WOT THE HELL?

E-Mail-Verschlüsselung, S/MIME, PGP, GPG, WoT

Was wir heute tun werden:

- Warum E-Mail unsicher ist.
- Was dagegen gemacht wird.
- SMIME & PGP
- Web of Trust (WoT)
- How-To: E-Mails verschlüsseln
  - SMIME & GPG (& YubiKey)
- Keysigning Party?

Mit wem?

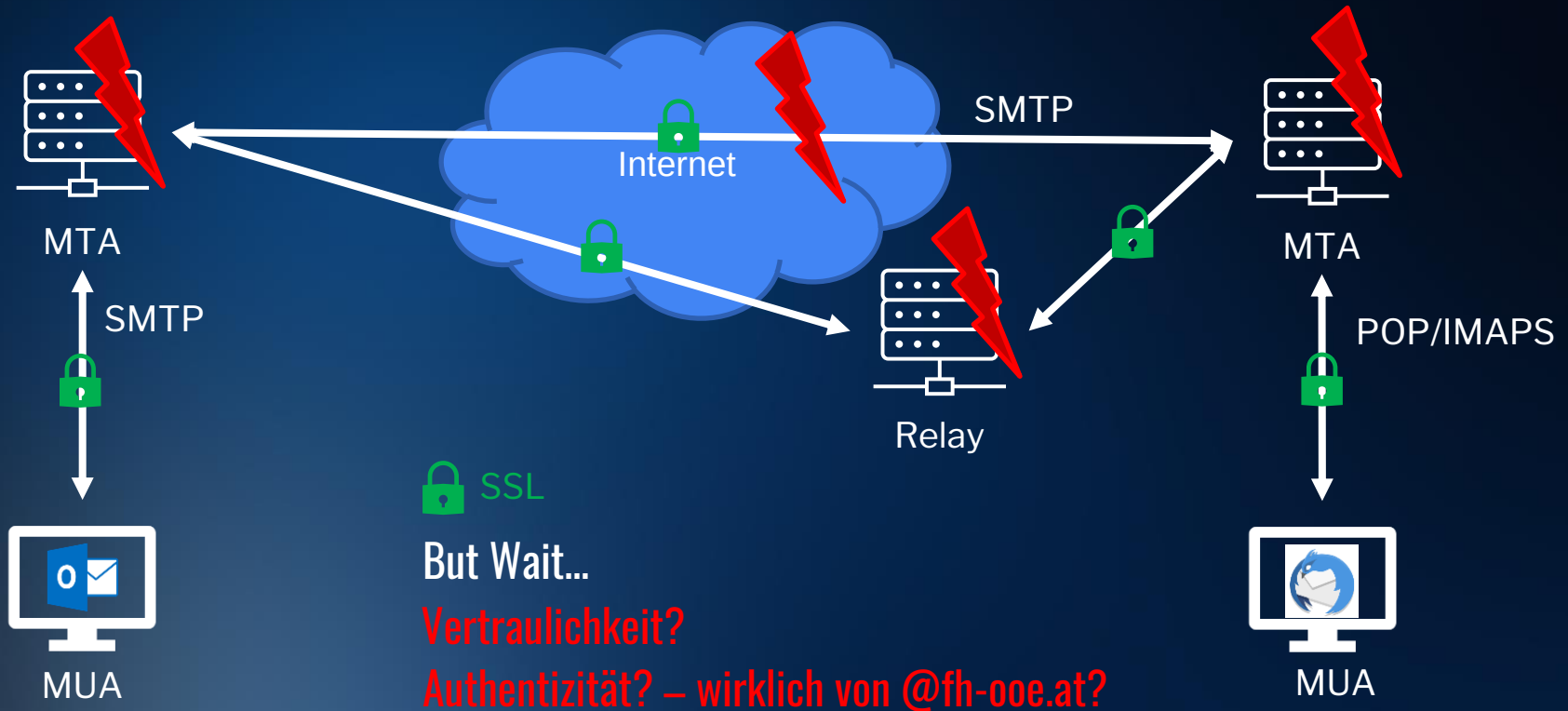
- Kristoffer (Krisi) Dorfmayr

- SIM-Student
- Informationssicherheit im Gesundheitsbereich

Warum Email unsicher ist. - setting the scene

Warum Email unsicher ist. - setting the scene

MUA ... Mail User Agent
MTA ... Mail Transfer Agent



SSL

But Wait...

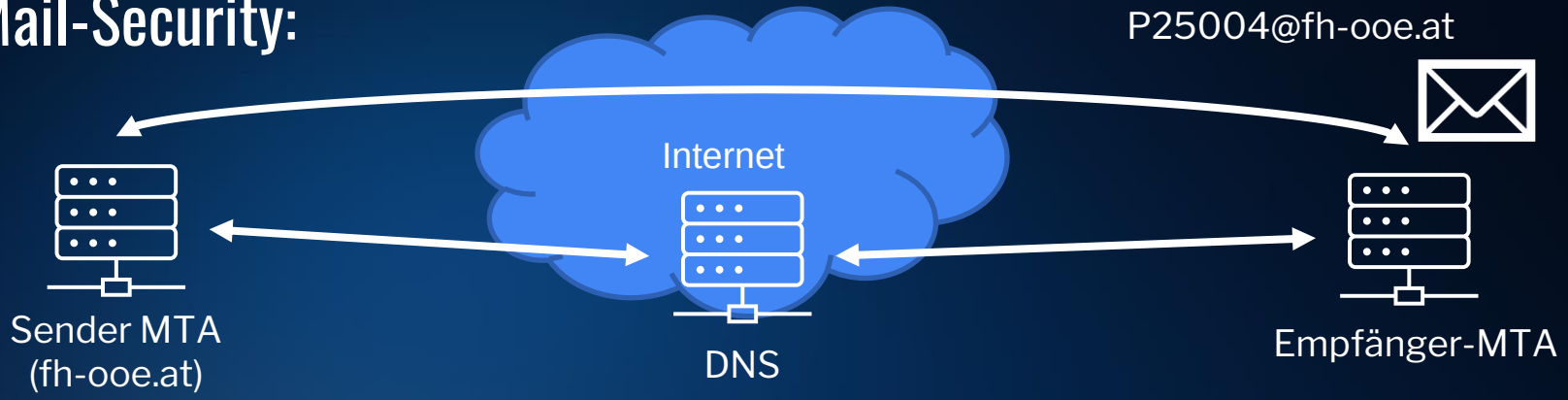
Vertraulichkeit?

