tübix

Have you tried turning it off and on again?

Herzlich Willkommen zum Tübix 2016!

Tuex
Maskottchen

Workshop: E-Mail-Verschlüsselung mittels GnuPG und das Web of Trust

Programm:

- Einführung in GnuPG / Motivation
- Symmetrische/Asymmetrische Verschlüsselung
- Generieren eines Schlüsselpaares auf dem eigenen Rechner
- Web of Trust
- Keyserver

→ *Im Anschluss: Keysigning-Party (16:00 – 17:00 Uhr in Raum V3)*

Was ist OpenPGP?

Standardisiertes Datenformat für verschlüsselte und digital signierte Daten (definiert im *RFC4880*).

Signatur

Merkmal, mit dem man den Unterzeichner einer E-Mail identifizieren kann.
Es kann mit der eigenhändigen Unterschrift auf Papierdokumenten gleichgesetzt werden.

Ziel: Sicherstellung der Korrektheit / Unversehrtheit von Daten *(Integrität)*, Nichtabstreitbarkeit *(Verbindlichkeit)*

Verschlüsselung

Von einem Schlüssel abhängige Umwandlung von *Klartext* in einen *Geheimtext* (*Chiffrat*), sodass der *Klartext* aus dem *Geheimtext* nur unter Verwendung eines *geheimen Schlüssels* wiedergewonnen werden kann.

Ziel: Geheimhaltung von Nachrichten *(Vertraulichkeit)*

🔧 Was ist GnuPG?

Freies Kryptographiesystem, das dem Ver- und Entschlüsseln von Daten sowie dem Erzeugen und Prüfen elektronischer Signaturen dient.

→ Implementiert den OpenPGP-Standard gemäß *RFC4880*.

Kann unter

GNU/Linux
Mac OS X → *GPG Tools* (https://gpgtools.org/)
Windows → *GPG4Win* (https://www.gpg4win.de/), beauftragt vom *BSI*

verwendet werden.

→ Web: https://www.gnupg.org/

Motivation – „Ich habe doch nichts zu verbergen“

Weshalb sollten E-Mails verschlüsselt werden?

Die Offenlegung der Massenüberwachung durch Geheimdienste („*Five Eyes*“, z.B. *NSA* und *GCHQ*) hat gezeigt, dass Informationen und Daten vieler Bürger (grundlos) aufbewahrt und analysiert werden.

„Tempora“

Aktuelles

Das Verschlüsseln von Daten bzw. das Verbergen von Informationen gewinnt durch die Einführung der Vorratsdatenspeicherung wiederholt an Bedeutung.

Ab dem 1. Juli 2017 müssen bestimmte Unternehmen (z.B. Telekommunikations-Provider) diese Speicherpflichten erfüllen. Es wurden aber bereits Klagen angekündigt.

Symmetrische/Asymmetrische Verschlüsselung

🔧 Wiederholung: Verschlüsselung im Allgemeinen

Von einem **Schlüssel** abhängige Umwandlung von ***Klartext*** in einen ***Geheimtext*** (*Chiffrat*), sodass der *Klartext* aus dem *Geheimtext* nur unter Verwendung eines ***geheimen Schlüssels*** wiedergewonnen werden kann.

Man unterscheidet zwischen **symmetrischer** und **asymmetrischer Verschlüsselung:**

Symmetrische Verschlüsselung

Es wird derselbe (geheime) Schlüssel zum Ver- und Entschlüsseln von Daten verwendet. Dieser Schlüssel muss unter Kommunikationspartnern ausgetauscht werden. Hierbei ergeben sich, insbesondere bei der unpersönlichen (digitalen) Übergabe erhebliche Probleme, da sich dieser Austausch erstmalig nicht verschlüsseln lässt.

