Kryptografie

Daten

Dateien

GnuPG = GNU Privacy Guard
PGP = Pretty Good Privacy

- Projekt GnuPG
- GnuPG Befehle
- GnuPG bei Ubuntuusers
- GnuPG-Software (auch für Windows-Systeme)

Installation

```
sudo apt-get install gnupg2
```

Verschlüsseln mit Passwort

Das Passwort bei u.a. Befehlen wird jeweils abgefragt.

Einzelne Datei mit Passwort verschlüsseln

```
gpg -c <Original_DATEINAME>
```

Neuer Dateiname automatisch = <Original_DATEINAME> + „.gpg“

Einzelne mit Passwort verschlüsselte Datei entschlüsseln

```
gpg -d <Krypt_DATEINAME>.gpg > <DATEINAME>
```

Verwaltung von Schlüsseln

Für das verschlüsseln wird ein Schlüsselpaar benötigt. Mit dem öffentlichen Schlüssel wird verschlüsselt, den kann jeder haben. So ist es kein Problem diesen auf der eigenen Webseite zu veröffentlichen. Mit dem privaten Schlüssel wird entschlüsselt, der muss geheim bleiben. Verschlüsselt ein Dritter mit dem öffentlichen Schlüssel, kann auch dieser die Datei nicht mehr ohne den privaten Schlüssel entschlüsseln. da er i.d.R. das Original besitzt, ist das auch nicht notwendig.

Der private Schlüssel muss unbedingt vor fremdem Zugriff geschützt bleiben!
Das gilt auch für den wie u.a. exportierten Schlüssel für die E-Mail-Nutzung.
Nach Aussen gegeben wird NUR der öffentliche Schlüssel.

Vorhandene **private Schlüssel** auflisten

```
gpg2 --list-secret-keys
```

```
gpg2 -K
```

Vorhandene **öffentliche Schlüssel** auflisten

```
gpg2 --list-keys
```

```
gpg2 -k
```

Schlüsselpaar erzeugen

```
gpg2 --full-gen-key
```

Der Schlüssel befindet sich dann im Home-Verzeichnis des ausführenden Users unter ~/.gnupg

Privaten Schlüssel exportieren (um ihn z.B. im E-Mailprogramm zu verwenden)

```
gpg --export-secret-keys --armor > my-secret-key.asc
```

Dateiname **my-secret-key.asc** nach Wunsch benennen (Erweiterung .asc sollte bleiben). Diese durch das System neu erstellte Datei befindet sich anschließend in dem Verzeichnis, in dem der Befehl ausgeführt wurde.

Öffentlichen Schlüssel exportieren

```
gpg -a --output gpg-key --export <Schlüssel-ID oder Name>
```

```
gpg -a --export <Schlüssel-ID oder Name> | tee gpg-key
```

Verschlüsseln mit Schlüsseln

Datei mit Schlüssel verschlüsseln

```
gpg2 --encrypt -a --recipient <Name oder Key_Id> test.txt
```

Datei mit Schlüssel entschlüsseln

```
gpg2 --decrypt --output entschluesselt.txt test.txt.asc
```

Fingerprint und Signatur

siehe Wikipedia/Elektronische_Signatur
siehe Wikipedia/Fingerprint

Fingerprint anzeigen

```
gpg2 --fingerprint <ID oder Name des Schlüssels>
```

Datei Signieren

```
gpg2 --detach-sig -a test.txt
```

Datei Signatur überprüfen

```
gpg2 --verify test.txt.asc
```

E-Mails

E-Mails digital signieren und verschlüsseln

In diversen E-Mail-Programmen ist OpenPGP bereits enthalten und neue Schlüssel können direkt damit erstellt werden.
Der öffentliche Schlüssel wird an den Kommunikationspartner geschickt, der die von ihm verfassten E-Mails damit verschlüsselt. (Nur) der Besitzer des privaten Schlüssels kann die E-Mail dann wieder entschlüsseln.
Lesen klappt nur mit einem System, auf dem der Schlüssel hinterlegt ist. Ist der private Schlüssel nur auf dem Notebook hinterlegt, kann die selbe E-Mail z.B. auf dem Handy nicht entschlüsselt werden. Der private Schlüssel muss dann auch dort hinterlegt werden.

Thunderbird

Schlüsselpaar erzeugen:
Extras → OpenPGP-Schlüssel verwalten → erzeugen → neues Schlüsselpaar.

Beim Einrichten von OpenPGP muss bei Thunderbird die Anwendung des privaten Schlüssels in den Konten-Einstellungen unter Ende-zu-Ende-Verschlüsselung aktiviert werden.