Authentizität? – wirklich von @fh-ooe.at?

Mail Security

(SPF, DMARC, DKIM)

Mail-Security:



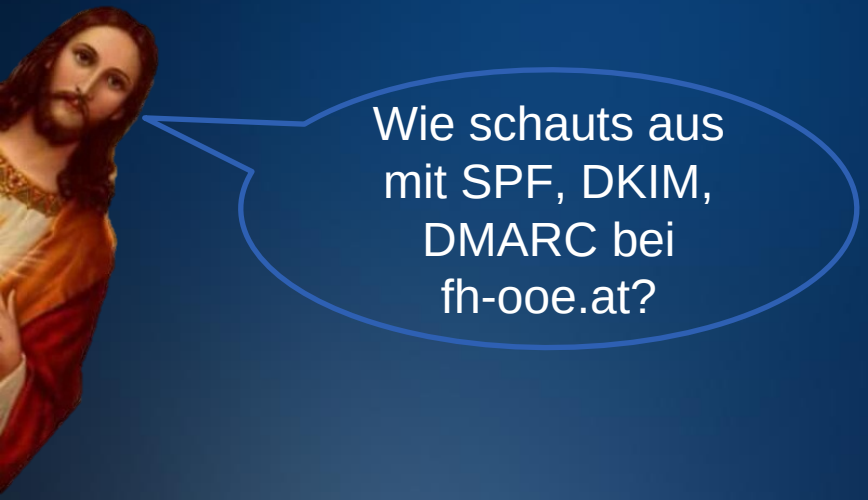

```
                                fh-ooe.at.
@ IN SOA dns1.fh-ooe.at.
@ IN MX  mail.fh-ooe.at
@ mail.fh-ooe.at IN A   78.46.220.229

@ IN TXT "v=spf1 ip4:78.46.220.229 -all"

  mail._domainkey.fh-ooe.at
@ IN TXT "v=DKIM1; k=rsa; p=BEKBAIWKD"

@ IN TXT "v=DMARC1;P=reject;pct=100;rua=m@fhooe.at"
```

SPF
(Sender Policy Framework)

DKIM
(DomainKeys Identified Mail)

DMARC
(Domain-based Message Authentication, Reporting and Conformance)

Mail-Security:

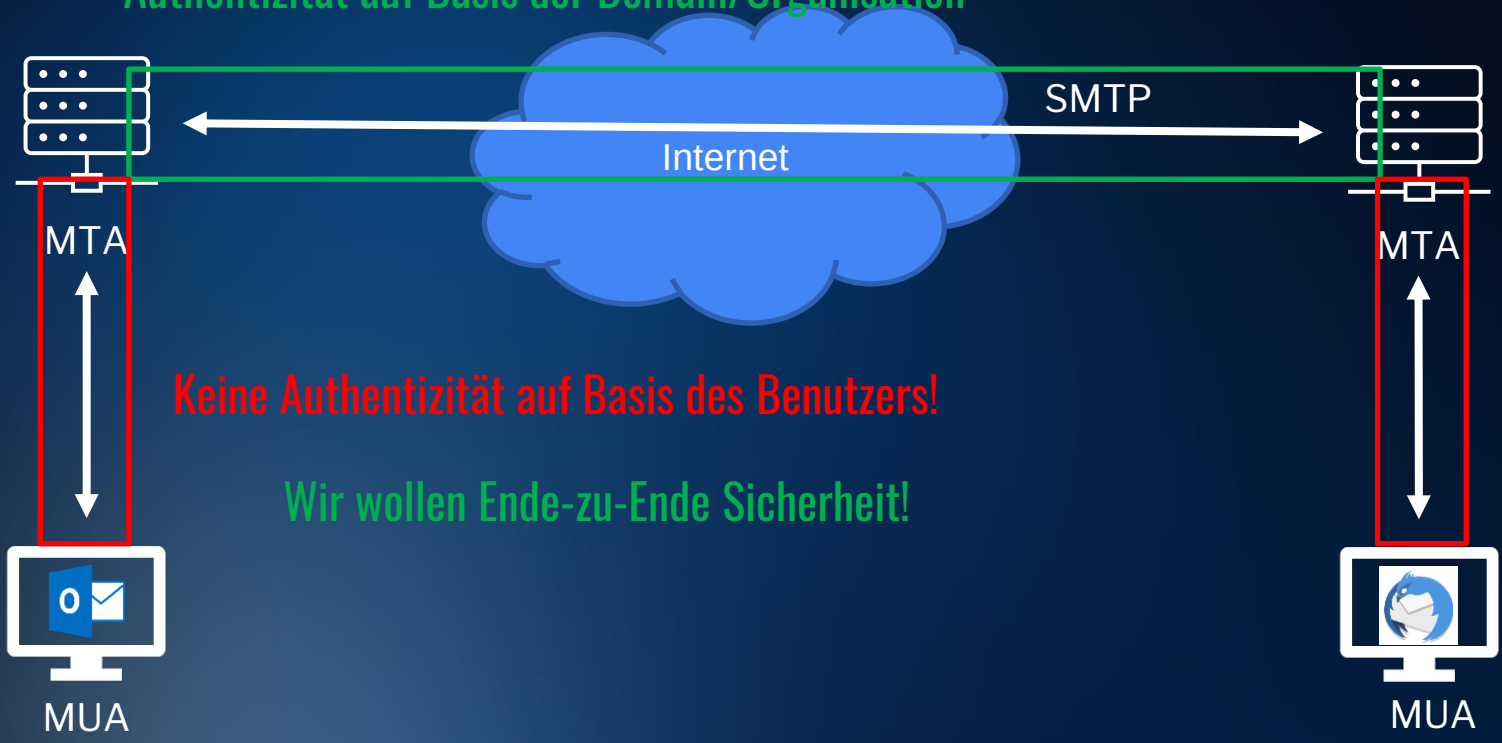


Befehle:

```
dig fh-ooe.at MX
dig fh-ooe.at TXT
dig selector1._domainkey.fh-ooe.at TXT
dig _dmarc.fh-ooe.at TXT
```