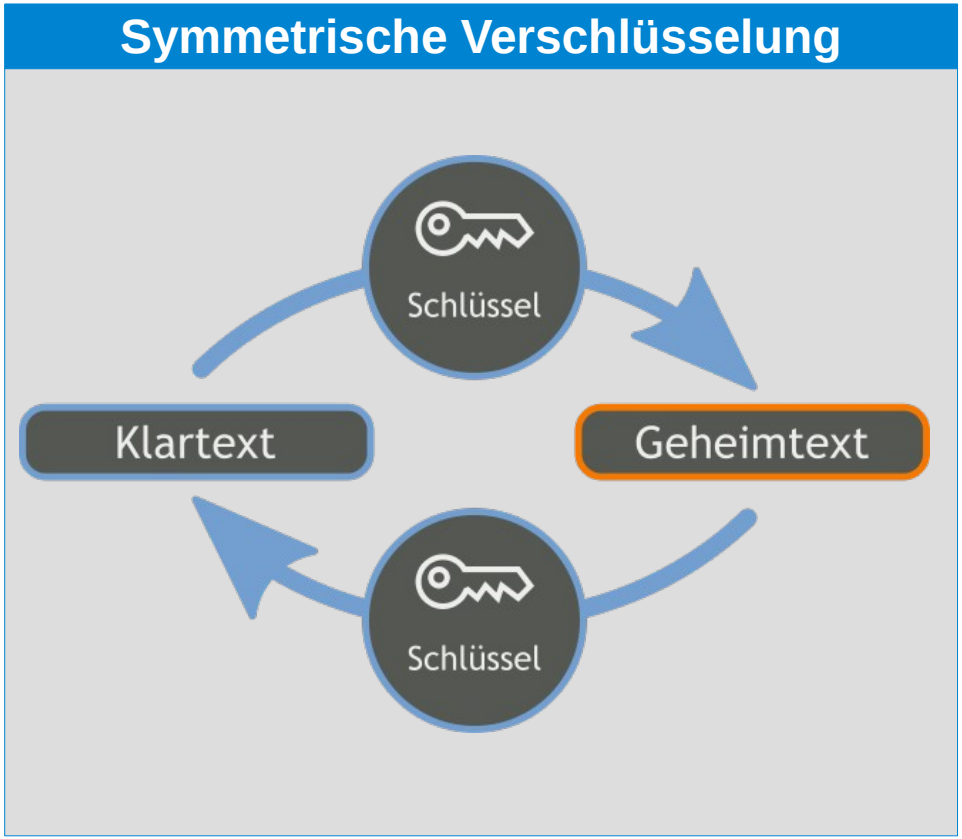
Asymmetrische Verschlüsselung

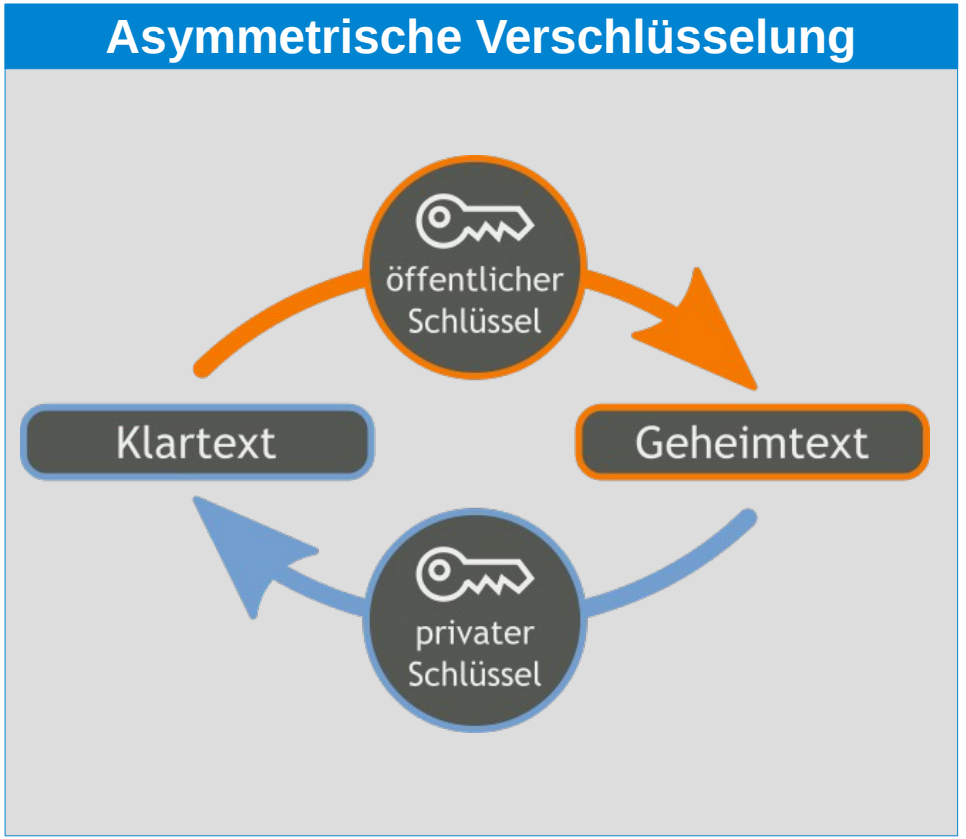
Es wird ein **öffentlicher** Schlüssel zum Ver-Schlüsseln und ein **privater** (geheimer) Schlüssel zum Entschlüssen verwendet. Es bedarf somit keinem ungesicherten Austausch des geheimen Schlüssels.

→ **E- Mail-Verschlüsselung mit GnuPG ist asymmetrisch!**

Symmetrische/Asymmetrische Verschlüsselung

Graphische Verdeutlichung



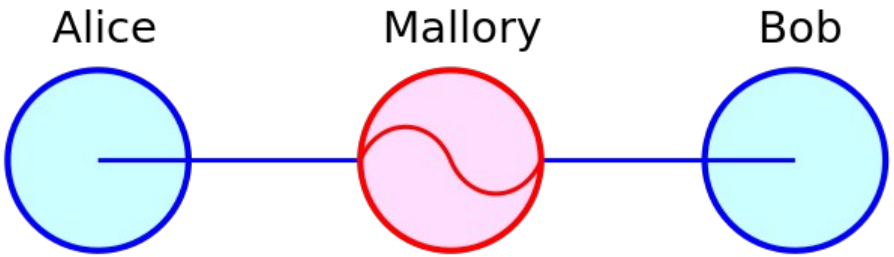


Exkurs: Man-In-The-Middle-Angriff (MitM)

🔧 Was ist ein Man-In-The-Middle-Angriff?

(dt. „Mittelsmannangriff“)

Ein Angreifer steht bei einem *Man-In-The-Middle-Angriff* unbemerkt zwischen zwei Kommunikationspartnern und gibt jedem Kommunikationspartner an, der jeweils andere zu sein (*Janusköpfigkeit*). Dabei hat er mit seinem System volle Kontrolle über den Datenverkehr zwischen den beiden Kommunikationsteilnehmern und kann somit Informationen einsehen und sogar manipulieren (*Kompromittierung*).



→ **Worin besteht der Bezug zur E-Mail-Kommunikation?**

Praxis: Generieren eines Schlüsselpaares auf dem eigenen Rechner

🔧 Generieren eines Schlüsselpaares auf dem eigenen Rechner:

Hinweis: *Wir generieren unsere Schlüsselpaare in diesem Workshop über das Terminal. Es ist auch möglich, dies über eine graphische Oberfläche zu erledigen (Mac OS X → GPG Tools / Windows → GPG4Win). Einige Mail-Clients bieten zudem betriebssystemunabhängige Möglichkeiten der Erzeugung an (Thunderbird → Enigmail).*

```
> gpg --gen-key
> gpg (GnuPG) x.x.x; Copyright (C) Free Software Foundation, Inc.
> This is free software: you are free to change and redistribute it.
> This is NO WARRANTY, to the extent permitted by law.
>
> Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
>    (1) RSA und RSA (voreingestellt)
>    (2) DSA und Elgamal
>    (3) DSA (nur unterschreiben/beglaubigen)
>    (4) RSA (nur signieren/beglaubigen)
> Ihre Auswahl? 1
```

