



# Vortrag Keysigning Party

Benjamin Bratkus

Fingerprint: 3F67 365D EA64 7774 EA09 245B 53E8 534B 0BEA 0A13 (Certification Key)

Fingerprint: A7C3 5294 E25B B860 DD3A B65A DE85 E555 101F 5FB6 (Working Key)



# Vortrag Keysigning Party

## Agenda

- Kryptologie
- Kryptosysteme
- „web of trust“
- Gnupg
- Schlüsselerzeugung
- Schlüsselmanagement
- Digitale Unterschriften
- Quellen



# Kryptologie

- Kryptographie:

- Lehre von der Entschlüsselung offener Geheimschriften
- Die Nachrichten werden sichtbar für Dritte übertragen

- Kryptoanalyse:

- Lehre von der Entschlüsselung der Algorithmen

- Steganographie:

- Lehre von den verdeckten Geheimschriften
- Die Nachrichten werden nicht erkennbar für Dritte übertragen

- Kryptologie:

- Kryptographie + Kryptoanalyse



# Kryptologie

- Warum überhaupt Kryptologie ?

- Vertraulichkeit:

- Digitale Nachrichten erreichen diskreteren Status
    - Nachrichten, die nicht für die Öffentlichkeit bestimmt sind

- Integrität:

- Die Unverletztheit der Daten wird gewährleistet
    - die Gesamtheit der Daten wird bestätigt

- Authentizität:

- Die Echtheit der Daten wird bestätigt



# Kryptosysteme

- Symmetrische Kryptosysteme
  - Ein Schlüssel für Ver- und Entschlüsselung
  - Blochchiffren (z.B. DES – Data Encryption Standard)
  - Stromchiffren (RC4, One-Time-Pad)
- Asymmetrische Kryptosysteme
  - Verwendung eines öffentlichen Schlüssels und geheimen Schlüssels
  - Benutzer verwalten einen Schlüsselbund
  - Beispiele: RSA , Elgamal, Rabin



## web of trust

- Idee:

- Digitale Schlüssel durch ein Netz von Bestätigungen (Signaturen) zu sichern

- Problem:

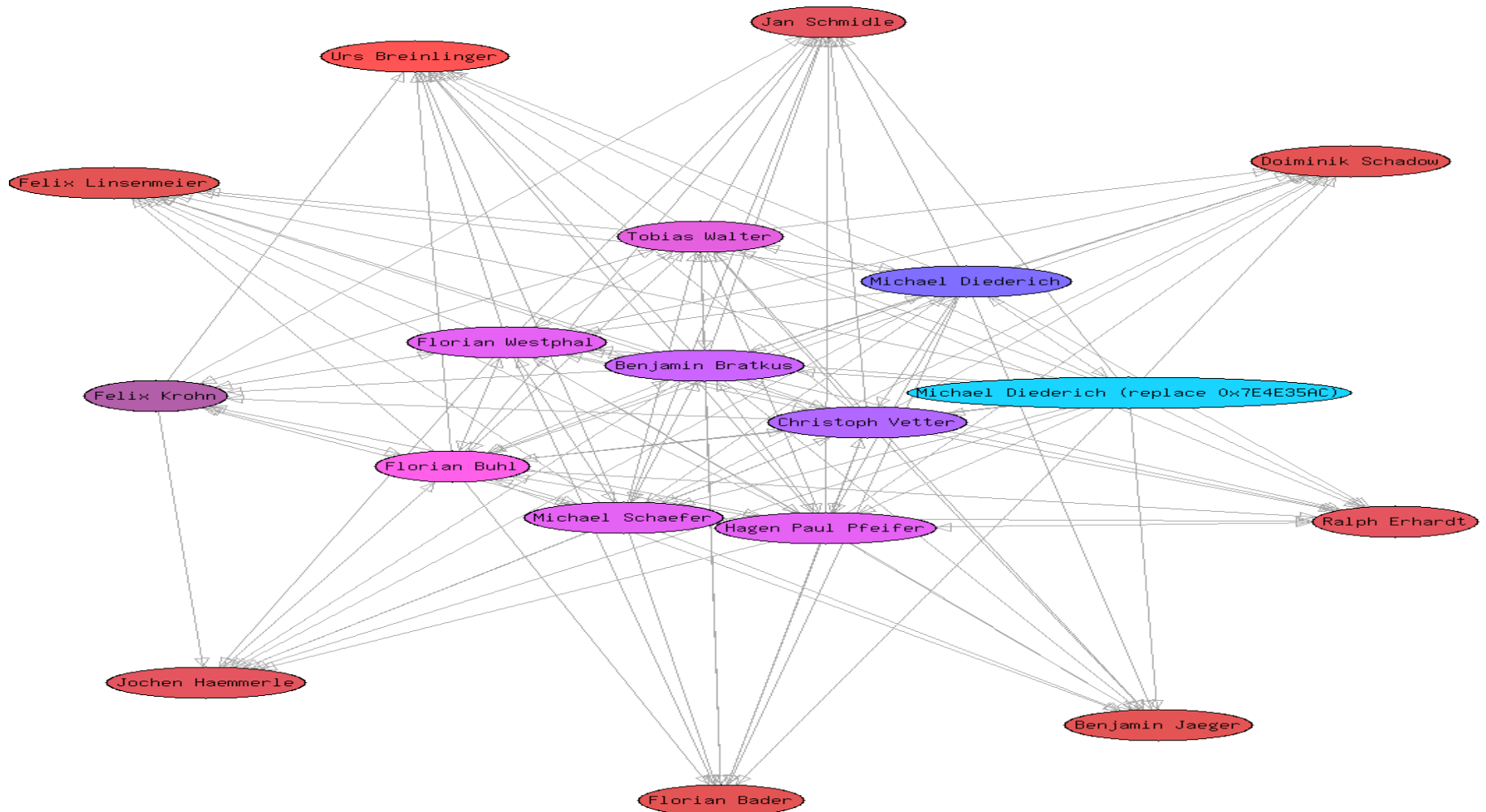
- Kontrolle der Echtheit der angebotenen Schlüssel