Fazit SPF, DKIM, DMARC

Authentizität auf Basis der Domain/Organisation

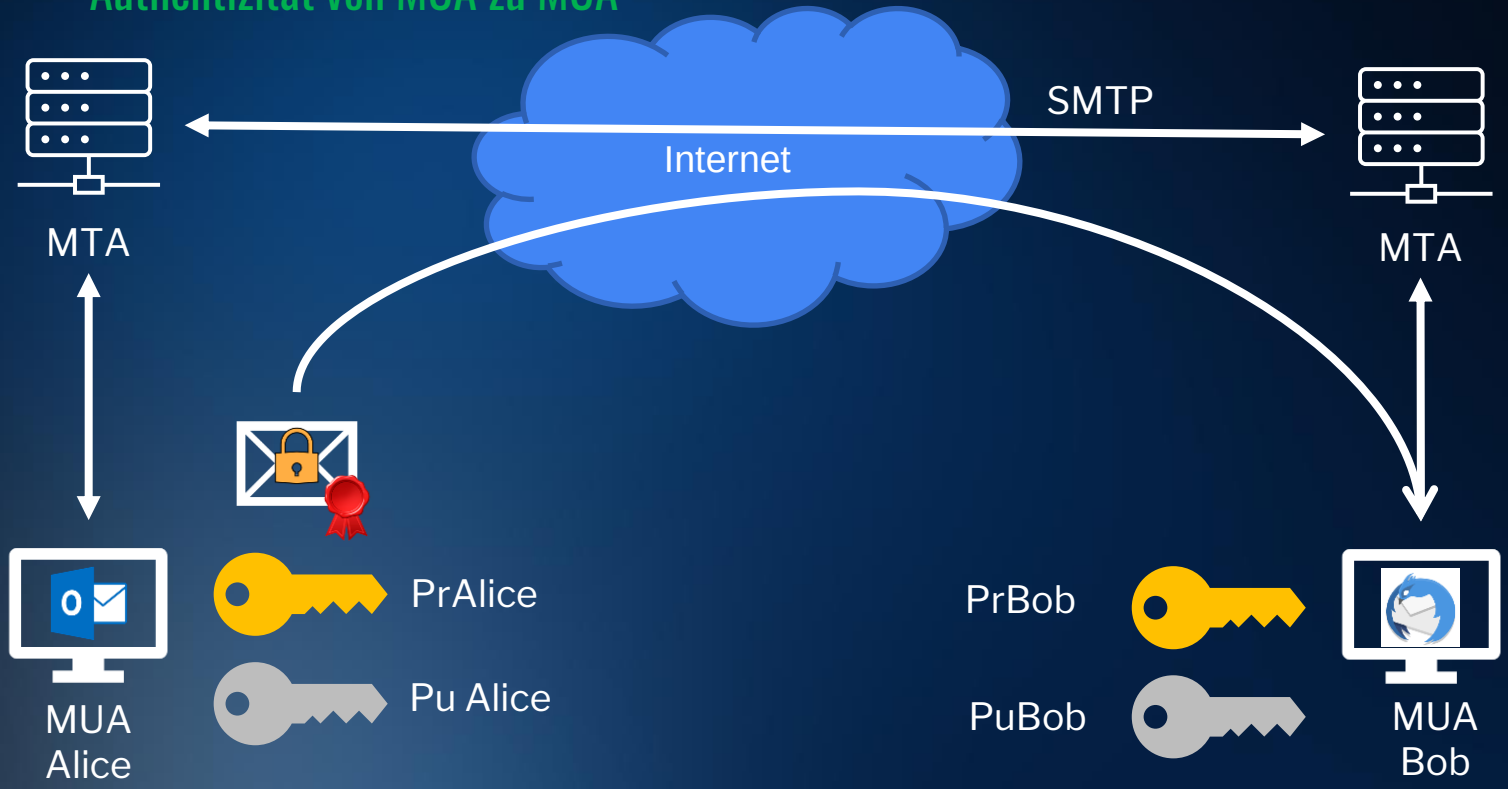


Keine Authentizität auf Basis des Benutzers!

Wir wollen Ende-zu-Ende Sicherheit!

Ende-zu-Ende-Sicherheit

Authentizität von MUA zu MUA



Ende-zu-Ende-Sicherheit Protokolle

- S/MIME (Secure / Multipurpose Internet Mail Extensions)
  - RFC 1847, Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted (seit 1995)
  - Vertraulichkeit, Authentizität und Integrität für E-Mail-Inhalte (Mail Body)
  - Basierend auf Public Key Kryptografie
  - Vertrauensmodell: X.509 Zertifikatssystem (wie bei HTTPS)

- PGP (Pretty Good Privacy)
  - 1991 von Phil Zimmermann vorgeschlagen
  - RFC 4880 (+ 5581 Errata), OpenPGP Message Format (seit 2007)
  - Vertraulichkeit, Authentizität und Integrität für E-Mail-Inhalte (Mail Body)
  - Basierend auf Public Key Kryptografie
  - Vertrauensmodell: Web of Trust (WoT)

S/MIME

S/MIME an der FH

S/MIME - Setup

- Actalis Beantragung
  - https://extrassl.actalis.it/portal/uapub/freemail?lang=en
- Outlook
  - File/Options/Trust Center/Trust Center Options...
  - Nach Erhalt signierter Email: rmc Username/Add to Outlook Contacts
- Thunderbird
  - Edit/Account Settings/End-To-End Encryption/Manage S/MIME Certificates
  - Automatisches Hinzufügen des Zertifikates in den Store bei Erhalt einer signierten Mail

Anatomie S/MIME Zertifikat

Tipps zu OpenSSL Befehlen:
https://help.internetx.com/display/SSL/OpenSSL+-+Die+wichtigsten+Befehle