Praxis: Generieren eines Schlüsselpaares auf dem eigenen Rechner

```
> RSA-Schlüssel können zwischen 1024 und 4096 Bit lang sein.
> Welche Schlüssellänge wünschen Sie? (2048) [2048, bei ausreichend
                                              Ressourcen (meist
                                              vorhanden) auch 4096]
>
> Die verlangte Schlüssellänge beträgt 4096 Bit
> Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll.
>     0 = Schlüssel verfällt nie
>     <n> = Schlüssel verfällt nach n Tagen
>     <n>w = Schlüssel verfällt nach n Wochen
>     <n>m = Schüssel verfällt nach n Monaten
>     <n>y = Schlüssel verfällt nach n Jahren
> Wie lange bleibt der Schlüssel gültig (0) 3y
> Key verfällt am Di 11 Jun 2019 17:02:25 CEST
> Ist dies richtig? (j/N) j
>
> Sie benötigen eine User-ID, um Ihren Schlüssel eindeutig zu machen;
  das Programm baut diese User-ID aus Ihrem echten Namen, einem
> Kommentar und Ihrer E-Mail-Adresse in dieser Form auf:
>    „<Vorname> <Nachname> (<Kommentar>) <local-part@domain-part.tld>“
```

i

*Es wird empfohlen, eine **maximale Gültigkeit** von **3 Jahren** zu wählen, denn bei Kenntnis des Schlüssels können alle Nachrichten entschlüsselt werden (**keine PFS!**) Von einer **unendlichen Gültigkeit wird abgeraten**!*

Praxis: Generieren eines Schlüsselpaares auf dem eigenen Rechner

```
> Ihr Name („Vorname Nachname“): <Vorname> <Nachname>
> E-Mail-Adresse: <local-part@domain-part.tld>
> Kommentar: (optional – nicht zwingend erforderlich)
> Sie haben diese User-ID gewählt:
>     „<Vorname> <Nachname> <local-part@domain-part.tld>“
>
> Ändern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden? F
> Sie benötigen eine Passphrase, um den geheimen Schlüssel zu
  schützen. (Passphrase eingeben und wiederholen)
>
> Wir müssen eine ganze Menge Zufallswerte erzeugen. Sie können dies
  unterstützen, indem Sie z.B. in einem anderen Fenster/Konsole
  irgendetwas tippen, die Maus verwenden oder irgendwelche anderen
  Programme benutzen.
>
> Es sind nicht genügend Zufallswerte vorhanden. Bitte führen Sie
  andere Arbeiten durch, damit das Betriebssystem weitere Entropie
  sammeln kann!
> (Es werden noch x Byte benötigt.)
```

Praxis: Generieren eines Schlüsselpaares auf dem eigenen Rechner

```
> gpg: Schlüssel YYYYYYYY ist als uneingeschränkt vertrauenswürdig
  gekennzeichnet
> Öffentlichen und geheimen Schlüssel erzeugt und signiert

  …

> pub 4096R/YYYYYYYY 2016-06-11 [verfällt: 2019-06-11]
>  Schl.-Fingerabdruck =  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
                          YYYY YYYY

> uid    <Vorname> <Nachname> <local-part@domain-part.tld>
> sub 4096R/ZZZZZZZZ 2016-06-11 [verfällt: 2019-06-11]
```

Erzeugten öffentlichen Schlüssel ausgeben

```
> gpg --export -a YYYYYYYY
> -----BEGIN PGP PUBLIC KEY BLOCK-----
> Version: GnuPG vx
>
> (Ausgabe des öffentlichen Schlüssels)
> -----END PGP PUBLIC KEY BLOCK-----
```