- Anwendung:

- Asymmetrisches Kryptosystem
- öffentliche Schlüssel sind für andere Teilnehmer verfügbar
- Kontrolle der Schlüssel durch Bestätigungen (Signaturen)



# GnuPG - web of trust





# GnuPG

- Warum GnuPG?

- Funktionalität auf das Verschlüsseln, Entschlüsseln und Signieren von digitalen Nachrichten beschränkt
- Dateiformat unabhängig
- Richtet sich an nach dem RFC 2440 OpenPGP- Spezifikation
- Freie Software (Quellcode verfügbar, frei von Patenten)
- GnuPG ist nicht durch Ausführungsbestimmungen beschränkt



# GnuPG – Symmetrische Kryptosysteme

- Symmetrische Kryptosysteme

- GnuPG ermöglicht die Verwendung von symmetrischen Kryptosystemen
  - Fokus liegt auf dem Keymanagement – nicht der Geheimhaltung des Algorithmus
  - Es wird versucht einen möglichst großen „key space“ zu erhalten
  - Moderne symmetrische Kryptosysteme wie Blowfish und IDEA werden realisiert
- ! Gefahr: Austausch des einzigen Schlüssels zwischen Kommunikationspartnern



# GnuPG – Asymmetrische Kryptosysteme

- Asymmetrische Kryptosysteme

- Vermeidung der Gefahren bei Schlüsselaustausch

- Trennung der Funktionalität:

- Verschlüsselung mit öffentlichem Schlüssel

- Entschlüsselung mit geheimen Schlüssel

- Moderne asymmetrische Kryptosysteme wie RSA und ElGamal werden realisiert



# GnuPG – Hybride Kryptosysteme

- Hybride Kryptosysteme

- Asymmetrische Verschlüsselung ist kein Allroundmittel
- Symmetrische Verschlüsselung kann sicherer sein
- Lösungsansatz: Mischung von asymmetrischen und symmetrischen Kryptosystemen
- Nachricht wird mit symmetrischem Sitzungsschlüssel verschlüsselt
- Sitzungsschlüssel wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt
- Sitzungsschlüssel wird zusammen mit der Nachricht übertragen



# GnuPG – Erzeugung eines Schlüsselpaares

- Erzeugung eines Schlüsselpaares bei GnuPG:
  - „gpg –gen-key“ Befehl erzeugt ein Schlüsselpaar
  - Eigenschaften des Schlüsselpaares:
    - Typ des Schlüsselpaares (z.B. „DSA“, „ElGamal“)
    - Schlüssellänge (Empfohlen: 2048 Bit)
    - Lebensdauer der Schlüssel
    - Zuordnung der Schlüssel zu Emailadresse(n)



# GnuPG – Schlüsselmanagement

- Übersicht über Schlüssel:
  - „gpg –list-keys“ - liefert eine Übersicht über die verfügbaren Schlüssel im Schlüsselbund
- Übersicht über die Signaturen
  - „gpg –list-sigs“ - liefert einen Überblick über die Signaturen
  - „gpg –list-sigs KeyID“ - liefert Signaturen für den entsprechenden Schlüssel
- Empfangen und Senden von Schlüsseln über einen Keyserver
  - „gpg –send-key KeyID“ - sendet den entsprechenden Schlüssel an den eingetragenen Keyserver
  - „gpg –recv-key KeyID“ - empfängt den Schlüssel von dem Keyserver