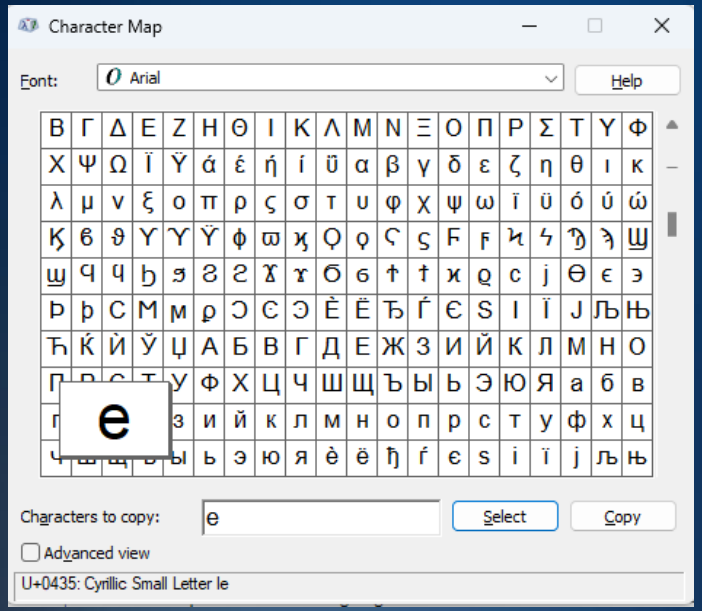
Befehle:
openssl pkcs12 –in <key>.pfx –out actalis.pem
openssl x509 –in actalis.pem -text

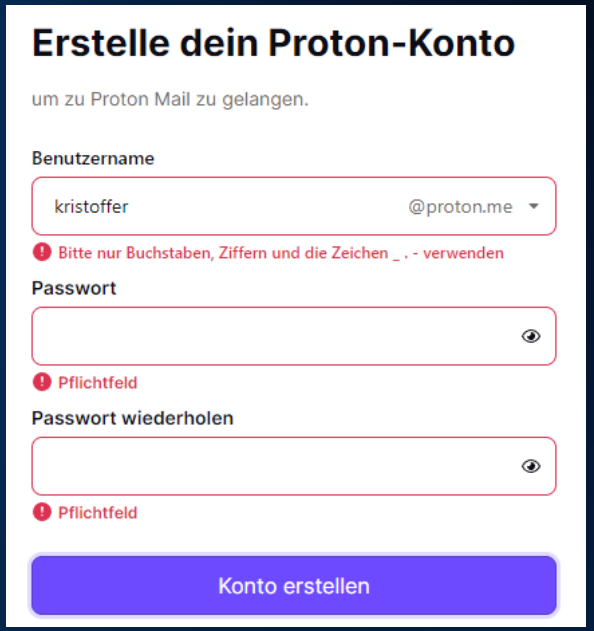
Fazit S/MIME

- Ende-zu-Ende Sicherheit für E-Mails
  - Vertraulichkeit, Authentizität und Integrität für E-Mail-Inhalte (Mail Body)
- Leichtes Setup
  - Signieren von Nachrichten ohne Mitwirken der Empfänger möglich!
- Phishing Problematik
  - Vertrauensmodell: X.509 Zertifikatssystem (wie bei HTTPS)
  - Menschliche Komponente bei Prüfung immer noch im Vordergrund!

Es ist nicht mehr so schlimm wie‘s mal war:

- IDN homograph attack (https://en.wikipedia.org/wiki/IDN_homograph_attack)





OpenPGP

OpenPGP (Pretty Good Privacy)

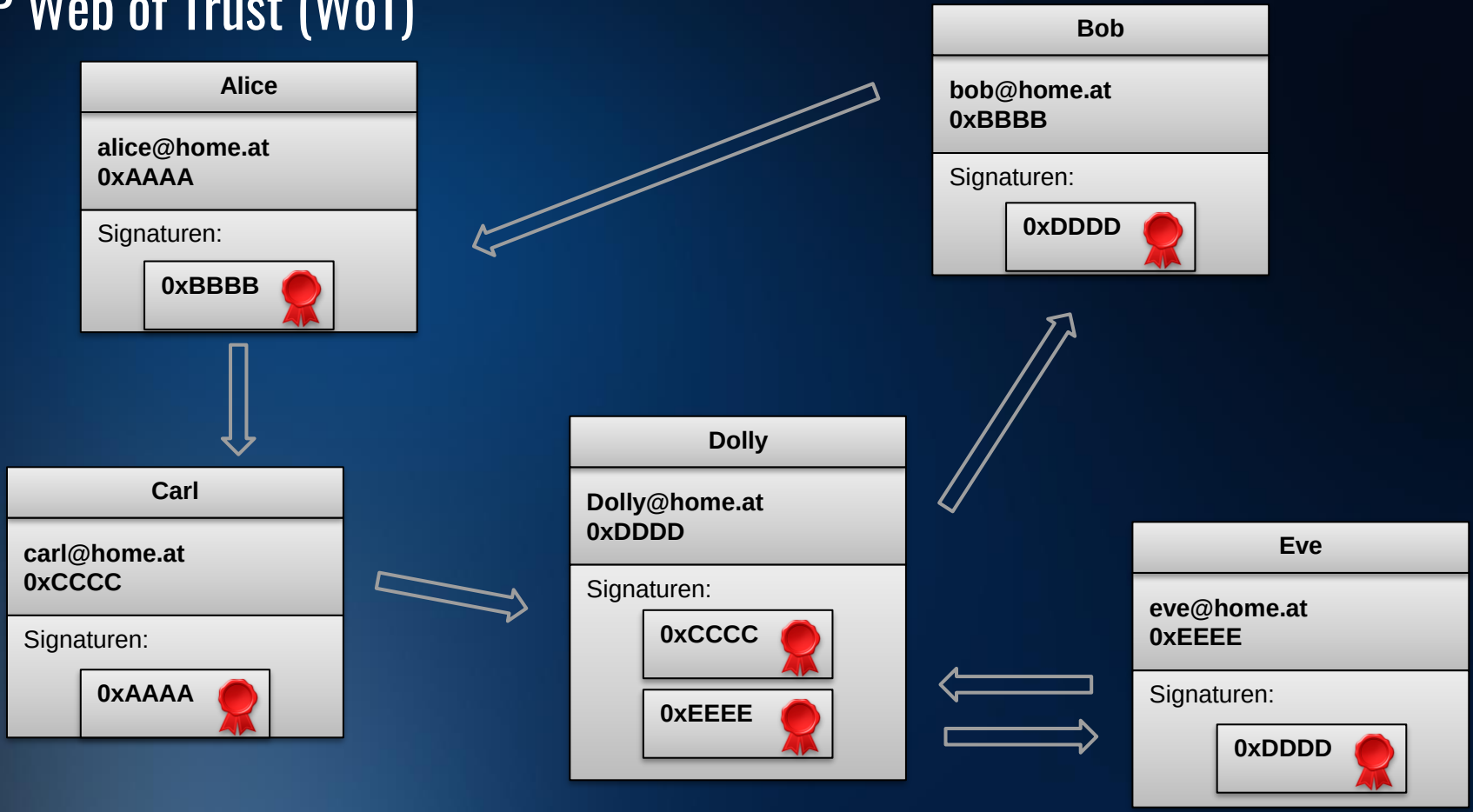
- GnuPG (GNU Privacy Guard)
  - Freie Implementierung des OpenPGP Standards
  - https://gnupg.org/
- Sicherheitsvorteile
  - Erstelle deine Schlüssel selbst
  - Setze das Vertrauen selbst
  - Vertrauensmodell: Web of Trust (WoT)

