



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386



Tools 2026 – Remote Working

Shigehiko Schamoni
28.05.2026

With material from F. Beenen (SuS)

Overview

1 Introduction

- Overview
- Remote Infrastructures

2 Virtual Private Networks (VPNs)

- Configuration types
- VPN Providers
- University VPN
- Multi-Factor Authentication
- Summary

3 Secure Shell

- Introduction
- Public Key Authentication
- Remote Shell
- Port Redirection and SOCKS Proxy
- File Transfers
- X11 Forwarding

4 Accessing Remote Desktops

- Introduction
- X2Go
- Microsoft Remote Desktop

5 Useful Tools

Objectives

Today we take a closer look on:

- How to work with remote infrastructures
 - Remote connection techniques
 - Virtual Private Networks (VPN)
 - Secure Shell (SSH)
 - Applications
- How to use URZ/university services
 - heiBOX
 - GigaMove
 - Terminal Server
- Useful tools for everyday work
 - Sharing files
 - Managing passwords

Connecting to a Remote Infrastructure

- Scenario: access remote services from...
 - University
 - Work place
 - Home
- Public services vs. restricted resources:
 - Websites, email, etc.
 - Fileserver, sensitive web pages, access to licensees, etc.
- Restrictions typically enforced by
 - Firewalls: filter traffic based on origin/destination/protocol
 - Applications: client authentication, restricted address origin
- Enable access by moving our origin address to a trusted zone
 - Overlay networks (VPNs)
 - Proxies / Jumphosts: connect through a trusted intermediate host (SSH)

Virtual Private Networks

- Idea: Use public network as transport network for a “virtual” overlay network.
- Creates a secure “tunnel” from a client’s machine to another network via an unsecure network (e.g. Internet)
- Connection usually is protected by
 - Authentication (certificates/username+password)
 - Encryption
- VPNs have nothing to do with illegal activity
 - Can be used to circumvent geo-blocking
 - Can be used to bypass government restrictions

Configuration Types (I)

Split Tunnel:

- Only certain addresses or networks are redirected through the tunnel (e. g. the university's network).
- Other traffic bypasses the VPN.
- Common scheme to access restricted resources without affecting other traffic.
- Name resolution (DNS) may or may not be redirected.

Configuration Types (II)

Closed Tunnel (“Full redirection”):

- All traffic is redirected through the VPN.
- Target IP of the VPN gateway is (obviously) excluded from redirection.
- Benefit from encryption of VPN: Additional security only if local network is untrusted (e.g. unencrypted airport/hotel WiFi) *and* VPN host is trusted!
- VPN gateway can see all traffic that passes through it!

Useful VPNs

- Use VPN to connect to the University network
- Access restricted resources:
 - HEIDI (University library)
 - KIP-Terminalserver (Linux)
 - URZ-Terminalserver (Windows)
 - ZITI-Applicationserver
 - ZITI-Fileserver (zitipool-home)
- URZ Heidelberg uses Cisco AnyConnect VPN
- Other VPN-solutions exist (e.g. OpenVPN)

URZ VPN Installation

- Installation guide here:
<https://urz.uni-heidelberg.de/de/vpn>
- Download VPN Client here:
<https://vpn-ac.urz.uni-heidelberg.de/>
- Setup Multi-factor-authentication
 - Visit <https://mfa.uni-heidelberg.de/>
 - Generate and register a token
 - Only accessible from University's network (Uni Heidelberg eduroam)
- Connect to server `vpn-ac.uni-heideberg.de`
- Log in with Uni-ID and password.
- VPN is set up as *fully redirected tunnel* by default!