# GnuPG – Schlüsselmanagement

- Schlüssel Management

- Schlüssel können von den Servern nicht gelöscht werden!
- Signaturen können nicht geändert werden !
- Schlüssel sollten nicht verloren gehen!
- Schlüssel können nur als ungütig deklariert werden (revoke)
- Speichern von Schlüsseln in einer Datei:
  - „gpg -a -o priv.asc –export-secret-keys“ - speichert die geheimen Schlüssel in der Datei
  - „gpg -a -o pub.asc –export KeyID“ - speichert einen öffentlichen Schlüssel in der Datei
- Importieren eines Schlüssel oder Schlüsselbundes in den eigenen Schlüsselbund:
  - „gpg –import datei.asc“ - importiert den/die Schlüssel in den Schlüsselbund



# GnuPG – Schlüsselmanagement

- Schlüssel Management
  - „gpg –refresh-keys“ aktualisiert alle Schlüssel des Schlüsselbund über den Server
  - „gpg –fingerprint KeyID“ gibt den Fingerprint des gewählten Schlüssels zurück
  - Generelle Einstellungen zu GnuPG werden in der Datei „gpg.conf“ festgelegt z.B:
    - „default key KeyID“ - Defaultschlüssel eintragen mit dem GnuPG arbeitet (notwendig bei mehreren geheimen Schlüsseln)
    - „keyserver“ - Verschiedene Keyserver eintragen für den Schlüsselaustausch



## GnuPG – Dateien verschlüsseln

- Verschlüsseln der Datei mit einem öffentlichen Schlüssel des Schlüsselbundes:
  - „gpg --output datei.gpg --encrypt --recipient KeyID datei“
    - „--output datei.gpg“ – entspricht der verschlüsselten Datei
    - „encrypt“ – Parameter zur Verschlüsselung einer Datei
    - „recipient KeyID“ – der zu verwendende öffentliche Schlüssel
    - „datei“ – die zu verschlüsselnde Datei



# GnuPG – Dateien entschlüsseln

- Entschlüsselung einer Datei mit den geheimen Schlüsseln:
  - „gpg --output datei --decrypt datei.gpg“
    - Geheimer Schlüssel wird benötigt
    - Mantra (Passwort) des geheimen Schlüssels wird benötigt
    - „--output datei“ – entspricht der entschlüsselten Datei
    - „decrypt“ – Parameter zur Entschlüsselung
    - „datei.gpg“ - entspricht der erhaltenen, verschlüsselten Nachricht



# GnuPG – Digitale Unterschriften

- Digitale Unterschriften
  - Generell: Hash-Funktion nutzen um eine Prüfsumme zu bilden
  - Hash-Funktion:
    - keine Dubletten bei der Erzeugung von Hashwerten
    - Rückschlüsse auf das Dokument über den Hashwert sollten schwer sein
  - Digitale Unterschrift erzeugen, durch Anwendung einer Hash-Funktion auf ein Dokument



# GnuPG – Digitale Unterschriften

- Anwendung bei GnuPG:
  - Geheimer Schlüssel wird für Unterschrift verwendet
  - Überprüfung mit dem entsprechenden öffentlichen Schlüssel möglich
  - Dokument kann unverschlüsselt übertragen werden wenn es öffentlich ist
  - Dokument kann zusätzlich verschlüsselt werden wenn es vertraulich ist
  - Unterschriftsprüfung fällt negativ aus, wenn das Dokument nach dem Unterzeichnen verändert wird
  - GnuPG verwendet den „DSA“ - „Digital Signature Algorithm“



# GnuPG – Digitale Unterschriften

- Signierung von öffentlichen Schlüsseln:
  - Notwendigkeit öffentliche Schlüssel dritter zu signieren um das „web of trust“ zu vergrößern
  - „gpg –sign-key KeyID“ - bietet die Möglichkeit öffentliche Schlüssel mit dem eigenen privaten zu signieren
  - Wie groß ist das Vertrauen, Kenntnis des Besitzers des geheimen Schlüssel ?
    - Fingerprint überprüfen
    - Ausweiskontrolle



# GnuPG – Digitale Unterschriften

- Möglichkeiten der Signierung:
  - „0“ - Voreinstellung – Ich antworte nicht. Daten einfach entschlüsseln.
  - „1“ - Es hat keine Überprüfung statgefunden.
  - „2“ - Ich habe es flüchtig geprüft.
  - „3“ - Ich habe es sehr sorgfältig geprüft.



## Was ist mit Windows?

- Werkzeug zur Schlüsselgenerierung und Verwaltung installieren
  - WinPT – Achtung: lokale Speicherung bei Signaturen entfernen !
- EmailClient einrichten
  - Outlook, Pegasus oder Eudora
- Schlüsselbund importieren

Mitmachen !



# Vortrag Keysigning Party

## Quellen:

- [www.gnupg.org](http://www.gnupg.org)
  - Anleitung Gnupg
  - HowTo
  - „man gpg“
- [de.wikipedia.org](http://de.wikipedia.org)
- [www.duden.de](http://www.duden.de)
  - Begriffsdefinitionen
- [winpt.sourceforge.net/de](http://winpt.sourceforge.net/de)
  - Tool zur Benutzung von Gpg unter Windows



# Vortrag Keysigning Party

Vielen Dank für die Aufmerksamkeit.