E-Mail Verschlüsselung

Identität: **alice@home.at** Schlüssel: **0xAAAA**
Signaturen: **0xBBBB**

Identität: **bob@home.at** Schlüssel: **0xBBBB**
Signaturen: **0xAAAA**

PGP Web of Trust (WoT)



PGP - WEB OF TRUST

Projekt Netzwerkanalyse

Ausgangssituation

- Untersuchung von 5 Millionen PGP-Schlüsseln von Kieseberg und Schacht im Jahr 2020 (FH St.Pölten)
  - Beschäftigten sich vorrangig mit Schlüssellängen und Algorithmen
  - Signaturinformationen blieben weitestgehend unbeachtet

- Aktuellste Untersuchung des Web-of-Trusts:
  - Ulrich et al. (Universität Tübingen) im Jahre 2011

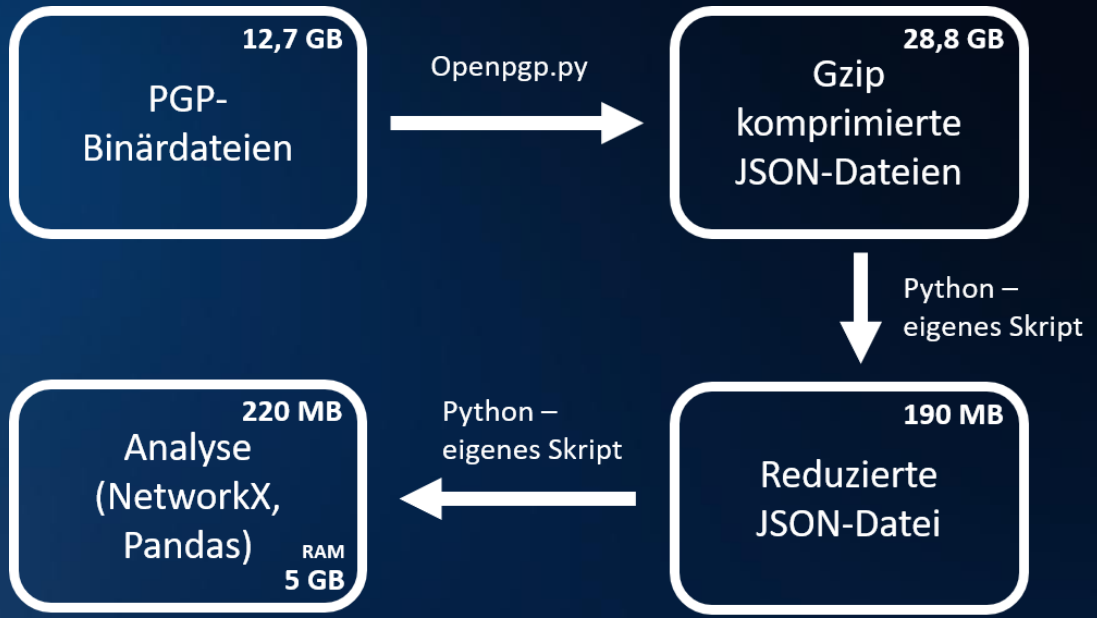
Ziel des Projektes

1. Erstellen und Überprüfen von Hypothesen zum Thema Web-of-Trust.
   Grundlegende Fragen:
   a. Welche Schlüssel signieren welche?
   b. Wie verbreitet ist das Signieren?
   c. Was kann man über die Struktur des WoT sagen?
   d. ...

2. Schaffen einer zentralen Arbeitsplattform, welche die Analyse von PGP-Schlüssel erlaubt
   a. Möglichkeit der Aufbereitung der Schlüssel
   b. Geeignete Speichermöglichkeit des aufbereiteten Schlüsselmaterials
   c. Toolset zur Abfrage der erhaltenen Daten

Parseweg des Schlüsselmaterials

1. Schlüssel online verfügbar: pgp.key-server.io
2. Parser von GitHub diafygi/openpgp-python
3. Reduzierung auf
   a. KeyID (8 Byte),
   b. UserID,
   c. Foreign KeyIDs
4. Analyse in einem Jupyter-Notebook



Hypothesen

Hypothese 1

Das Signieren ist stark verbreitet. Das Signieren gilt als stark verbreitet, wenn mehr als 50% der Schlüssel eine Fremdsignatur aufweisen.

- Das Signieren ist nicht stark verbreitet, da lediglich **9.70%** aller Schlüssel Fremdsignaturen besitzen!

Auszug-Statistik Key Server (Stand März 2021)	
Anzahl Schlüssel	6.055.205
Schlüssel mit Fremdsignaturen	587.547

Hypothese 2

Im Web-of-Trust entstehen Netzwerke, in denen zumindest jeder Teilnehmer einen fremden Schlüssel signiert hat und der eigene Schlüssel von jemand Anderen signiert wurde.

- Im Web-of-Trust gibt es Weakly-Connected-Components, welche somit die Existenz von Netzwerken im WoT bestätigt.

