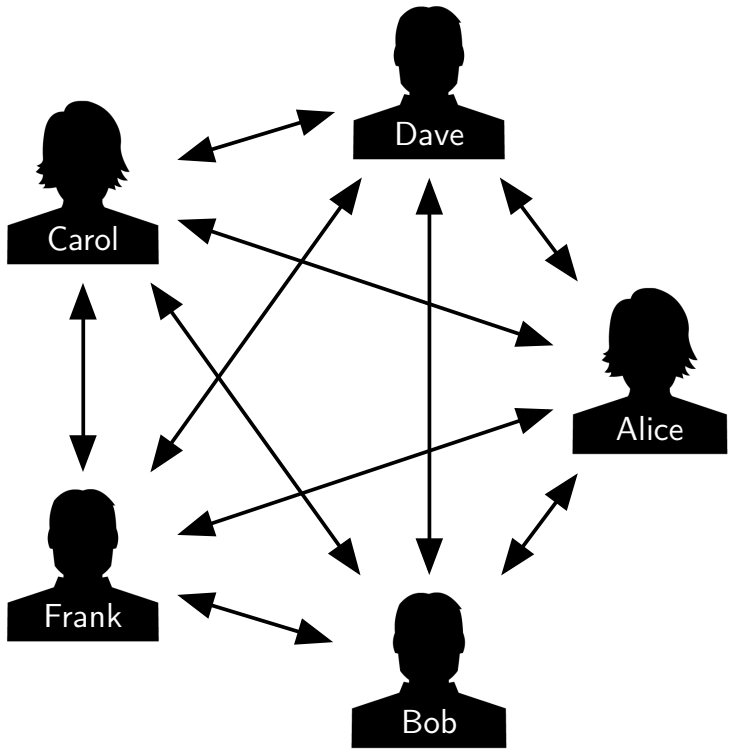


Abb. 4: Direktes Vertrauen in Gruppen

Hierarchische Vertrauenssysteme

Einigen wenigen (hoffentlich) zuverlässigen vertrauen, deren Zertifizierungen ich folge.

