

How to create an SSH key pair

The SSH key pair is used to log in to LeoMed via the command line.

- the ed25519 algorithm for generating the SSH key pair is mandatory
- the passphrase (non-empty, of minimum 16 characters) protection of the SSH key pair is mandatory
- format convention for sharing the public SSH key: as file named <username>.pub

Linux / Unix (macOS)

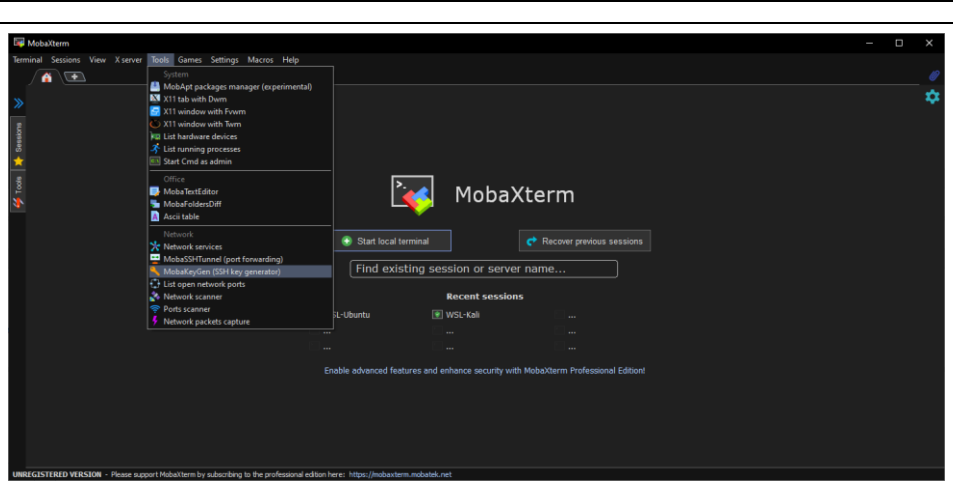
1	<p>a. Open your terminal application of choice</p> <p>b. Generate the SSH key using the following command.</p> <pre>[user@workstation]\$ ssh-keygen -t ed25519</pre>
2	<p>a. Fill in the required information</p> <p>b. Introduce a strong <i>passphrase</i></p> <p>Important</p> <p>It is mandatory to encrypt the key with a strong passphrase with at least 16 characters.</p>