Ausgabe im **ASCII-Format** und nicht als Binär-Code *(andernfalls kann es Probleme bei der Ausgabe geben)*

Generieren eines Widerrufszertifikats

```
> gpg --gen-revoke YYYYYYYY
>
> sec 4096R/YYYYYYYY 2016-06-11 <Vorname> <Nachname>
  <local-part@domain-part.tld>
>
> Ein Widerrufszertifikat für diesen Schlüssel erzeugen? (y/N) y
> Grund für den Widerruf:
>     0 = Kein Grund angegeben
>     1 = Hinweis: Dieser Schlüssel ist nicht mehr sichtbar
>     2 = Schlüssel ist überholt
>     3 = Schlüssel wird nicht mehr benutzt
>     Q = Abbruch
> (Wahrscheinlich möchten Sie hier 1 auswählen)
> Ihre Auswahl? <X>
> Geben Sie eine optionale Beschreibung an. Beenden mit einer leeren
  Zeile
> (optional – nicht zwingend erforderlich)
```

Praxis: Generieren eines Widerrufszertifikats

```
> Grund für Widerruf: <Grund>
> (Keine Beschreibung angegeben)
> Ist das OK? (j/N) j
>
> Sie benötigen eine Passphrase, um den geheimen Schlüssel zu
  entsperren. (Passphrase eingeben)
> Benutzer: »<Vorname> <Nachname> <local-part@domain-part.tld>«
> 4096-Bit RSA Schlüsse, ID YYYYYYYY, erzeugt 2016-06-11
>
> Ausgabe mit ASCII Hülle erzwungen
> Widerrufszertifikat wurde erzeugt.
>
> (Hinweis)
> -----BEGIN PGP PUBLIC KEY BLOCK-----
> Version: GnuPG vx
> Comment: A revocation certificate should follow
>
> (Ausgabe des Widerrufszertifikats)
> ---END PGP PUBLIC KEY BLOCK-----
```

i

*Es wird empfohlen, das Widerrufszertifikats auf einem **externen Speichermedium zu sichern**. Jeder, der im Besitz dieses Zertifikats ist, kann das Schlüsselpaar **unbrauchbar** machen!*

Umgang mit dem Schlüsselpaar

Wie geht man richtig mit dem erzeugten Schlüsselpaar um?

Öffentlicher Schlüssel

Wie der Name bereits gesagt, kann und muss dieser Schlüssel öffentlich herausgegeben werden. Kommunikationsteilnehmer verwenden diesen Schlüssel, um Nachrichten an einen selbst zu verschlüsseln. Ebenso verwendet man den öffentlichen Schlüssel des Gegenüber, um verschlüsselte Nachrichten an ihn/sie zu senden und die digitale Signatur zu überprüfen.

Privater / Geheimer Schlüssel

Der private / geheime Schlüssel darf unter keinen Umständen herausgegeben werden und muss stets an einem sicheren Ort aufbewahrt werden!
Mit ihm und der zugehörigen Passphrase kann man verschlüsselte Nachrichten an einen selbst entschlüsseln oder aber Nachrichten digital signieren.

Wie macht man den öffentlichen Schlüssel für andere zugänglich?

Vergleichbar mit Telefonbüchern gibt es auch für öffentliche Schlüssel Verzeichnisse, sogenannte **Keyserver**. Dort kann man öffentliche Schlüssel bereitstellen und beziehen.

privatekeycheck.com – Ist mein privater Schlüssel sicher?

Überprüfung von öffentlichen Schlüsseln

Wie stellt man sicher, dass man den richtigen Schlüssel des Gegenüber (über einen Keyserver) bezogen hat?