Connecting to URZ VPN (I)

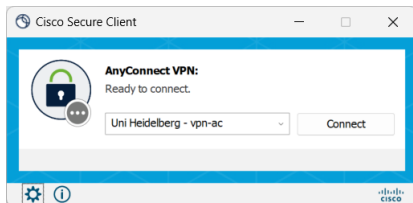


Figure: Use Standard VPN Gateway

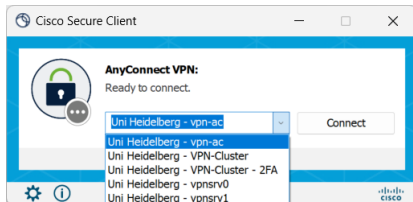
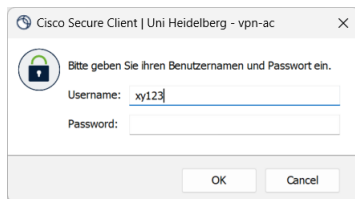


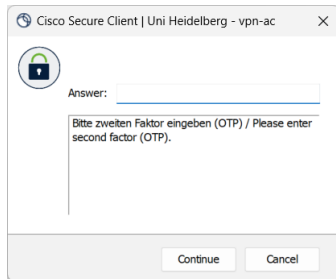
Figure: Select VPN Gateway

Connecting to URZ VPN (II)



A screenshot of the Cisco Secure Client login window titled "Cisco Secure Client | Uni Heidelberg - vpn-ac". The window contains a padlock icon and the text "Bitte geben Sie ihren Benutzernamen und Passwort ein." (Please enter your username and password). Below this, there are two input fields: "Username:" with the text "xy123" entered, and "Password:" which is empty. At the bottom right, there are "OK" and "Cancel" buttons.

Figure: Enter UniID and password

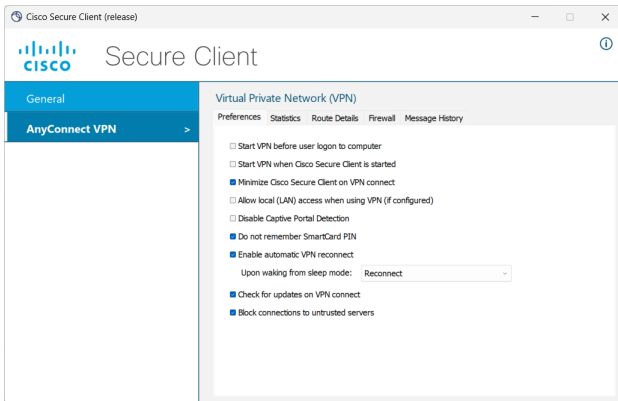


A screenshot of the Cisco Secure Client second factor window titled "Cisco Secure Client | Uni Heidelberg - vpn-ac". The window contains a padlock icon and the text "Answer:" followed by an empty input field. Below this, there is a text box with the text "Bitte zweiten Faktor eingeben (OTP) / Please enter second factor (OTP)." (Please enter second factor (OTP)). At the bottom right, there are "Continue" and "Cancel" buttons.

Figure: Enter Second Factor

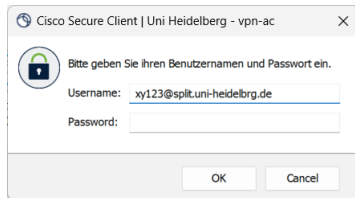
Enable LAN Connections

- LAN connections are usually blocked as well
 - No access to network printers
 - Router
 - ...
- Can allow LAN connections (only in the same subnet) with configuration option:



Split Tunnel with URZ VPN

- Split-Tunnel only redirects university network.
 - 129.206.0.0/16
 - 147.142.0.0/16
- Uses BelWü DNS (but does not enforce it).
- No access to IEEE and other subscriptions.
- Setup via changed username:
`UniID@split.uni-heidelberg.de`



URZ VPN with Linux

- If you use Linux, you do not have to install suspicious software from Cisco!
- On Debian systems, install `OpenConnect` via:

```
$ sudo apt install openconnect
```
- OpenConnect supports a large number of VPN protocols, among them Cisco AnyConnect.
- Command line example for opening a split tunnel:

```
user@comp:~$ sudo /usr/sbin/openconnect --protocol=anyconnect \  
--useragent='AnyConnect' \  
--no-external-auth \  
--user=UniID@split.uni-heidelberg.de \  
vpn-ac.uni-heidelberg.de
```

- Remote system then asks for password and second factor as usual.

