# OpenPGP
# Eine Einführung

**Vortragender: Ole Richter**
**Seminar: Electronic Identity**
**Dozent: Dr. Wolf Müller**


19. Dezember 2013


HOW TO USE PGP TO VERIFY THAT AN EMAIL IS AUTHENTIC:
LOOK FOR THIS TEXT AT THE TOP:
REPLY
-----BEGIN PGP SIGNED MESSAGE-----
HASH: SHA256
HEY,
IF IT'S THERE, THE EMAIL IS PROBABLY FINE.

# kurzer Überblick

- PGP steht für Pretty Good Privacy
- Verschlüsseln und Signieren von Nachrichten
- openPGP standardisiert das Protokoll
- wird meist für E-Mails genutzt
- kombiniert Symmetrische und Asymmetrische Verschlüsselung
- Vertrauen der Schlüssel basiert auf dem Web of Trust

# Vorgeschichte

- 1991 schrieb Phil Zimmermann die erste Version von PGP
- Ziel: Verschlüsselung für alle Bürger gegen Geheimdienste etc.
- aufgrund von Exportbeschränkung wurde der Quellcode als gedrucktes Buch exportiert ("PGP Source Code and Internals")
- 1998 wurde openPGP-Standard entwickelt (unter GNU-PGL, im RFC 4880)

# Ver- und Entschlüsselung

- Hybrides Kryptosystem aus symmetrischer und asymmetrischer Kryptographie
  - hohe Geschwindigkeit
  - sichere Schlüsselübertragung

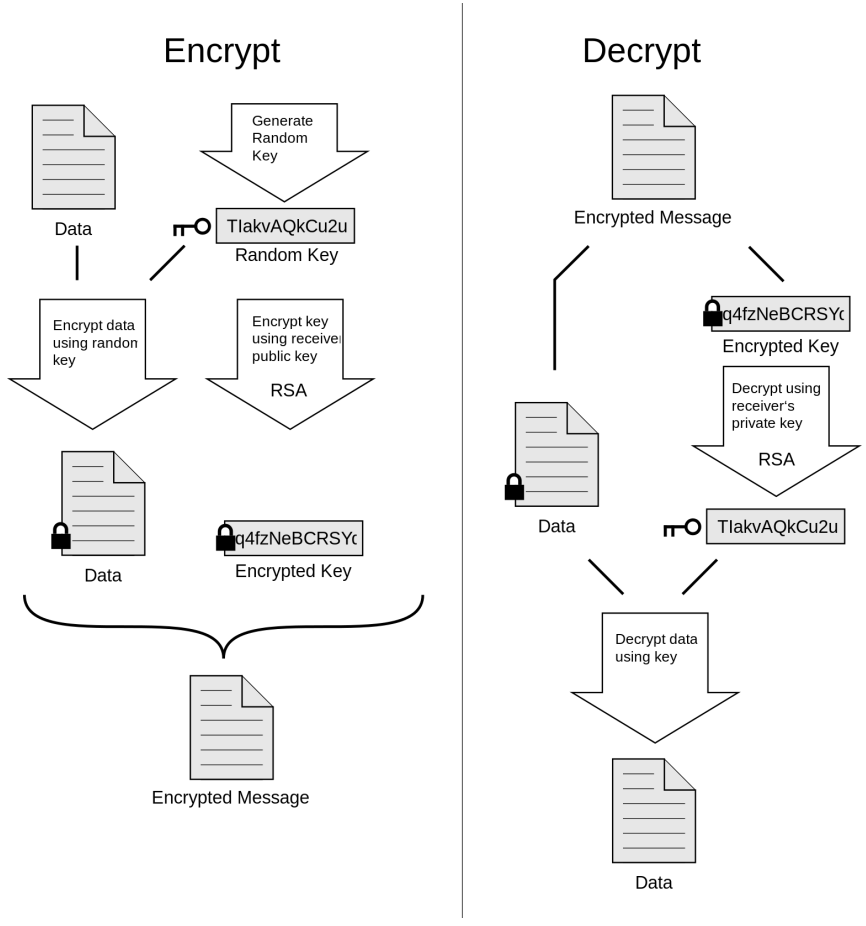


Abbildung 1: `http://commons.wikimedia.org/wiki/File:PGP_diagram.svg`

- Symmetrische Verschlüsselung
  - mehrere Verfahren unterstützt
  - es können Präferenzen angegeben werden
  - 3DES zwingend unterstützt (RFC 4880)
  - AES-128 und CAST5 empfohlen
  - weitere möglich: Blowfish, Camillia usw
- Asymmetrische Verschlüsselung
  - meist unterstützt: RSA, DSA, ElGamal

# Signieren

- dient der Feststellung von Echtheit und Unverfälschtheit der Nachricht
  - Identität des Absenders
  - Integrität der Daten