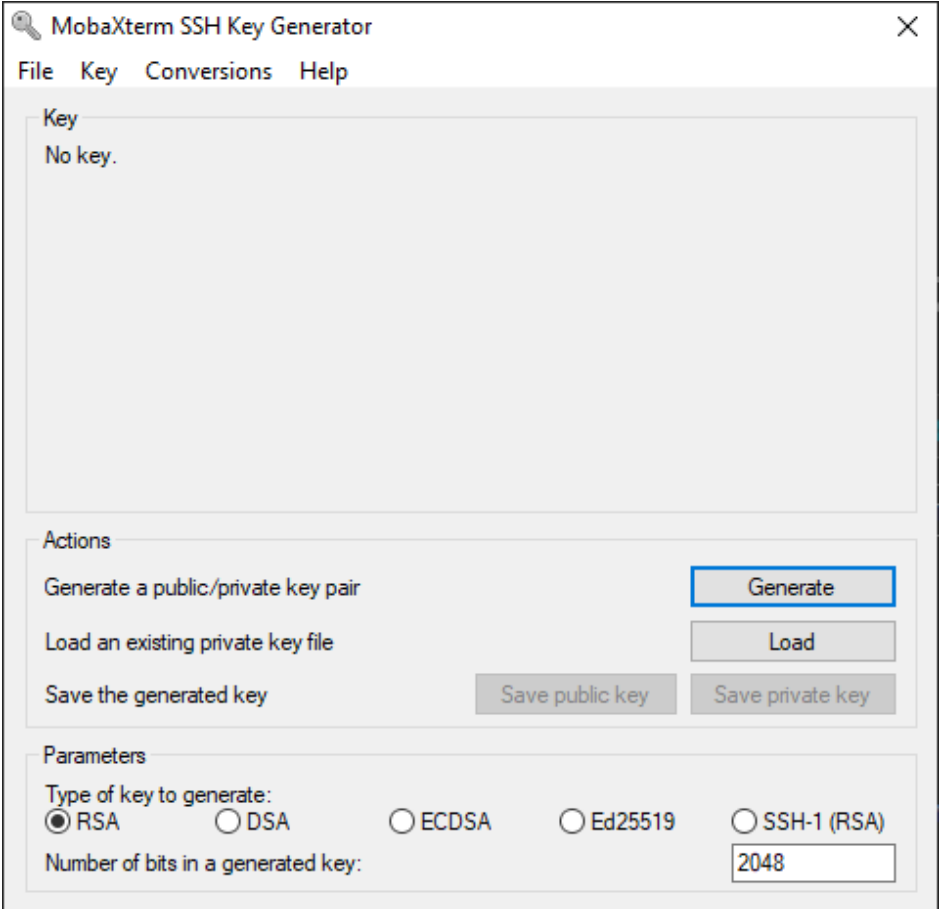
3	<p>Example</p> <pre>[user@workstation]\$ ssh-keygen -t ed25519 Generating public/private ed25519 key pair. Enter file in which to save the key (/home/<username>/.ssh/id_ed25519): Enter passphrase (empty for no passphrase): ***** Enter same passphrase again: ***** Your identification has been saved in /home/<username>/.ssh/id_ed25519 Your public key has been saved in /home/<username>/.ssh/id_ed25519.pub The key fingerprint is: SHA256:LhpfwohLVJM2h2N/q4UHIJNxzUysZ8pD2J1isJ91sBg bmx@LAPTOP-GHKPS84N The key's randomart image is: +--[ED25519 256]--+ ..* .+Eo* +=@=.=.+ o*BBB . .= Bo.S ..*o = . o o.= * .. + B .. o +----[SHA256]-----+</pre>
4	<p>Copy the public key and rename it to <code><username>.pub</code> (where <code><username></code> should be replaced by your ETH user name).</p> <p>The following command will copy the public key in the right format to your Desktop (assuming that the key was saved to <code>/home/<username>/.ssh/id_ed25519</code>; see the output above):</p> <pre>[user@workstation]\$ ~/.ssh/id_ed25519.pub ~/Desktop/<username>.pub</pre>