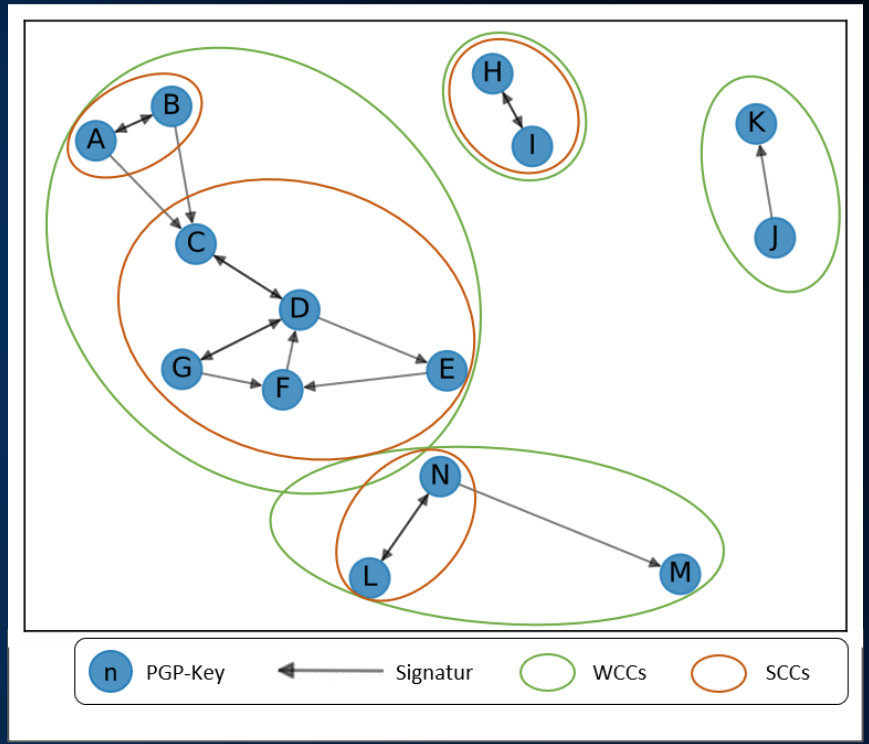
Hypothese 2 - Components

- Im WoT wird zwischen
  - Strongly-Connected-Components(SCC)
  - Weakly-Connected-Components(WCC)

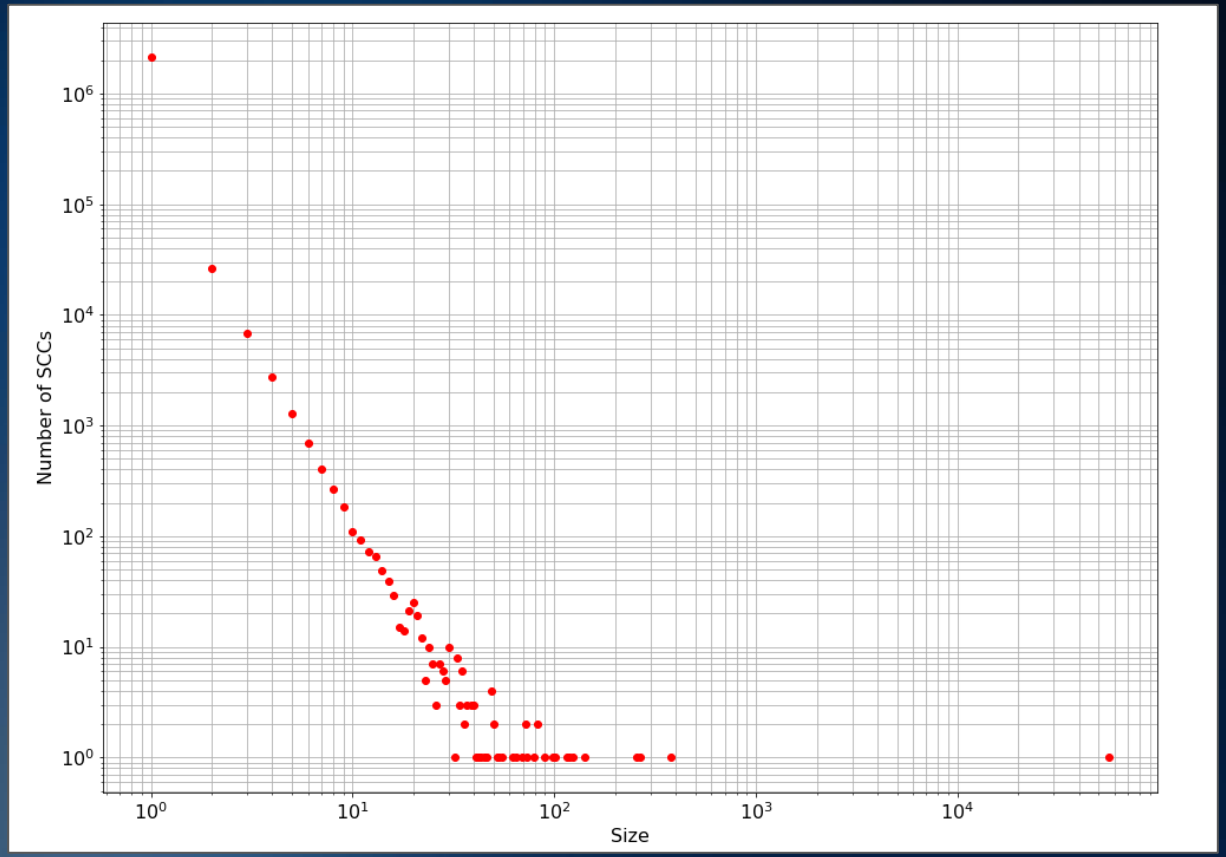
  unterschieden.

- Größe WoT:

Anzahl Schlüssel	2.293.263
Anzahl Fremdsignaturen	3.106.913
Anzahl WCCs	139.915
Anzahl SCCs	2.161.634