Jeder Schlüssel besitzt ein eindeutiges Identifikationsmerkmal, den sogenannten **Fingerabdruck** (engl. **Fingerprint**):

```
> Schl.-Fingerabdruck =  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
                         YYYY YYYY
```

Durch eine zweifelsfrei echte Bestätigung dieses Fingerprints durch den Kommunikationspartner kann die Echtheit (Authentizität) des Schlüssels überprüft werden und dieser somit zum Verschlüsseln von Nachrichten verwendet werden.

Praxis: Zusenden von E-Mails unter den Workshop-Teilnehmern

Von: https://xkcd.com/1181/

Web of Trust (WOT)

🔧 Was ist das Web of Trust und weshalb existiert es?

Das Web of Trust ist ein Vertrauensnetzwerk, das durch das gegenseitige Zuweisen von Benutzervertrauen für öffentliche Schlüssel mehr Sicherheit schaffen soll.

🔧 Problematik

Ein Benutzer könnte einen öffentlichen Schlüssel mit falscher User-ID in Umlauf bringen. Wenn dann mit diesem Schlüssel Nachrichten verschlüsselt werden, kann der Eindringling die Nachrichten entschlüsseln und lesen – der beabsichtigte Empfänger erhält die Nachricht aber nie. Wenn er sie dann noch mit einem echten öffentlichen Schlüssel verschlüsselt an den eigentlichen Empfänger weiterleitet, fällt dieser Angriff nicht einmal auf (MitM-Angriff).

Wie wirkt das Web of Trust dieser Problematik entgegen?

Durch das gegenseitige Unterschreiben von öffentlichen Schlüsseln bestätigt ein Nutzer, dass er jeweils die Identität seines Gegenüber überprüft hat. Somit wird bestätigt, dass ein Schlüssel zu der in der User-ID angegebenen Person gehört.

Vertrauensstufen des Web of Trust *

Unbekannt	*Noch keine Angabe gemacht. Unter Umständen fragt GnuPG bei Bedarf nach dem Vertrauen in diesen Kontakt, wenn dessen Signatur benötigt wird.*
Kein Vertrauen	*Signaturen von diesem Kontakt werden niemals als gültig anerkannt.*
Marginales Vertrauen	*Diese Stufe sollte man den meisten Kontakten verpassen, denen man ein grundlegendes Verständnis des Web of Trust zutraut und von denen man nicht annimmt, dass sie betrügen.*
Volles Vertrauen	*Diese Stufe sollte man nur Leuten verleihen, die man wirklich gut kennt, auf menschlicher Ebene voll vertraut, und die auch nicht zu einem laxen Umgang bei Sicherheitsprozeduren neigen.*
Ultimatives Vertrauen	*Diese Stufe benutzt man üblicherweise nur für sich selbst.*

* nach https://wiki.ubuntuusers.de/GnuPG/Web_of_Trust/

🔧 Benutzervertrauen festlegen

```
> gpg --edit-key <Key-ID oder Name>
> gpg (GnuPG) x.x.x; Copyright (C) Free Software Foundation, Inc.
> This is free software: you are free to change and redistribute it.
> This is NO WARRANTY, to the extent permitted by law.
>
> pub    4096R/AAAAAAAA  erzeugt: 2016-06-11    verfällt: 2019-06-11
  Aufruf: SCA  Vertrauen: unbekannt   Gültigkeit: vollständig
> sub    4096R/BBBBBBBB  erzeugt: 2016-06-11    verfällt: 2019-06-11
  Aufruf: E
> [vollständig (1). <Vorname> <Nachname> (<Kommentar>)
  <local-part@domain-part.tld>
>
> gpg> trust
> …
```

Web of Trust (WOT)

```
 > Bitte entscheiden Sie, inwieweit Sie es diesem Benutzer zutrauen,
   Schlüssel anderer Benutzer korrekt zu überprüfen (durch Vergleich
   mit Personalausweisen, Vergleich der Fingerabdrücke aus
   unterschiedlichen Quellen usw.)
>
>     1 = Weiß nicht so recht oder will es nicht sagen
>     2 = Nein, ihm vertraue ich NICHT
>     3 = Ich vertraue ihm ein wenig
>     4 = Ich vertraue ihm vollständig
>     5 = Ich habe absolutes Vertrauen
>     m = zurück zum Hauptmenü
>
> Ihre Auswahl? <X>
```

Keyserver

Was sind Keyserver und wozu dienen sie?