Multi-Factor Authentication (MFA)

- Obligatory for URZ VPN since Dec 2026
- System uses *time-based one-time passwords* (TOTP)
- Token needs to be generated within the University's network
- Multiple tokens can be registered
 - In some cases, the soft token can be selected (MFA Token Self Service)
 - In some cases, the topmost token is used (Cisco AnyConnect VPN)
- Recommendations:
 - Register one main and one backup token
 - Use KeePassXC (or similar) to generate your main token

Using MFA on Multiple Devices

- When using KeePassXC (or similar) as the main token, multiple devices can use the same token:
 - Share KeePassXC database via heiBOX
 - Register multiple devices using the secret or the QR-code within KeePassXC
- Recommendations:
 - Register one main and one backup token
 - Use KeePassXC to generate your main token
- KeePassXC:
 - Available for Windows, Linux, MacOS
 - <https://keepassxc.org/>

MFA on KeePassXC (I)

- Copy the *secret* from the MFA Token Self Service page:

Add soft token (time)

✓ Install app — ✓ Create token — 3 Import token into app — 4 Done



A token with the serial number TOTP610030EF has been created. Please add the token by scanning the QR code in the Authenticator app.

Enter your first OTP (one-time password) to verify the correct installation:

✓ VERIFY

The OTP field cannot be empty

 HIDE TOKEN DETAILS

Configuration URL

otppath://totp/Universit%C3%A4t%20Heidelberg: ?secret=**GVUCH5E0C32U6MPT64WGUVKDBXYLUAW&ist**



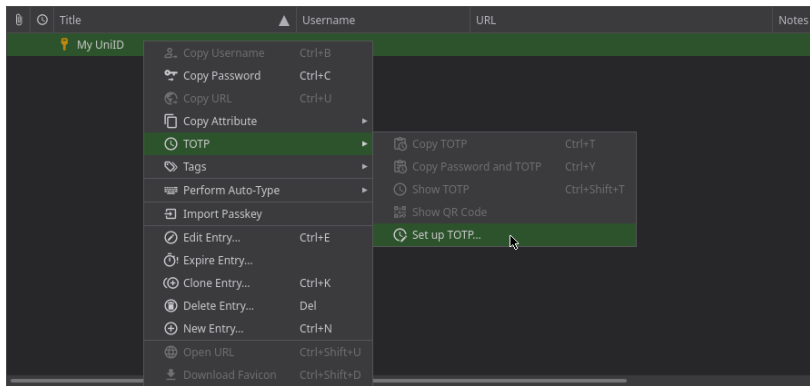
Seed

seed://356823f48e16f54f31f3f72c6a55430df185d016



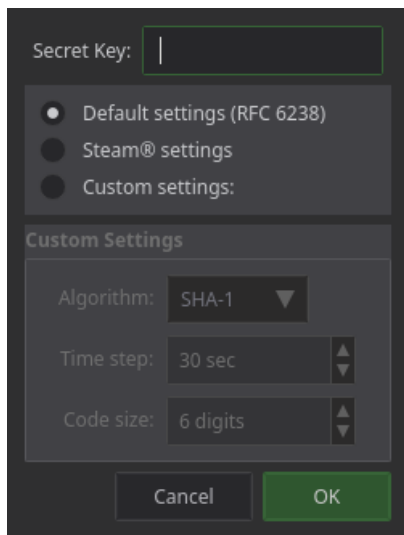
MFA on KeePassXC (II)

- Set up TOTP for your UniID-entry in KeePassXC



MFA on KeePassXC (III)