Hypothese 2 - Größenverteilung der Strongly Connected Components



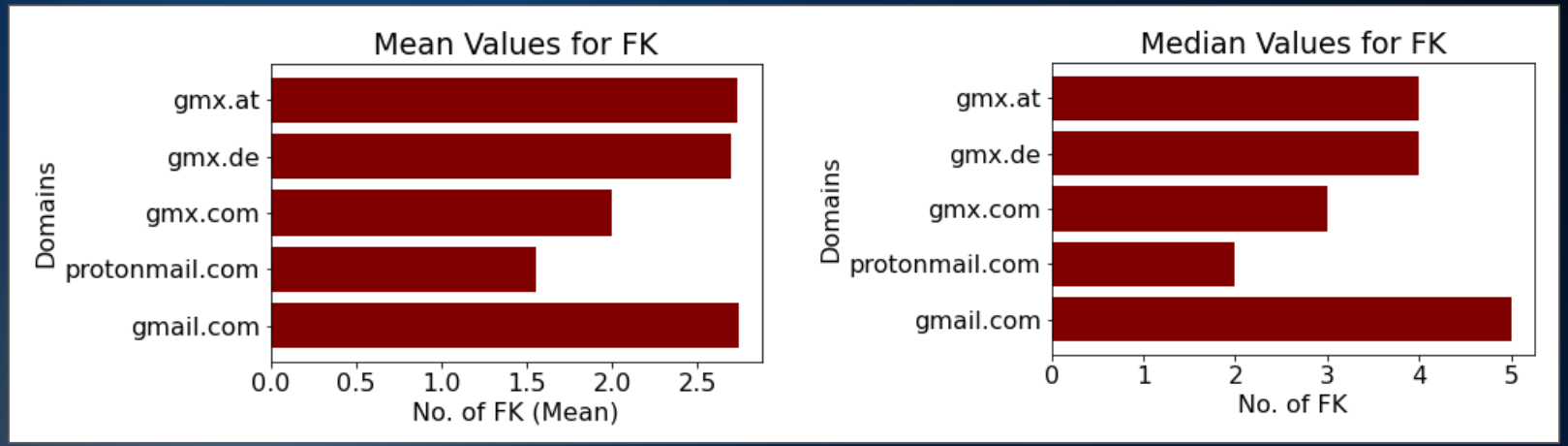
Hypothese 3

Schlüssel, welche Fremdsignaturen aufweisen, werden häufig nur eine Fremdsignatur haben, da OpenPGP nur für einzelne spezielle Beziehungen verwendet wird.

- Insgesamt haben 62,78% aller Schlüssel mit mindestens einer Fremdsignatur genau eine Fremdsignatur.

Hypothese 4

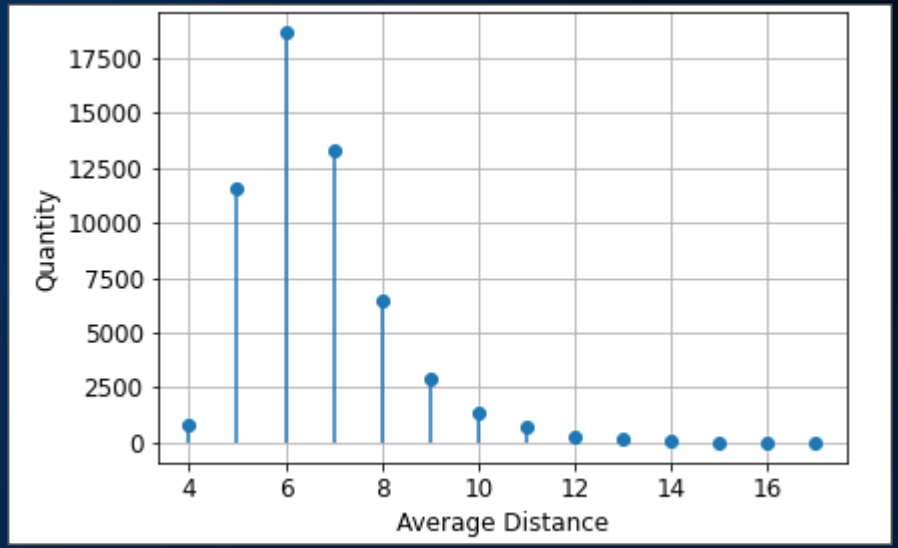
Teilnehmer einer als "sicher" angebotenen Maildomäne (z.B.: Protonmail) weisen einen signifikant höheren Anteil von Fremdsignaturen auf.



Hypothese 5

Die am weitesten verbreitete Implementierung des PGP-Standards „GPG“ schränkt Vertrauensketten mit einer maximalen Distanz von 5 Signaturen ein. Im LSCC können trotz dieser Einschränkung, mehr als 50% der Schlüssel authentifiziert werden.

- Über 50% ist die Mean Shortest Distance > 5
- Hypothese also widerlegt

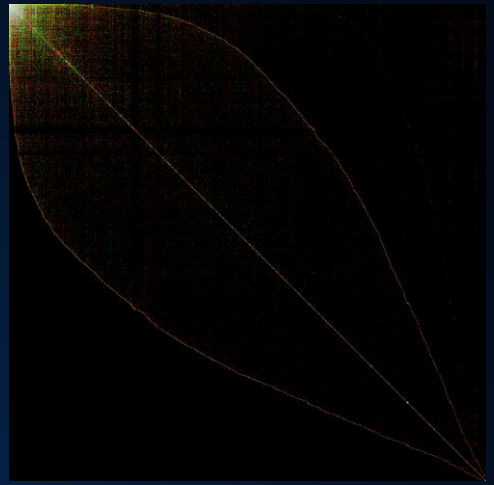


Wie könnte es weitergehen?

- ✓ Parsing der Schlüssel implementiert
- ✓ Einige Hypothesen aufgestellt und analysiert

Wie könnte es weitergehen?

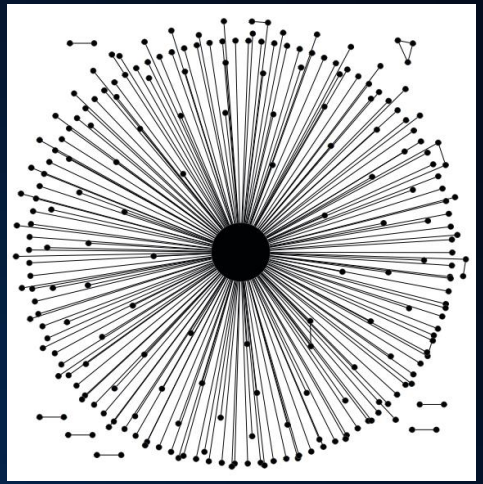
- ✓ Parsing der Schlüssel implementiert
- ✓ Einige Hypothesen aufgestellt und analysiert
- ○ Untersuchungen in Matrix-Darstellung



“Dissecting the leaf of trust”
Jörgen Cederlöf 2004
http://www.lysator.liu.se/~jc/wotsap/leafoftrust.html

Wie könnte es weitergehen?

