SSH-Keys erzeugen und verwalten

SSH-Key erzeugen

> Shortcut: `ssh-keygen -t ed25519 -a 100 -C 'email@mydomain.de'`

Es wird ein Passphrase abgefragt. Soll das Keyfile unverschlüsselt gespeichert werden, kann die Eingabe eines Passworts mit Enter übersprungen werden. Dies ist allerdings nicht empfehlenswert, denn falls der Key in die falschen Hände gelangt, wären alle Server mit dem Pubkey in den authorized_keys ohne weitere Passwortabfrage offen.

```
ssh-keygen -t [ (dsa) | (ecdsa) | ed25519 | rsa | (rsa1) ] -b 4096 -a 100 -C '<kommentar>'


# als copy&paste Beispiel mit ed25519 (hat eine fixe Bitgröße von 256 bit,
# sie muss deshalb nicht angegeben werden):
ssh-keygen -t ed25519 -a 100 -C 'email@mydomain.de'
```

Dies erzeugt zwei Dateien in `~/.ssh/id_<key_type>*` je nach verwendetem Verfahren. Beispiel RSA: in der Datei `id_rsa` steht der (verschlüsselte) private Key. In der Datei `id_rsa.pub` steht der öffentliche Teil des Schlüssels, der verteilt werden darf/muss.

Der Typ ed25519 ist zum heutigen Stand (05.03.2021) die beste Wahl. Allerdings wird er noch nicht überall unterstützt. In diesem Fall empfiehlt es sich (zusätzlich) einen RSA-Key mit 4096 bit zu erstellen.

ecdsa sollte nicht verwendet werden, da möglicherweise eine Hintertür für die US-Regierung eingebaut ist. RSA1 und DSA sollten auch nicht mehr verwendet werden.

Unter Windows gibts mit `puttygen.exe` ein grafisches Pendant dazu. Hier lassen sich die erzeugten Keys auch ins Unix-Format konvertieren.

Passphrase des SSH-Keys nachträglich ändern

```
ssh-keygen -p -f ~/.ssh/id_rsa
```

SSH-Key in den SSH-Agenten laden

Der SSH-Agent muss gestartet sein, bei Fedora Core passiert dies automatisch beim Starten der X-Oberfläche.

```
ssh-add
```

Dies lädt defaultmäßig alle id_*-Files in ~/.ssh in den Agent. Der Agent dient der Bequemlichkeit, d.h. man lädt einmal seinen Key und von nun an kümmert sich der Agent um alle anfragenden Programme, wie z.B. ssh oder scp. Für Windows gibts im Putty-Paket ein Programm namens `pagent.exe`, welches den gleichen Zweck erfüllt.

Hier ist eine Anleitung, wie man seinen ssh-Key für die Anmeldung an KDE/Gnome verwenden und ihn gleich in den Agent laden kann.

SSH-Pubkey verteilen

Um nun den SSH-Key für die Authentifizierung an anderen Servern verwenden zu können muss er noch dort abgelegt werden. Dazu bringt openssh ein schönes Tool mit:

```
ssh-copy-id [user@]machine
```

Dies erzeugt auf dem Zielserver ggf. das Verzeichnis ~/.ssh, legt dort den öffentlichen Teil des Schlüssels in der Datei ~/.ssh/authorized_keys ab und vergibt gleich passende Berechtigungen.

Fingerprint des SSH-Keys anzeigen

Mit dem Fingerprint läßt sich der Key schnell von anderen verifizieren (könnte z.B. telefonisch abgeglichen werden).

```
# Fingerprint als SHA256 ausgeben (Default bei Debian 8.8)
ssh-keygen -l -f ~/.ssh/id_rsa.pub


# Fingerprint als MD5 ausgeben
ssh-keygen -l -E md5 -f ~/.ssh/id_rsa.pub
```

Revision #5
Created 2021-03-05 10:10:58 UTC by magenbrot
Updated 2021-07-17 14:45:38 UTC by magenbrot