



*Verschlüsselungssoftware für eine freie, demokratische
Gesellschaft*

Markus Held

markusheld@online.de

Folien von Hagen Paul Pfeifer - hagen@jauu.net

7. Dezember 2006

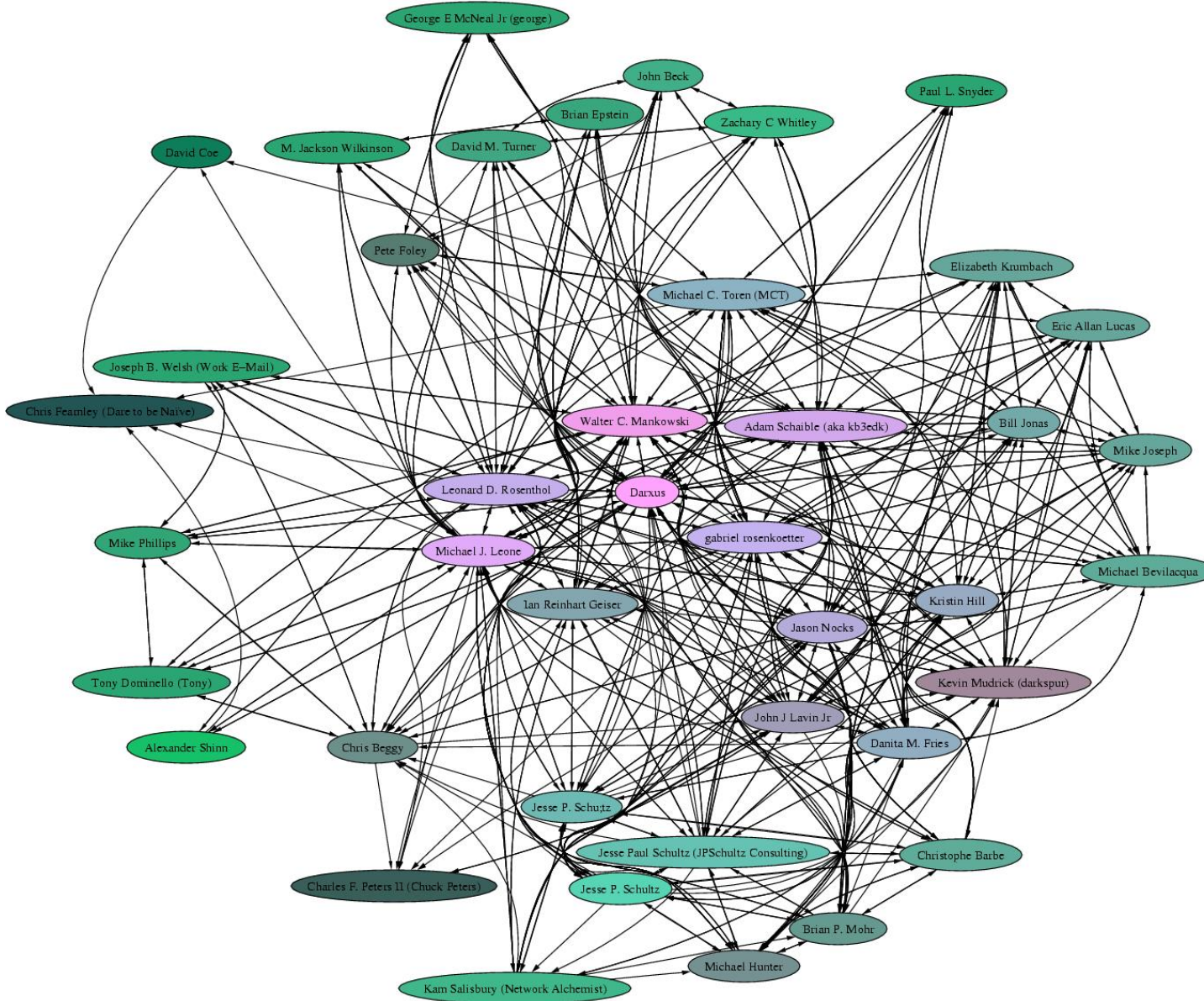
Argumentationsslogans und Signaturen

- ▶ Die EU-Vorratsdatenspeicherung weiß was Du letzten Sommer getan hast!
- ▶ Wer dies liest ist .. gerade gespeichert worden! Schluss mit der EU-Überwachung! Setze dich dafür ein!
- ▶ Nur Gott kennt Deine gesamten Geheimnisse? Jetzt auch die EU! Gegen Vorratsdatenspeicherung!
- ▶ Du hast nichts zu verbergen? Erzähl mal von deinen sexuellen Präferenzen!
- ▶ Niemand hat die Absicht, einen Überwachungsstaat zu errichten.
- ▶ Gegen Terrorhysterie und manischen Kontrollzwang: Vorratsdatenspeicherung abschaffen!
- ▶ Wer nichts zu verbergen hat, hat auch nichts zu bieten. Interessante Leute haben Geheimnisse.
- ▶ Und mehr Slogans die den letzten denkenden Wesen die Augen öffnen gibt es unter: <http://wiki.vorratsdatenspeicherung.de/>