E-Mail Schlüssel veröffentlichen

Der öffentliche Schlüssel kann auf der Webseite von openpgp.org veröffentlicht werden. Dann hat jeder Zugriff auf den öffentlichen Schlüssel und kann danmit E-Mails an den dort hinterlegten Empfänger verschlüsseln. Der Schlüssel kann direkt über Thunderbird (Konto-Einstellungen → Ende-zu-Ende-Verschlüsselung→Veröffentlichen (Buttom im Feld des Schlüssels) veröffentlicht oder auf der Seite von openpgp.org hochgeladen werden. Die abschließende Veröffentlichung erfolgt nach Bestätigung der eingehenden E-Mail.

E-Mail Anhang verschlüsseln

Wenn die Entschlüsselung nach dem Verfahren mit Schlüssel für den Empfänger nicht geeignet/möglich ist, kann stattdessen ein E-Mail-Anhang verschlüsselt werden, bevor er der E-Mail angehängt wird. Siehe einzelne Datei mit Passwort verschlüsseln. Der Empfänger benötigt dann

natürlich das Passwort und das sollte ihm im günstigsten Fall über einen 2. Kanal (z.B. per Telefon) zur Verfügung gestellt werden. In keinem Fall darf es in derselben E-Mail enthalten sein!
Abhängig von der notwendigen Sicherheitsstufe sollte auch die Komplexität bzw. die Länge des Passwortes gewählt werden. Einfache Passwörter sind ungeeignet, letztlich nur für unwichtige Dokumente richtig und dann kann man auch Unverschlüsselt senden.

Partitionen

Handelt es sich um die Systempartition, muss dies bereits während der Installation des OS erfolgen.
Achtung: Muss ein Server mit verschlüsseltem System neu gestartet werden, dann kann man dies nicht aus der Ferne durchführen. Das Passwort muss lokal, vor Ort eingegeben werden!!!
Siehe auch Installation Debian, Partitionierung.
Option: System und Daten trennen und nur Datenbereiche verschlüsseln, dann lässt sich das auch aus der Ferne realiseren. Hierzu kann z.B. Veracrypt genutzt werden.

Partitionen anzeigen

```
lsblk
```

Speicherplätze anzeigen

```
df -h
```

https://wiki.ubuntuusers.de/LUKS/

Mit cryptsetup (LVM-LUKS)
/dev/nvme0n1p3 = verschlüsselte Partition.
Anzeige der Header-Information:

```
sudo cryptsetup luksDump /dev/nvme0n1p3
```

Anzeige Status:

```
sudo cryptsetup status /dev/nvme0n1p3_crypt
```

Passwort: Bis zu 8 Passwörter/Slots möglich (Keyslot 0 bis 7) / Vor dem Löschen immer erst neues anlegen
Neues Passwort/Slot anlegen

```
sudo cryptsetup luksAddKey /dev/nvme0n1p3
```

Passwort/Slot löschen <SLOT> = 0 bis 7

```
sudo cryptsetup luksKillSlot /dev/nvme0n1p3 <SLOT>
```

Passwort ändern

```
sudo cryptsetup luksChangeKey /dev/nvme0n1p3
```

Container & Laufwerke

LUKS

https://www.tuxedocomputers.com/de/Infos/Hilfe-Support/Anleitungen/LUKS-Verschluesselungskennwort-aendern.tuxedo

Veracrypt

Hier Installation und Befehle über Konsole.
Beispielhaft für ein verschlüsseltes und am Server angeschlossenes USB-Speichermedium.

https://veracrypt.fr/en/
https://github.com/arcanecode/VeraCrypt-CommandLine-Examples/blob/main/Linux/post.md

Download Programm

```
wget
https://launchpad.net/veracrypt/trunk/1.25.9/+download/veracrypt-console-1.25.9-Debian-11-amd64.deb
```

Versionsnummer entsprechend der aktuellen bzw. verwendeten Version anpassen.
Installation

```
sudo apt install ./veracrypt-console-1.25.9-Debian-11-amd64.deb
```

Anzeige der Speichermedien

```
lsblk
```

```
df -h
```

Einhängen

```
veracrypt --text --mount /dev/<SDB> /<ZIELPFAD>/ --password <PASSWORD> --pim 0 --keyfiles "" --protect-hidden no --slot 1 --verbose
```

Anpassen:
<SDB> Ort der Festplatte → Abfrage über „lsblk“
<ZIELPFAD> (existierendes) Verzeichnis
<PASSWORD>