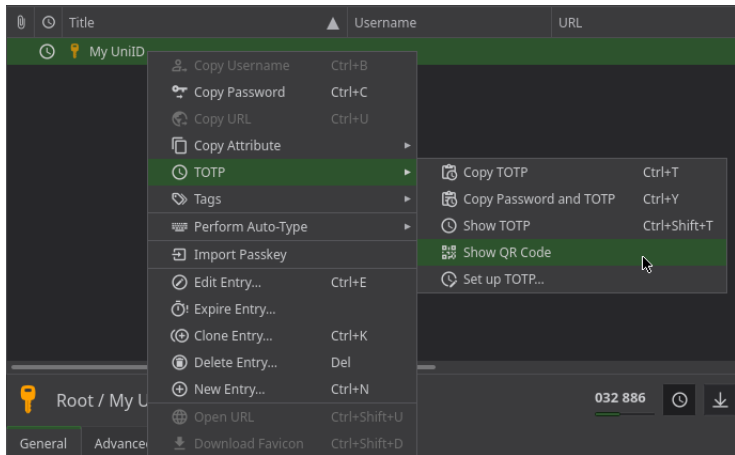
- Paste the secret into KeePassXC's TOTP setup window:



The screenshot shows the TOTP setup window in KeePassXC. At the top, there is a 'Secret Key' label followed by a text input field containing a single vertical bar '|'. Below this, there are three radio button options: 'Default settings (RFC 6238)' (which is selected), 'Steam® settings', and 'Custom settings:'. The 'Custom settings' section is expanded, showing 'Algorithm' set to 'SHA-1' with a dropdown arrow, 'Time step' set to '30 sec' with up/down arrows, and 'Code size' set to '6 digits' with up/down arrows. At the bottom, there are 'Cancel' and 'OK' buttons.

MFA on KeePassXC (IV)

- Transfer the token to a smartphone app, e.g. AEGIS Authenticator



Summary

VPNs are cool:

- Applications do not need to be changed: Destination addresses of remote service remain the same.
- VPNs can redirect your traffic and increase privacy in untrusted local networks.
- Site-to-site connections are easy to set up (e.g. link your house with your parents' network).
- They are easy to use: client applications manipulate the routing tables transparently.

VPNs are not so cool:

- they can redirect your traffic: VPN gateway can see things that you may not want it to see.
- VPN client does unwanted stuff: Redirect all traffic instead of only the required networks. May filter or inspect DNS.
- From admin's point of view: Often need very sophisticated filtering rules to prevent issues originating from VPN users in your network.

Secure Shell Overview

- SSH is commonly known for “getting a shell on a remote computer”.
- Can do much more than just providing a command prompt:
 - Providing a remote command prompt (obviously...).
 - Transferring files (with sftp subsystem).
 - Displaying remote windows (with local X-Server).
 - Relaying/forwarding TCP ports.
 - Running SOCKS proxy.
- Traffic is encrypted and uses only one single port (unlike FTP).
- SSH is extremely useful for automation / writing scripts:

```
user@comp:~$ ssh UniID@zitigate.ziti.uni-heidelberg.de "ip a | grep global"  
inet 147.142.40.245/22 brd 147.142.43.255 scope global enp1s0
```

Secure Shell (SSH)

- SSH connections use port 22 by default.
 - May be blocked by many firewall configurations. URZ filters SSH from outside University network.
 - Often useful to run SSH servers on other ports (e.g. port 2222).
 - Insane amount of unauthorized login attempts when using default port.
- Different programs available:
 - Console access: PuTTY / `ssh` command
 - Data transfer: FileZilla / WinSCP / `scp` / `rsync`
- Authentication methods
 - Username/password (okay for occasional usage, bad for scripts)
 - Cryptographic key (enables automated logins)

Public Key Authentication

- Generate *pair* of cryptographic keys
 - Private Key: Belongs to you – *never* share this key with anyone. Do not upload to OneDrive, etc...
 - Public Key: Also belongs to you – this key is shared with anyone that wants to communicate with you using encryption.
 - Both keys are “inverse” to each other: Data encrypted with public key can only be decrypted with private key (and vice versa).
 - Use ED25519 key (elliptic curve crypto) or RSA with a large key size (at least 2048 bit).
- Put public key on SSH server in special directory.
- When authenticating with server your client tries to authenticate with the server using your keys.
- If the server knows your public key, it will let you in.