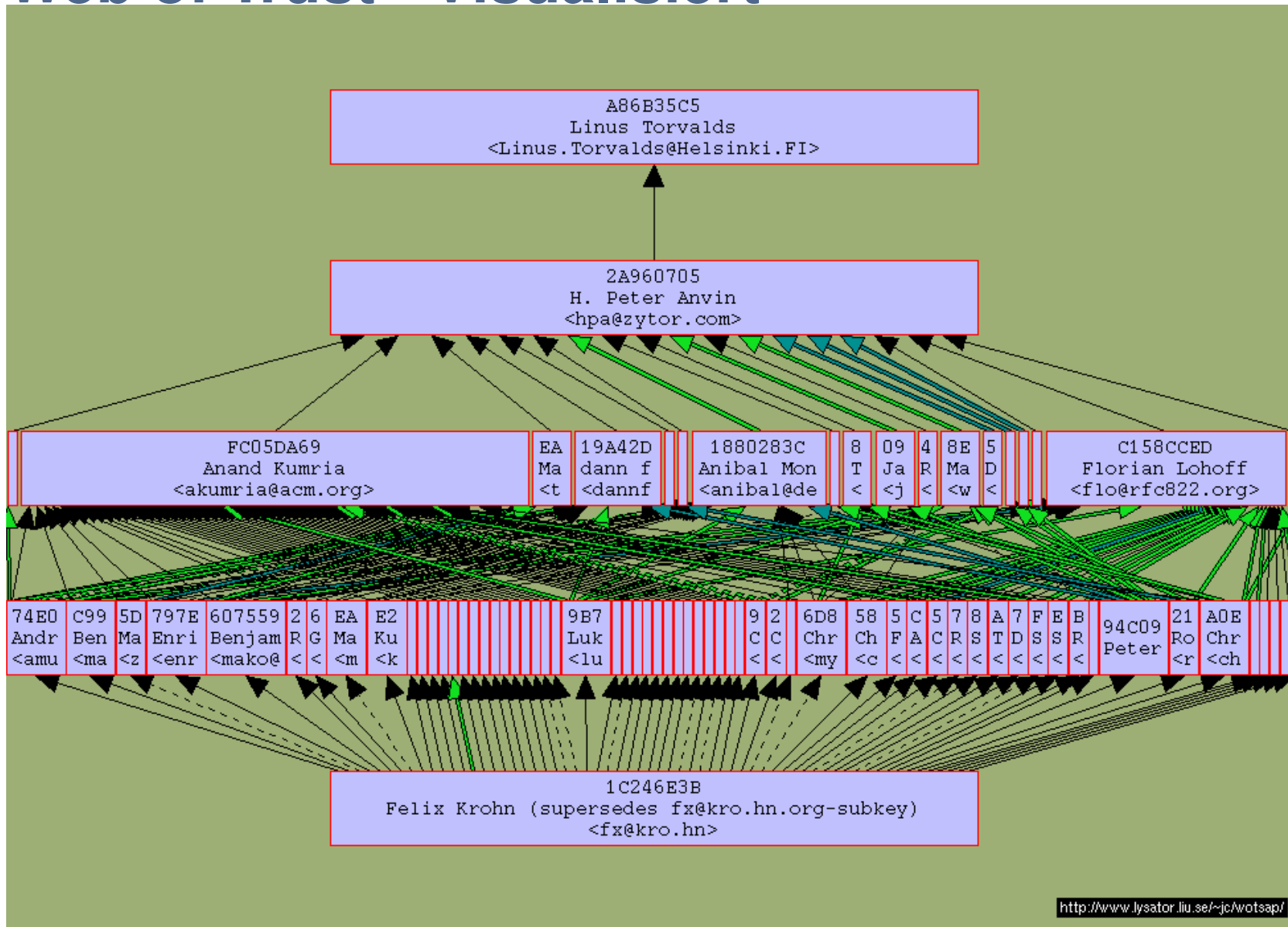
Web of Trust

- ▶ Beschreibt Vertrauensbeziehungen zwischen einer Gruppe von Schlüsseln
- ▶ Schlüssel durch ein Netz aus Bestätigungen sichern (Signaturen)
- ▶ Web of Trust ist Summe aller Vertrauenspfade
- ▶ Vertrauenspfade können in eine oder beide Richtungen verlaufen
- ▶ Kontrolle der Echtheit der Signaturen („korrumpierende Signaturwolke“)
- ▶ Je mehr im Web of Trust involviert sind um so sicherer ist es

Web of Trust - Visualisiert



Web of Trust - Visualisiert



GnuPG

- ▶ Frei Implementierung des OpenPGP Standards (RFC2440)
- ▶ Für verschlüsseln und signieren von Daten mit netten Keymanagement
- ▶ Für Unix Derivate und Windows erhältlich