- ✓ Parsing der Schlüssel implementiert
- ✓ Einige Hypothesen aufgestellt und analysiert
- ○ Untersuchungen in Matrix-Darstellung
- ○ Analyse der Integration des LSCC ins WoT



“Investigating the OpenPGP WoT”
Alexander Ulrich et al. 2011
https://www.researchgate.net/publication/225151405_Investigating_the_OpenPGP_Web_of_Trust

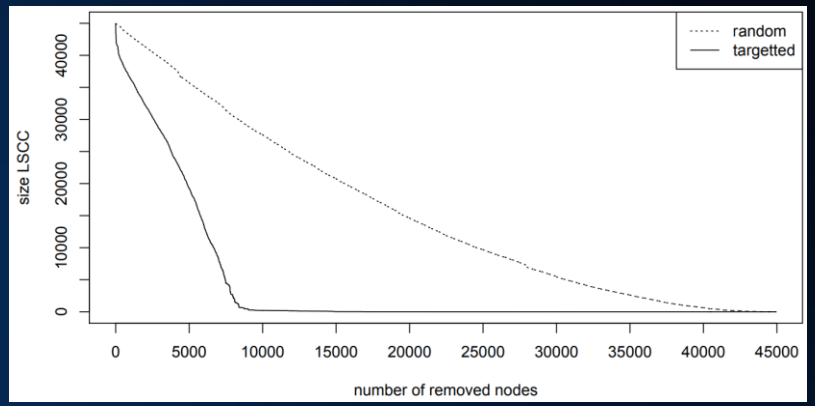
Wie könnte es weitergehen?

- [x] Parsing der Schlüssel implementiert
- [x] Einige Hypothesen aufgestellt und analysiert
- [ ] Untersuchungen in Matrix-Darstellung
- [ ] Analyse der Integration des LSCC ins WoT
- [ ] Deanonymisierung - Wer sind die Big Player?

	Key-ID	Anzahl	User-ID
0	b19c61d61333360c	174611	Yegor Timoshenko <yegortimoshenko@riseup.net>
1	0b7f8b60e3edfae3	170173	Kristian Fiskerstrand <kf@gnupg.net>
2	79be3e4300411886	150172	Linus Torvalds <torvalds@linux-foundation.org>
3	94c32ac158aea35c	150000	Matt Rude <matt@mattrude.com>
4	13de25eed1bb5151	125029	peter@palfrader.org
5	d16c3a41949d203a	100008	Todd Fleisher <fleish@fleetstreetops.com>
6	3b4a0efbbd368329	100006	Gabor Kiss <kissg@uhusystems.com>
7	74bdbff760f08ca2	100003	Pressesprecher des Chaos Computer Club (Chaos ...

Wie könnte es weitergehen?

- [x] Parsing der Schlüssel implementiert
- [x] Einige Hypothesen aufgestellt und analysiert
- [ ] Untersuchungen in Matrix-Darstellung
- [ ] Analyse der Integration des LSCC ins WoT
- [ ] Deanonymisierung - Wer sind die Big Player?
- [ ] Robustheit des Web of Trust



“Investigating the OpenPGP WoT”
Alexander Ulrich et al. 2011
https://www.researchgate.net/publication/225151405_Investigating_the_OpenPGP_Web_of_Trust

Ressourcen? - Kein Problem!

- Mindestens 8 GB RAM (exklusiv für das Projekt)
- Mindestens 50 GB HDD (inklusive Paging xD)

- Berechnungs-Highlights:
  - Initiales Parsing ~ 0,5 Tag (AMD Ryzen 5 2600 6 Core @3,4 GHz; 16GB DDR4 @2800MHz)
  - Key-Reduktion ~ 2 h (AMD Ryzen 5 2600 6 Core @3,4 GHz; 16GB DDR4 @2800MHz)
  - Mean Shortest Distance ~ 3 h (Intel i7-3770 CPU 8 Core @ 3,4 GHz; 16GB DDR3 @ 1600MHz)

Hands On PGP

PGP- Setup

- Schlüssel erstellen
  - YubiKey-Guide: https://github.com/drduh/YubiKey-Guide
  - YubiKey Wiki: https://misteruber.github.io/YubiKeyWiki/docs/OpenPGP/
  - Yubico: https://support.yubico.com/hc/en-us/articles/360013790259-Using-Your-YubiKey-with-OpenPGP
- Outlook
  - Gpg4win
  - https://www.gpg4win.de/
- Thunderbird Setup
  - Builtin
  - https://misteruber.github.io/YubiKeyWiki/docs/OpenPGP/mailverschluesselung.html

PGP Schlüssel

```
Befehle:
gpg –K
gpg --armor --export <email-Id>
gpg --armor --export-secret-keys --output file.asc <ID>
gpgdump file.asc

Verschlüsseln:
gpg --encrypt --recipient <recipient-user-email> <file-name>
gpg --symmetric <file-name>
gpg --decrypt <encrypted-file>
```

WoT Schlüsselserver

- SKS Keyserver
  - Abgeschaltet (Ehemaliges Web of Trust)
  - https://www.golem.de/news/sks-das-ende-der-alten-pgp-keyserver-2106-157613.html
  - Problem der fehlende Löschung und Verifikation
- Keyserver https://keys.openpgp.org/
  - öffentlicher, weltweiter Keyserver
  - nicht verbunden mit dem SKS Keyserver Verbund
  - veröffentlicht Identitätsinformationen nur nach Verifikation der E-Mail-Adresse
  - Standardserver von Thunderbird
  - Unterstützt keine Third Party Signaturen! (https://keys.openpgp.org/about/faq#third-party-signatures)

PGP Schlüssel Signieren

```
Befehle:
gpg --keyserver keys.openpgp.org --armor --export <keyID>
gpg --search-keys <email>
gpg --recv-keys <id>
Gpg --fingerprint <email>
gpg --sign-key --ask-cert-level <id>
```

- How-To Keysigning-Party
  - https://rhonda.deb.at/projects/gpg-party/gpg-party.de.html#toc2
  - https://www.golem.de/news/sks-das-ende-der-alten-pgp-keyserver-2106-157613.html
- Es gibt im Grunde kein Web-of-Trust mehr
  - https://www.heise.de/hintergrund/PGP-Der-langsame-Tod-des-Web-of-Trust-4467052.html
  - Initiative „Rebooting the Web of Trust RWOT“ https://www.weboftrust.info/papers/

WOT THE HELL?

E-Mail-Verschlüsselung, S/MIME, PGP, GPG, WoT