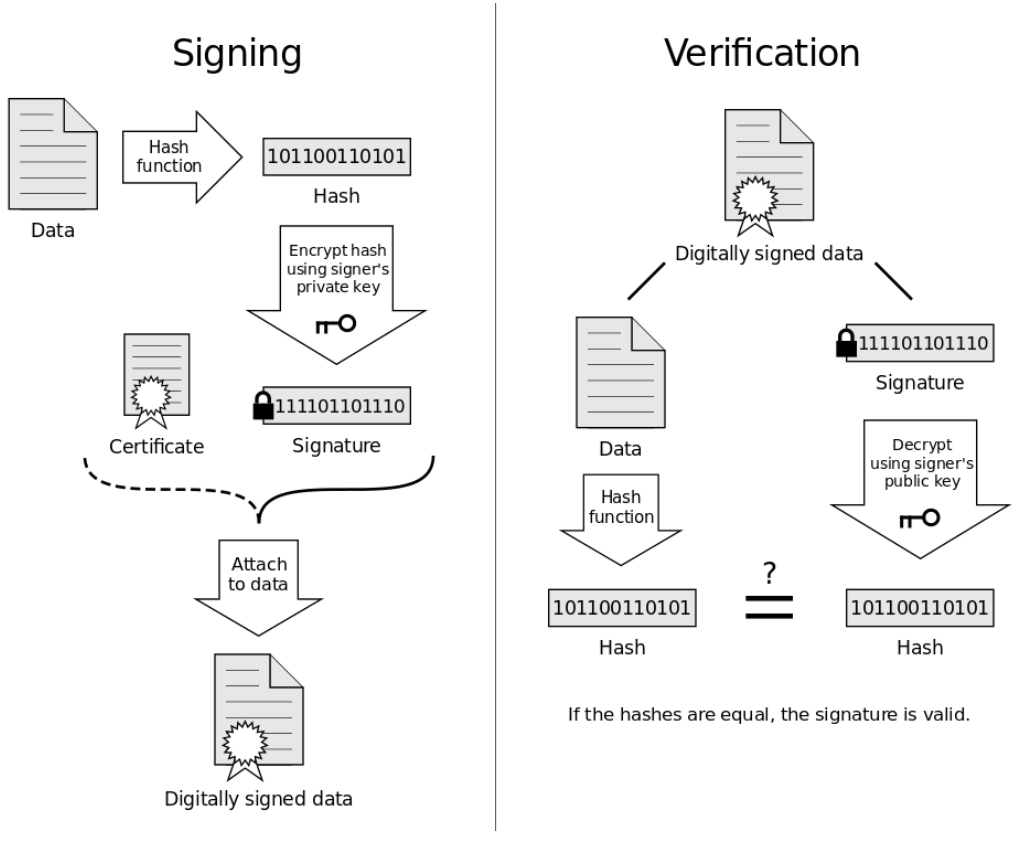


Abbildung 2: `http://commons.wikimedia.org/wiki/File:Digital_Signature_diagram.svg`

- es werden mehrere Hashfunktionen unterstützt
- SHA-1 zwingend unterstützt
- MD5 kann implementiert sein, sollte aber nicht mehr verwendet werden
- weitere Algorithmen: SHA256, SHA512, RIPE-MD/160 usw

# openPGP-Zertifikate

- bestehen aus mehreren Komponenten
  - öffentlicher Hauptschlüssel (Fingerprint bezieht sich auf diesen)
  - User-IDs
  - Unterschlüssel
  - Zusatzinformationen zur Verwendung der Schlüssel
  - Signaturen des Schlüsselbesitzers oder von Dritten (bestätigen oder Widerrufen Echtheit der Komponenten
- werden mit Schlüsselservern synchronisiert

# Beispiel-Zertifikat

```
pub     4096R/4ACDCB5C 2013-12-03 [expires: 2014-12-03]
uid          Peter Griffin (Ein Mensch, wie du und ich)
uid          Peter Griffin (Comment1) <lois@hu-berlin.de>
uid          Peter Griffin (Comment2) <Stewie@lala.org>
sub     4096R/4DDCC78D 2013-12-03 [expires: 2014-12-03]
sub     4096R/32BC8AAD 2013-12-03 [expires: 2023-12-03]
sub     4096R/4CBA1F8D 2013-12-03 [expires: 2015-12-03]
```

- Komponenten können geändert, gelöscht oder hinzugefügt werden
- Dritte signieren die Kombination aus Hauptschlüssel und einzelnen User-ID
- Schlüsselbesitzer signiert alle User-IDs

# User-IDs

- mehrere User-IDs pro Zertifikat möglich
- Aufbau: Vorname Nachname (Kommentar) <E-Mail-Adresse>

# Algorithmen

- zur Auswahl stehen unterschiedliche Algorithmen und Schlüssellängen
- können für Haupt- und Unterschlüssel gewählt werden
- bisher von GnuPG unterstützte Algorithmen
  - RSA
  - DSA
  - ElGamal (kann nur verschlüsseln (in GnuPG))
- in näherer Zukunft wohl auch Elliptic Curve Cryptography
- manche Implementationen unterstützen noch weitere Algorithmen