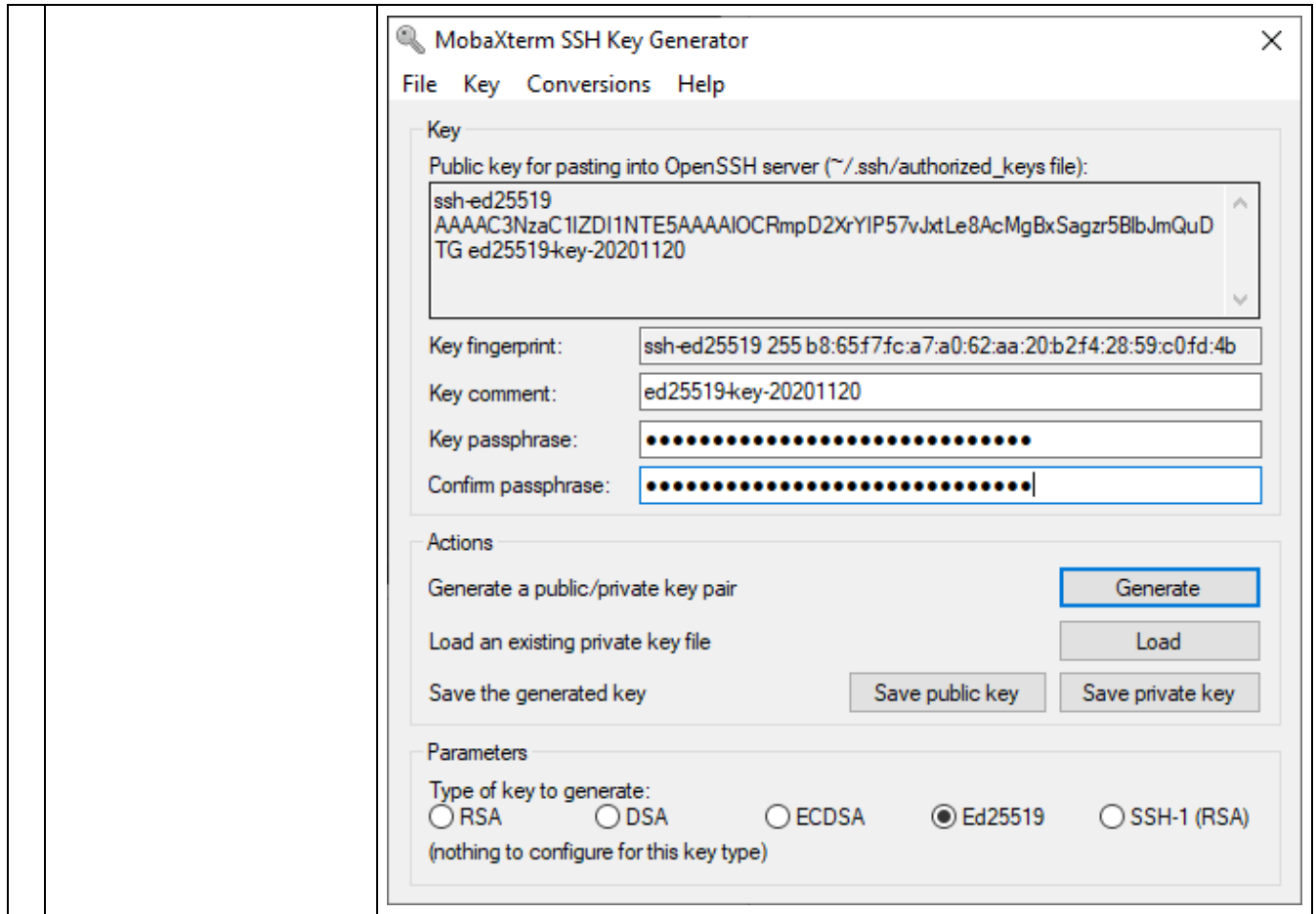
Windows

Important

Windows 10 includes a built-in OpenSSH client since the April 2018 update, however, we will use [MobaXterm](#) for compatibility reasons.

<p>1</p> <ul style="list-style-type: none">a. Open mobaXtermb. Open the <i>Tools</i> menuc. Click <i>MobaKeyGen</i> (SSH key generator)	 <p>The screenshot shows the MobaXterm application window. The 'Tools' menu is open, displaying a list of utilities. The 'MobaKeyGen (SSH key generator)' option is highlighted. Other visible options include 'MobaSSH Tunnel (port forwarding)', 'MobaXterm (SSH key generator)', 'List open network ports', 'Network scanner', 'Ports scanner', and 'Network packets capture'. The main interface shows a 'Start local terminal' button and a 'Recover previous sessions' button. A search bar for existing sessions is also visible.</p>
---	---

<p>2</p> <p>a. Select the type <i>Ed25519</i></p> <p>b. Click <i>Generate</i></p> <p>c. Introduce a strong passphrase</p> <p>Important</p> <p>It is mandatory to encrypt the key with a strong passphrase with at least 16 characters.</p>	 <p>The screenshot shows the MobaXterm SSH Key Generator window. The title bar reads 'MobaXterm SSH Key Generator'. The menu bar includes 'File', 'Key', 'Conversions', and 'Help'. The main area is titled 'Key' and contains the text 'No key.'. Below this is the 'Actions' section with three options: 'Generate a public/private key pair' (with a blue 'Generate' button), 'Load an existing private key file' (with a 'Load' button), and 'Save the generated key' (with 'Save public key' and 'Save private key' buttons). The 'Parameters' section at the bottom has 'Type of key to generate:' with radio buttons for 'RSA' (selected), 'DSA', 'ECDSA', 'Ed25519', and 'SSH-1 (RSA)'. The 'Number of bits in a generated key:' is set to '2048' in a text box.</p>
---	---