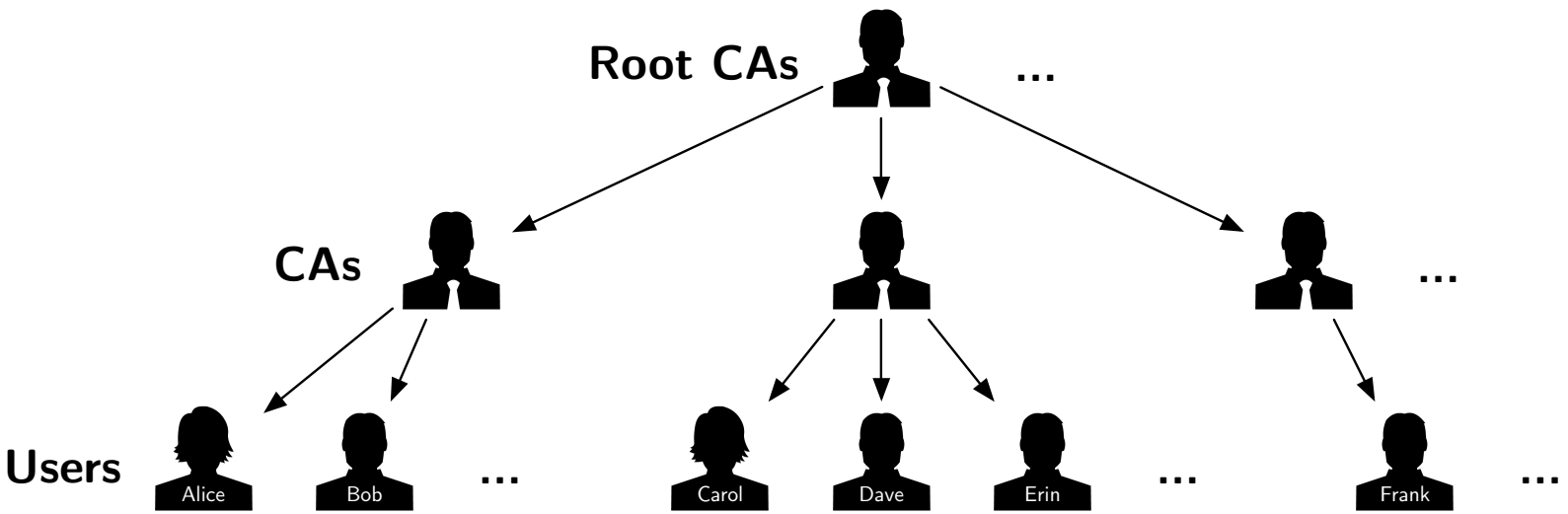


Abb. 5: Hierarchiches Vertrauenssystem

Das Web of Trust

Ziel: Menschen ohne direkten, persönlichen Kontakt vertrauen können.

- Gültiger Schlüssel = Vertrauen in Identität
- Selbst beglaubigte Schlüssel
- Nur hohe Zahl eingehender fremder Beglaubigungen nicht hinreichend

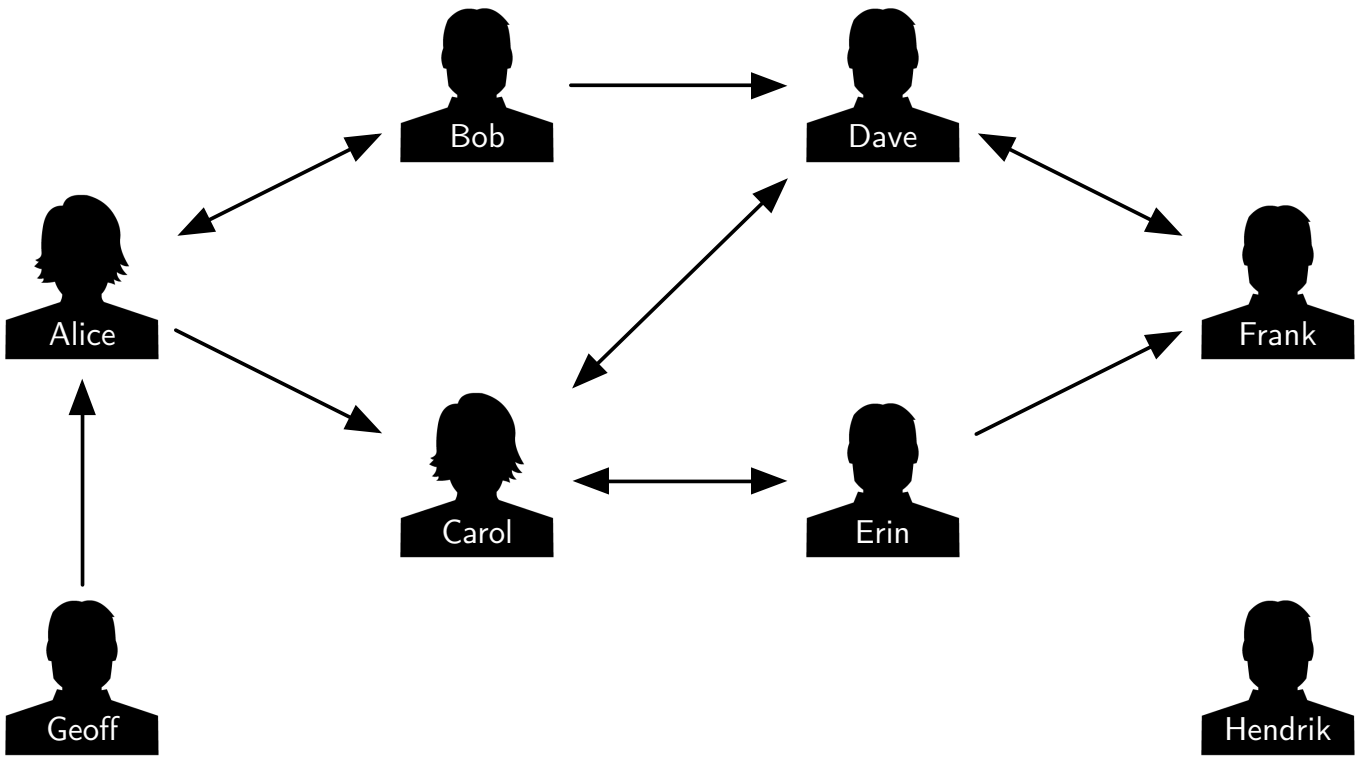


Abb. 6: Vertrauensnetzwerk

Das Web of Trust

- Web of Trust nutzt Vertrauenspfade
- Schlüssel, die Personen beglaubigt haben, denen ich vertraue
- Benötigt Beglaubigungen und **Vertrauens-Bekundung**
  - Beglaubigungen sind öffentlich, Vertrauen lokal