Linux Key Management

- Generate key with `ssh-keygen -t ed25519`.
- Can create key without passphrase.
- Keys will be located at `~/.ssh/id_ed25519` and `~/.ssh/id_ed25519.pub`.
- Get the public key (`cat ~/.ssh/id_ed25519.pub`) – we need it for the next step.

Connect to SSH servers:

- Use the `ssh` command!
- Connect to testserver via:

```
$ ssh UniID@physik1.kip.uni-heidelberg.de
```

Adding and removing a passphrase:

- Use `ssh-keygen -p -f ~/.ssh/id_ed25519`
- Add password by entering a password
- Remove password by setting an empty password

Store the Public Key on the Server (I)

Now connect to

```
ssh UniID@physik1.kip.uni-heidelberg.de :
```

- 1 When connecting you get warnings:
 - The server key is unknown:
 - Your system does not know the server because you are connecting for the first time.
 - You can connect in this case.
 - Do not ignore this warning under normal circumstances!
 - The server key refused our key:
 - This is okay as the server does not know about our key yet.
- 2 Go into the `~/.ssh` directory – create it if it does not exist (`cd ~/.ssh` or `mkdir ~/.ssh`).
- 3 Open the `authorized_keys` file with an editor (e.g. `nano` or `vim`)
- 4 Paste the public key into the file.

Store the Public Key on the Server (II)

- Ensure that the permissions are correct:
 - `$ chmod 700 ~/.ssh`
 - `$ chmod 600 ~/.ssh/authorized_keys`
 - SSH servers do not like it if the permissions are too open.
- Terminate the session by typing `exit` or pressing Ctrl+D.
- Reconnect to `physik1.kip.uni-heidelberg.de`.
- This time the server should accept your key.

Getting a Remote Shell

- Most basic functionality of SSH: Get a command prompt of a remote server.
- Why to run things remotely:
 - Own computer is not powerful enough: More compute power at university to run compute-intensive tasks.
 - License agreements prohibit installation e.g. of Virtuoso on private machines.
 - Data access: work files are located on remote infrastructure due to size or privacy issues.
- A shell allows you to execute any command on the remote server.
- Possible to redirect graphical output of applications to your local machine.

Logging into KIP Servers

- The KIP hosts five different servers for working remotely:
 - `physik1.kip.uni-heidelberg.de`
 - ...
 - `physik5.kip.uni-heidelberg.de`
- A lot of useful software is installed on the system.
- Machines are older and thus not very powerful (AMD Opteron 8350 from 2007).
- You can run licensed software such as Mathematica.

Redirecting TCP Ports

- You can use SSH to redirect ports of remote hosts to you via the SSH gateway.
- SSH gateway acts as proxy: Remote host thinks that the SSH gateway connects.
- Very useful for accessing restricted resources!
- Method:
 - Local port is virtually bound to remote port.
 - Connect to remote resources by accessing `localhost` with the bound port number.
 - Connection is held as long as SSH connection is open.
- Firewalls only see an SSH connection and thus cannot block the tunneled traffic!

Configuring Local Forwarding in Linux

- Use case: Connect to a service on an internal network from the outside.

- The option `-L` in `OpenSSH` can be used for this:

```
$ ssh -L 8080:ziti-peac:443 UniID@zitigate.ziti.uni-heidelberg.de
```

- By default the bound local port is open to everyone!

- Specify a bind address to listen only to a specific address:

```
$ ssh -L localhost:8080:ziti-peac:443 UniID@zitigate.ziti.uni-heidelberg.de
```

- The option `-N` suppresses execution of a remote command.
Hit `Ctrl-C` to end the tunnel.

```
$ ssh -N -L 8080:ziti-peac:443 UniID@zitigate.ziti.uni-heidelberg.de
```

Using SOCKS Proxy over SSH

- Using a SOCKS Proxy is possible to send (almost) arbitrary data via a proxy server.
- *Proxy server* acts as a Man-in-the-middle between you and your desired destination.
- SSH supports “dynamic port forwarding”
 - Port redirections that we have previously defined manually can be set up automatically “on-the-fly”.
 - Applications that support SOCKS proxy servers can redirect their traffic through SSH connections.
- Why is this useful?
 - We can redirect traffic without using VPN!
 - We only need an SSH client and a proxy-aware application.
 - Does not modify routing tables $\Rightarrow$ does not need admin privileges!