# Hauptschlüssel

- Hauptschlüssel entspricht eigenständiger Identität
- Hauptschlüssel sollte besonders gesichert werden (externer Datenträger, Smartcard)
- Hauptschlüssel wird nur selten (für Beglaubigungen) benötigt

# Unterschlüssel

- jeder Hauptschlüssel kann mehrere Unterschlüssel haben
- Unterschlüssel sind einzeln widerrufbar
- es können verschiedene Unterschlüssel zum Verschlüsseln und Signieren verwendet werden
- Verfallsdatum:
  - wichtig, da Schlüssellängen angepasst werden müssen
  - kann nachträglich geändert werden

# Widerrufszertifikat

- wichtig, wenn private key kompromittiert wurde
- Widerrufszertifikat muss angelegt werden
- Schlüssel können mit private key widerrufen werden

# Web of Trust

- Glaubhaftigkeit der Schlüssel durch gegenseitige Bestätigung

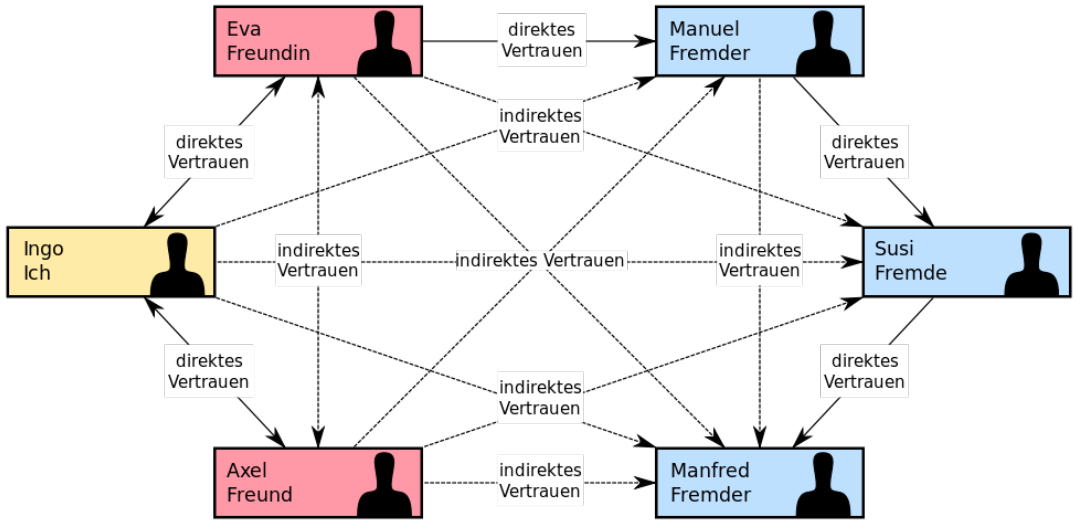


Abbildung 3: `http://commons.wikimedia.org/wiki/File:Web_of_Trust.svg`

- Es gibt 2 unabhängige Arten von Vertrauen:
  - 1. Vertrauen darauf, dass ein Schlüssel authentisch ist, also zur entsprechenden Person gehört
  - 2. Owner Trust: Vertrauen auf die Schlüsselsignatur eines Dritten (bestimmte Vertrauensstufen möglich)
- Schlüssel können direkt ausgetauscht oder auf key-servern gelagert werden

- Schlüsselsignatur kann mit Hinweis versehen werden (Wie genau habe ich die Identität überprüft)
- Stärke des Vertrauens abhängig von:
  - Anzahl der Schlüsselsignaturen
  - Länge des Vertrauenspfades
- Beispiel-Server:
  - `hkp://pool.sks-keyservers.net`
  - `idap://keyserver.pgp.com`

# Tools

- GnuPG
  - Linux: gnupg
  - Windows: GPG4Win
  - Mac: GPGTools
- Email-Programme
  - Tunderbird: Enigmail
  - Outlook: GPGol
  - Apple Mail: GPGMail

# Was kann OpenPGP nicht leisten?

- Sobald der private key kompromittiert wurde, können alle Nachrichten entschlüsselt werden
  - kann durch regelmäßiges erneuern der Unterschlüssel teilweise behoben werden
- Nicht-Abstreitbarkeit
- Meta-Daten werden nicht verschlüsselt
- Privatsphäre durch Web of Trust gefährdet
- Lösung: Off-the-Record Messaging
  - Abstreitbarkeit
  - Perfect Forward Secrecy: wenn private key kompromittiert ist, sind alte Nachrichten weiterhin sicher

# Literatur

J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer.
OpenPGP Message Format.
RFC 4880 (Proposed Standard), November 2007.
Updated by RFC 5581.

Wikipedia.
Gnu privacy guard — wikipedia, the free encyclopedia, 2013.
[Online; seite aufgerufen am 19. Dezember 2013.

Wikipedia.
Phil zimmermann — wikipedia, the free encyclopedia, 2013.
[Online; seite aufgerufen am 19. Dezember 2013.

Wikipedia.
Pretty good privacy — wikipedia, the free encyclopedia, 2013.
[Online; seite aufgerufen am 19. Dezember 2013.