GnuPG-Vertrauensmodell

- Zählt Anzahl Beglaubigungen **gültiger** und **vertrauenswürdiger** Schlüssel
  - **Volles Vertrauen**: Eine Beglaubigung ausreichend
  - **Eingeschränktes Vertrauen**: Drei Beglaubigungen benötigt
- Maximale Tiefe von fünf Schritten

Das Web of Trust

- Web of Trust nutzt Vertrauenspfade
- Schlüssel, die Personen beglaubigt haben, denen ich vertraue
- Benötigt Beglaubigungen und **Vertrauens-Bekundung**
  - Beglaubigungen sind öffentlich, Vertrauen lokal

GnuPG-Vertrauensmodell

- Zählt Anzahl Beglaubigungen **gültiger** und **vertrauenswürdiger** Schlüssel
  - **Volles Vertrauen**: Eine Beglaubigung ausreichend
  - **Eingeschränktes Vertrauen**: Drei Beglaubigungen benötigt
- Maximale Tiefe von fünf Schritten

Gültigkeitsprüfung im Web of Trust

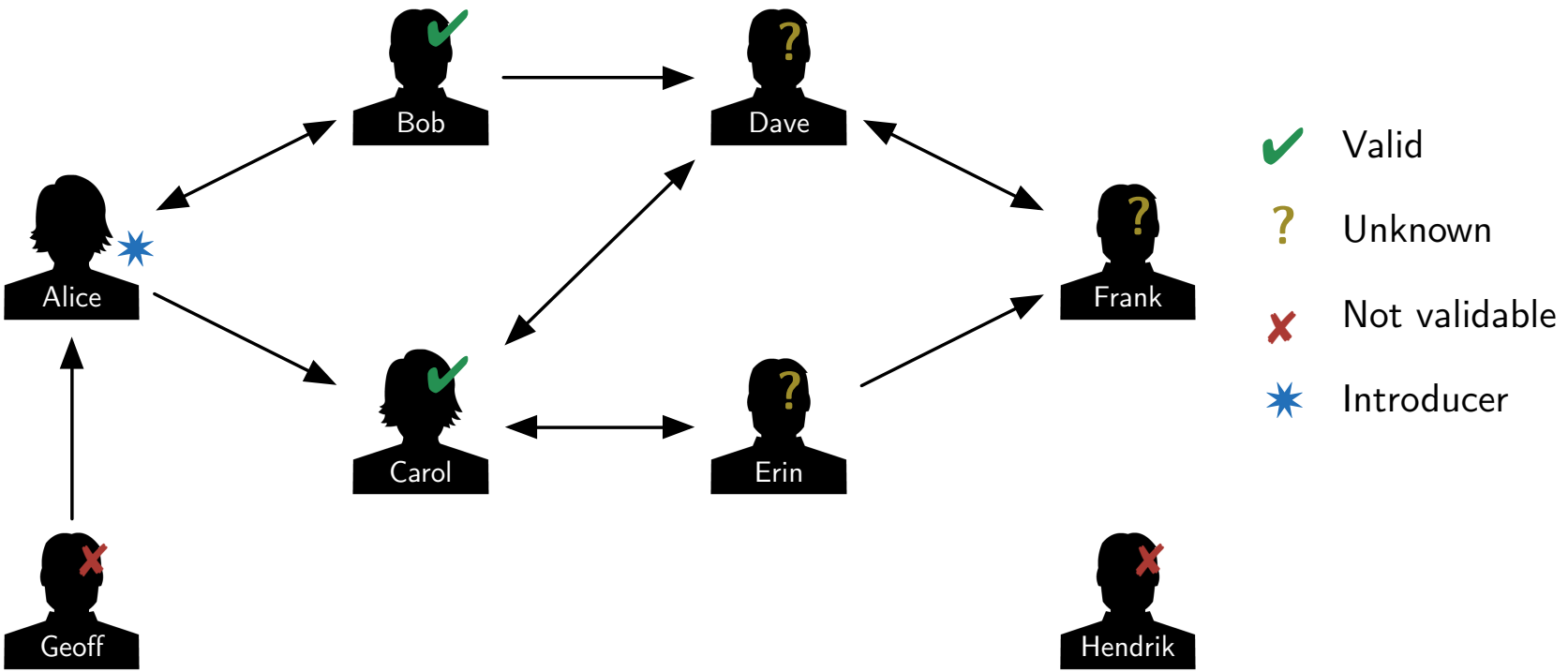


Abb. 7: Alice hat Bob und Carol geprüft

Gültigkeitsprüfung im Web of Trust

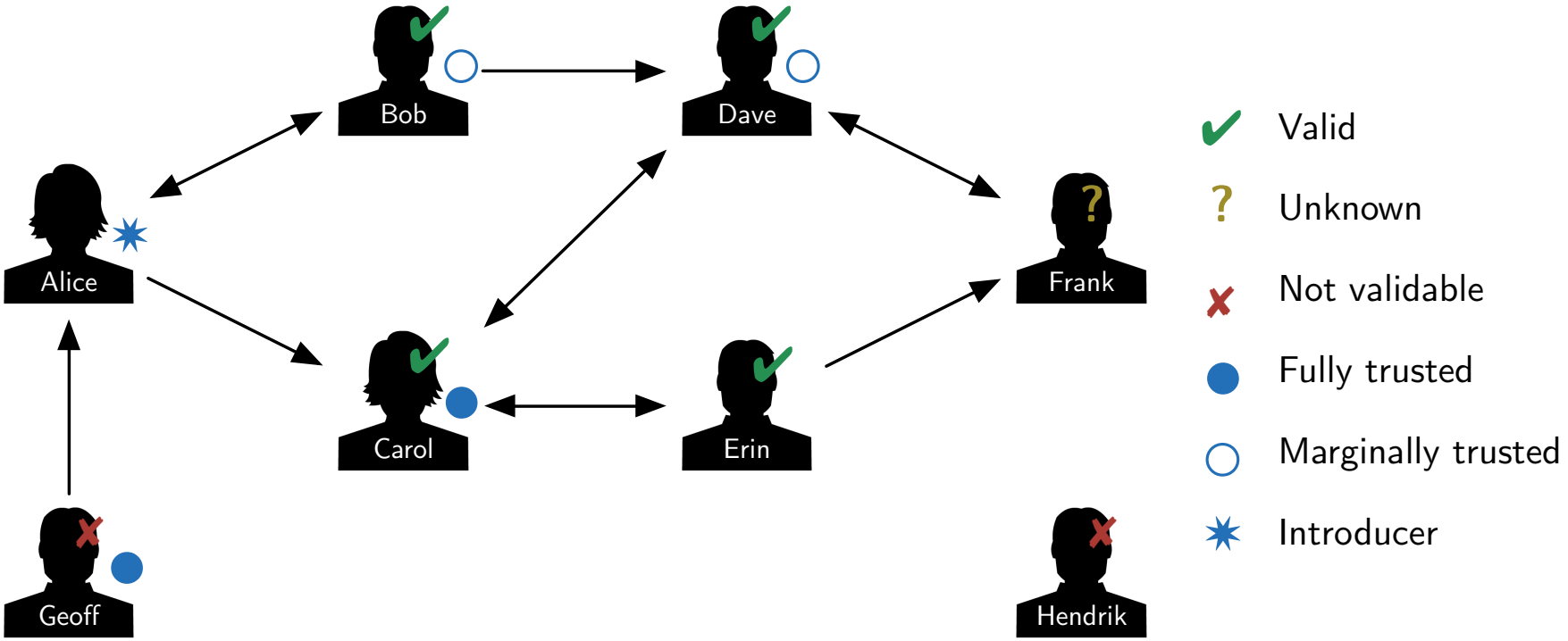


Abb. 8: PGP-Vertrauensmodell: Zwei *teilweise* oder ein *voll* vertraute Signaturen

Fragen

Fragen und Feedback?

Jens Erat

email@jenserat.de

OpenPGP
0D69 E11F 12BD BA07 7B37 26AB 4E1F 799A **A4FF 2279**