Ein Keyserver (dt. „*Schlüsselserver*“) bietet Zugang zu öffentlichen Schlüsseln, die in asymmetrischen Kryptographiesystemen dazu benutzt werden, einer Person verschlüsselte Nachrichten zu senden oder ihre Signaturen zu verifizieren. Schlüssel können sowohl hochgeladen, als auch widerrufen und aktualisiert werden. Auf einem Keyserver werden zudem die gegenseitigen Benutzervertrauensstufen angezeigt.

Welche Keyserver gibt es?

Die drei bekanntesten Keyserver sind:

SKS Keyserver Verbund → https://sks-keyservers.net/i/
MIT PGP Public Key Server → https://pgp.mit.edu/
PGP Global Directory → https://keyserver.pgp.com/

Synchronizing Key Server (SKS)
=
Weltweiter, synchronisierter Verbund von Keyservern

Keyserver

Gibt es Probleme bei der Nutzung solcher Keyserver aus Sicht des Datenschutzes?

Durch das gegenseitige Bestätigen der Schlüssel besteht die Möglichkeit, nutzerspezifische Beziehungsdiagramme („Personennetzwerke“) anzulegen, die Auskunft über die eigenen Bekanntschaften geben. Aufgrund der Verbindlichkeit bei der Bestätigung ist ein Abstreiten (fast) nicht möglich!

Wie kann ich mich vor einem solchen „Tracking“ schützen?

Ein umfangreicher und ausreichender Schutz ist kaum möglich. Denkbare Lösungsansätze sind, Schlüssel im Rahmen von groß angelegten Keysigning-Parties zu bestätigen oder Schlüssel mit zufälligen Begegnungen gegenseitig zu bestätigen.

Praxis: Schlüssel auf bzw. vom Keyserver laden

Wie kann ich meinen öffentlichen Schlüssel auf einen Keyserver laden?

```
> gpg --keyserver <keyserver.company.tld> --send-key
  <local-part@domain-part.tld>
> success sending to <keyserver.company.tld> (status=200)
```

Wie kann ich einen öffentlichen Schlüssel eines anderen von einem Keyserver laden?

```
> gpg --keyserver <keyserver.company.tld> --recv-key 0xAAAAAAAA
> requesting key AAAAAAAA from <keyserver.company.tld> …
> key AAAAAAAA: 1 new signature
```

Feedback

Hat es Spaß gemacht? - Fragen / Kritik / Positives?

Vielen Dank!

Vielen Dank für das Interesse und die Teilnahme!

Im Anschluss:
Keysigning-Party bis ca. 17 Uhr in Raum V3

Quellen

https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html
https://de.wikipedia.org/wiki/OpenPGP
https://de.wikipedia.org/wiki/Informationssicherheit
https://de.wikipedia.org/wiki/Elektronische_Signatur
https://de.wikipedia.org/wiki/National_Security_Agency
https://de.wikipedia.org/wiki/Government_Communications_Headquarters
https://de.wikipedia.org/wiki/Vorratsdatenspeicherung
https://de.wikipedia.org/wiki/Tempora
https://de.wikipedia.org/wiki/PRISM
https://de.wikipedia.org/wiki/Verschl%C3%BCsselung
https://de.wikipedia.org/wiki/Man-in-the-Middle-Angriff
https://de.wikipedia.org/wiki/Schl%C3%BCsselserver
https://www.gnupg.org/
https://wiki.ubuntuusers.de/GnuPG/Web_of_Trust/
https://de.wikipedia.org/wiki/Warnzeichen