Projekt Ägypten

- ▶ Unterstützung für S/MIME und X.509v3
- ▶ Plug-In für KMail und Mutt

GnuPG Version 2.0.0

- ▶ S/MIME Support (implementiert CMS und X.509)
- ▶ Funktionalität in Module gesplittet
- ▶ Private Schlüssel und Passphrasen im gpg-agent
- ▶ gpgsm für X.509 Support
- ▶ gpg-agent kann als Ersatz für ssh-agent dienen
- ▶ Unterstützung für Smart-Cards
- ▶ Library Dependencies: libgpg-error, libgcrypt, libassuan und libksba

Verschlüsseln von Datei (symmetrisch/asymmetrisch)

▶ Symmetrisches Verschlüsseln von Dateien:

- `openssl enc -e -a -salt -bf -in secret.jpg -ogpg genkeyut secret.blowfish`
- `openssl enc -d -a -bf -in secret.blowfish -out secret.jpg`

▶ Asymmetrisch:

- `gpg --output datei.gpg --encrypt --recipient ID datei`
- `gpg --output datei --decrypt datei.gpg`

Schlüssel Management

- ▶ Schlüssel erzeugen:
 - `gpg --gen-key`
- ▶ Übersicht über Schlüssel
 - `gpg --list-keys`
- ▶ Übersicht über die Signaturen
 - `gpg --list-sigs`
- ▶ Empfangen und Senden von Schlüsseln über Keyserver
 - `gpg --send-key ID`
 - `gpg --recv-key ID`
- ▶ Schlüssel Updaten
 - `gpg --refresh-keys`
- ▶ Exportieren von Schlüssel

- `gpg -a -o priv.asc export-secret-keys`

- `gpg -a -o pub.asc --export ID`

▶ Importieren von Schlüssel

- `gpg --import datei.asc`

▶ Fingerprint ausgeben

- `gpg --fingerprint ID`

▶ Schlüssel signieren

- `gpg sign-key ID`

▶ Schlüssel bearbeiten

- `gpg --edit-key ID`

Signierungsstufen

- ▶ Unbekannt (q)
- ▶ kein Vertrauen (n)
- ▶ teilweises Vertrauen (m)
- ▶ volles Vertrauen (f)

Ein paar mahnende Worte

- ▶ Wenn der geheime Schlüssel verloren geht ist man verloren - endgültig (wenn der Haustuerschlüssel abhanden kommt ist dies noch nicht einmal halb so schlimm (kann ja neu angefertigt werden))
- ▶ Daten die auf dem Schlüsselservers sind, können nicht wieder gelöscht werden (nur revoked)
- ▶ Revoke Zertifikat erstellen
- ▶ Revoke Zertifikat und privaten Schlüssel auf sicherem Medium und an einem sicherem Ort verwahren
- ▶ Keysigning ist nicht gruscheln bei StussiVZ oder OpenBC – nur wenn wirklich und exakt geprüft wurde wird signiert
- ▶ Passphrase kontra Password

Keysigning Party

1. Ein Schlüssel-Paar erstellen (`gpg --gen-key`)
2. Den öffentlichen Schlüssel an `mike@tty23.net` (Michael Faath) senden
3. Eigene Schlüsseldaten ausdrucken/aufschreiben (ID, Typ, Schlüsselgröße und Fingerprint (`gpg --fingerprint ID`))
4. Zur Party kommen. Mitzubringen ist:
 - a) Ausweisdokument(e)
 - b) Informationen über Schlüssel-ID, Schlüssel-Typ, Hex-Fingerprint und Schlüsselgröße
 - c) Stift
5. Deine Schlüssel-Informationen auf der Party bestätigen
6. Die Schlüssel-Informationen von den anderen auf der Party prüfen (ID, Fingerprint und stimmt die reelle Person mit der Key-Person überein)

7. Alle geprüften Schlüssel signieren (gpg –sign-key ID)
8. Die signierten Schlüssel an den gewählten Keyserver (oder den Schlüssel-Besitzer) zurückschicken

Konfigurationsdatei

▶ `$HOME/.gnupg/gpg.conf`

`verbose`

`no-greeting`

`default-key ID`

`encrypt-to ID`

`throw-keyid`

`auto-check-trustdb`

`expert`

`verify-options show-notations show-policy-urls no-show-photos show-keyserver-urls`

`list-options show-notations show-policy-url no-show-photos show-keyserver-urls show-uid-valid`

`force-v3-sigs`

`escape-from-lines`

`lock-once`

`keyserver wwwkeys.de.pgp.net`

`keyserver-options auto-key-retrieve no-include-revoked no-include-disabled no-honor-keyserver`

Weiterführende Informationen

- ▶ Weiterführende Informationen:
 - Ausführlicher Überblick zu GnuPG
 - Keysigning Party
 - Aktion Stoppt die Vorratsdatenspeicherung
 - <http://www.ronsc.de/misc/signtool/signtool>

Kontakt

- ▶ Markus Held
- ▶ E-Mail: markusheld@online.de
 - Key-ID: 0x67E06383
 - Fingerprint: 307FE9C0C75B89FE58508729F39604467E06383