Aushängen

```
sudo veracrypt --text --dismount --slot 1
```

Anzeige gemounteter Volumen

```
sudo veracrypt --text --list
```

Veracrypt Auto-Mount

Soll eine Partition oder eine Container-Datei automatisch nach dem Einloggen eingebunden werden, dann muss hierfür auch das Passwort auf dem PC hinterlegt werden. Wenn dies in einem nicht geschützten Bereich erfolgt, dann ist das grundsätzlich problematisch, da auch ein Angreifer, der im Besitz des Gerätes ist, das Passwort auf der Festplatte finden kann. Hier würde sich Variante 1 anbieten.
Ist die Systemplatte geschützt, Passworteingabe bereits beim Booten des Rechners, kann das Passwort für weitere Festplatten in diesem geschützten Bereich abgelegt werden. Siehe Variante 2.

Variante 1

https://www.computercorrect.com/2018/operating-systems/linux/ubuntu/auto-mounting-a-veracrypt-volume-under-ubuntu-debian-linux/
https://www.freedesktop.org/software/systemd/man/latest/crypttab.html
Block (Volumen) beim Booten anlegen

```
sudo nano /etc/crypttab
```

```
volume_name /dev/sda password tcrypt-veracrypt,tcrypt-
keyfile=/path/to/keyfile
```

- volume_name kann frei gewählt werden und wird als bekanntes Volumen im Pfad /dev/mapper/ hinzugefügt
- /dev/sda = Pfad des Volumens oder des Containiers, das verschlüsselt ist
- password
  - '-' = Abfrage des Passwortes beim booten. Die beiden Anführungszeichen müssen auch gesetzt werden. Versuche, das Passwort direkt hier einzutragen sind gescheitert.
  - Alternative: Pfad zu einer Datei, die ausschließlich das Passwort enthält.
  - Alternative: Wird ein keyfile genutzt, dann wird hier eingetragen /dev/null
- Wird eine keyfile benutzt, dann wird der Pfad hier eingetragen. Wenn keiner verwendet wird, dann ist der Text „,tcrypt-keyfile=/path/to/keyfile“ inkl. Komma zu entfernen.

Ansicht der bestehenden Blocks

```
lsblk
```

Block/Volumen beim Booten einbinden (mount)

```
sudo nano /etc/fstab
```

```
/dev/mapper/volume_name /mnt/point auto nosuid,nodev,nofail 0 0
```

- volume_name, wie oben ausgewählt (Pfad /dev/mapper gehört trotzdem dazu)
- /mnt/pont = wie gewünscht

Variante 2

```
sudo nano /secureplace/decrypt.sh
```

```
#!/bin/bash
sudo veracrypt -t /dev/sda /mnt/xyz --non-interactive -p <PASSWORD>
```

<PASSWORD> ersetzen mit dem echten Passwort

Datei /etc/fuse.conf editieren:

```
sudo nano /etc/fuse.conf
```

```
mount_max = 1000        | das # davor entfernen
user_allow_other        | das # davor entfernen
```

<USER> (Name ändern) zur Gruppe users hinzufügen

```
sudo usermod -a -G users <USER>
```

Datei /etc/sudoers editieren (am Ende einfügen):

```
sudo visudo
```

```
%users  ALL=(root) NOPASSWD: /usr/bin/veracrypt
```

Automatisch beim Starten über /etc/crontab ausführen

```
sudo nano /etc/crontab
```

```
@reboot <USER>      /secureplace/decrypt.sh
```

am Ende einfügen, Pfad s.o., und <USER> anpassen

Datenübertragung

Siehe auch Netzwerktraffic_analysieren

SSH

Secure Shell oder SSH bezeichnet ein kryptographisches Netzwerkprotokoll für den sicheren Betrieb von Netzwerkdiensten über ungesicherte Netzwerke.
Siehe SSH-Verbindungen.

VPN

Virtual Private Network (deutsch „virtuelles privates Netzwerk“; kurz: VPN) bezeichnet eine Netzwerkverbindung, die von Unbeteiligten nicht einsehbar ist.
Siehe WireGuard - VPN-Server

TLS/SSL

Transport Layer Security (TLS, englisch für „Transportschichtsicherheit“), auch bekannt unter der Vorgängerbezeichnung Secure Sockets Layer (SSL), ist ein Verschlüsselungsprotokoll zur sicheren Datenübertragung im Internet.
Siehe SSL-Verbindung mit Apache Web-Server

SFTP/FTP

SFTP: SSH File Transfer Protocol bzw. Secure File Transfer Protocol, ein Netzwerkprotokoll
FTP: File Transfer Protocol ohne Verschlüsselung & Signatur.
Siehe FTP-Zugang

E-Mails

Siehe oben unter E-Mails