SOCKS Proxy in Firefox

- Use the option `-D` in `OpenSSH` to create a SOCKS proxy:

```
$ ssh -D 6000 UniID@zitigate.ziti.uni-heidelberg.de
```

- Configure the browser to use the SOCKS proxy:
 - E.g. in Firefox, open *Settings* → *General* and scroll down to *Network Settings*.
 - Under *Network Settings* → *Settings*, select *Manual proxy configuration* and enter:
 - SOCKS host: localhost
 - SOCKS port: 6000
- Find out your current IP (type `ip a` into a terminal).
- Check which IP a webserver thinks you have (e.g. visit <https://www.wieistmeineip.de/>)

Copying Files over SSH

- Secure methods for moving files over an SSH connection:
 - `sftp` – secure File Transfer Protocol
 - `rsync` – very powerful tool to synchronize files and directories
 - `scp` – secure copy
- `sftp` is an optional subsystems of SSH.
- Functionality is similar to standard FTP (File Transfer Protocol) but addresses several issues of the original:
 - Uses encryption provided by SSH.
 - Only uses one port (from the SSH connection)
- Works with GUI programs like FileZilla or WinSCP for transferring files.

Rsync

- Transferring large directories via command line is often required.
- Command line tool `rsync` synchronizes directories very fast.
 - Only copies files that are different.
 - Can compress data during transfer.
 - Can copy over SSH connection.
 - Can optionally compare checksums of files.
 - Can create true mirrors of filesystems (even with hardlinks!)
- Needs the binary `rsync` installed.

Secure Copy

- If `rsync` is not available (for whatever reason), you can still copy files from one machine to another using `scp` (Secure Copy).
- Uses SSH connection to transfer data.
- Very basic feature set, enough to quickly copy files and directories (e.g. option `-r` for recursive copy)
- Requires the remote server to allow console access:
 - If only the SFTP subsystem is allowed (and no shell), `scp` will not work.
- Example:

```
$ scp somefile.pdf user@server:remoteDirectoryInHome/
```

Redirect GUI windows

- Sometimes you want to run applications that open a graphical interface on a remote server.
 - Mathematica on KIP servers (needs license)
 - Virtuoso for full-custom chip design (needs license and large design kit files)
 - Vivado for FPGA design (needs license and a lot of computing power)
- We can run a local X-Server on our private machine that displays remote windows (the environment is called “X Window System”).
- Remote window contents are redirected through SSH connection and displayed on local machine!
- Requirements:
 - Local X-Server, e.g. a Linux Desktop based on X11
 - Connection to SSH server that allows X11-Forwarding.

X11 Forwarding with Linux

- Allow X-Forwarding via `-X` flag with the `ssh` command:
`$ ssh -X UniID@physik1.kip.uni-heidelberg.de`
- We can now run a GUI application on the KIP server and have the output window on our local machine:
 - For example, run `gedit ~/.ssh/authorized_keys`.
 - Wait for a looong time...
 - Editor window opens on local machine!
- This works within the University's network, but is not usable via slower connections such as DSL.

Accessing Remote Desktops

- Reasons for using a remote desktop:
 - Sometimes just redirecting windows is not enough.
 - Accessing a full desktop is sometimes desirable.
 - Latency of SSH X-Forward is crucial: Cannot use SSH X-Forward over WAN connections properly (especially with DSL/cable modem).
- Optimized Remote Desktop clients work better:
 - X2Go: Optimized version of X-Forwarding that performs much better than the “direct” method.
 - Microsoft Remote Desktop: Powerful tool to connect to RDP-Servers.

X2Go

- Consists of server and client.
- Remote system must have X2Go server installed.
- Client is available for all usual operating systems.
- Enables to open virtual desktop in a local window.
- Also supports redirection of audio (for playing music on remote server).
- Traffic is tunneled via SSH (again).
- Download and install X2Go client from here:
<https://wiki.x2go.org/doku.php/download:start>.

X2Go Configuration Example

- Open the client and configure session:
 - Create a new session.
 - Enter the host `physik1.kip.uni-heidelberg.de`
 - Use your UniID as username.
 - Leave the port at 22 (default).
 - Select *XFCE* instead of KDE as the session type.
- Save and open the connection.
- Enter your password and accept the warning that the host key is unknown.
- It is possible to mount a local folder to the remote server for file sharing.

Microsoft Remote Desktop

- Microsoft Remotedesktop is a built-in tool to connect to Windows machines.
- A remote administration tool for graphic environments.
- Has several useful features:
 - View and control a (virtual) remote desktop (obviously).
 - Exchange files.
 - Redirect audio devices.
 - Pass-through of local printers: Print document from remote server on local printer.
 - Sharing of clip-board (Ctrl+C, Ctrl+V between local and remote system).
- Uses the *Remote Desktop Protocol* (RDP), Port TCP/3389.
 - Can be encrypted (but does not have to be).
 - Usage of RDP has many security implications because of interactions with the Windows host.
 - Usage is almost always firewalled $\Rightarrow$ Not available from the internet.

Remote Desktop Software

■ Remote Desktop Client:

- Pre-installed on every Windows client. Accessible via:
Windows + R → `mstsc`
- Open Source clients for Linux: `rdesktop`, `remmina`,
`xfreerdp3`.
- Windows App for Mac clients from the App Store.

■ Remote Desktop Server:

- Included in every Windows system except Windows Home.
- Needs to be activated manually and local firewall needs to allow RDP connections.
- For Linux: Use `xrdp`. Requires display manager and desktop environment (e.g. XFCE).

Remote Desktop in University

- University has remote desktop servers for different purposes:
 - `physik1.kip.uni-heidelberg.de`
...
 - `physik5.kip.uni-heidelberg.de` : Hosted by KIP, Linux machines with `xrdp` . Can run Mathematica.
 - `ts.ad.uni-heidelberg.de` : Windows terminal server hosted by URZ.
- You can use these servers and access the installed software from within the University network.

Sharing Files without Dropbox (I)

- Use case: you need to send someone a file that cannot/should not be attached to an email.
- You might use file sharing services like OneDrive, Dropbox or WeTransfer.
- Universities have their own sharing services that are better, less shady and more trustworthy:
 - heiBOX: <https://heibox.uni-heidelberg.de>
 - GigaMove: <https://gigamove.rwth-aachen.de>
- Data is stored within the University's network.

Sharing Files without Dropbox (II)

■ heiBOX

- Uses Seafile. Comparable to Dropbox.
- Every student has 10 GB free online storage.
- Unlimited data retention period $\Rightarrow$ files are not deleted!
- Can create download and upload links for files and directories: public links or only for selected users.
- Can invite other people (without Uni-ID) for working together on shared files.
- Supports WebDAV access $\Rightarrow$ use scripting for automated up- and downloads.

■ GigaMove

- Comparable to WeTransfer.
- Is hosted by RWTH Aachen and accessible for people with DFN accounts (i.e. UniID).
- Everyone has 1 TB storage, files get deleted automatically after 7 to 14 days.
- Easy alternative for sending large E-Mail attachments.

Password Manager KeePassXC

- <https://keepassxc.org/>
- Manage multiple databases containing sensitive login data
- Generate secure passwords
- Generate time-based one-time passwords (TOTPs)
- Browser integration: automatically fill in login forms
- Available for Windows, Linux, MacOS
- Very useful in combination with some cloud storage like heiBOX.

Exercise

- VPN/SSH/RDP Exercise