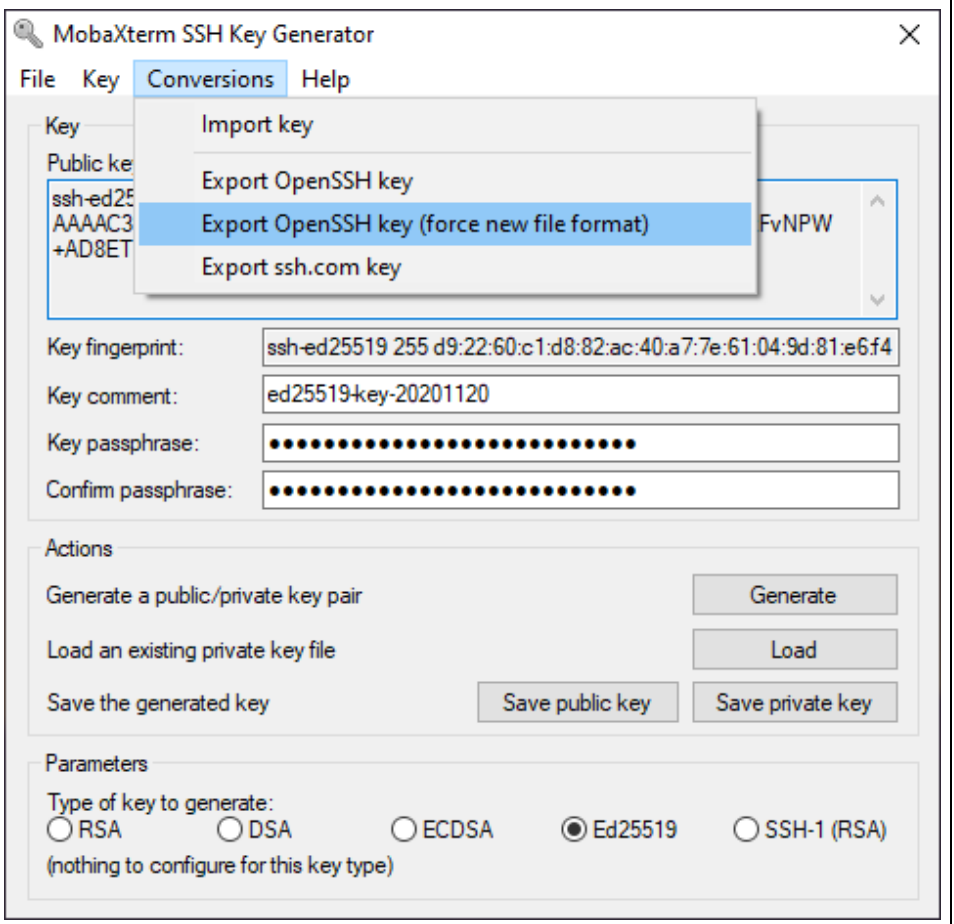


3

Save the *Private key*

a. Open the *Conversions* menu

b. Select *Export OpenSSH key (force new file format)*



The screenshot shows the MobaXterm SSH Key Generator application window. The 'Conversions' menu is open, and the option 'Export OpenSSH key (force new file format)' is highlighted. The application interface includes a 'Key' section with a 'Public key' field containing the text 'ssh-ed25519 AAAAC3...+AD8ET'. Below this, there are fields for 'Key fingerprint', 'Key comment', 'Key passphrase', and 'Confirm passphrase'. The 'Actions' section contains buttons for 'Generate', 'Load', 'Save public key', and 'Save private key'. The 'Parameters' section shows radio buttons for 'Type of key to generate', with 'Ed25519' selected.

4 Save the *Public key* by clicking the *Save public key* button

The screenshot shows the MobaXterm SSH Key Generator application window. The window title is "MobaXterm SSH Key Generator" and it has a menu bar with "File", "Key", "Conversions", and "Help". The main content area is divided into several sections:

- Key:** A text area containing the public key for pasting into an OpenSSH server. The key text is:
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIOCRmpD2XrYIP57vJxtLe8AcMgBxSagzr5BlbJmQuD
TG ed25519-key-20201120
- Key fingerprint:** ssh-ed25519 255 b8:65f7fc:a7:a0:62:aa:20:b2f4:28:59:c0:fd:4b
- Key comment:** ed25519-key-20201120
- Key passphrase:** A field with 20 dots representing a masked password.
- Confirm passphrase:** A field with 20 dots representing a masked password.
- Actions:** A section with four buttons: "Generate" (highlighted with a blue border), "Load", "Save public key", and "Save private key".
- Parameters:** A section with radio buttons for "Type of key to generate": RSA, DSA, ECDSA, Ed25519 (selected), and SSH-1 (RSA). Below this, it says "(nothing to configure for this key type)".

<p>5</p> <p>Open the <i>Public key</i> file you just saved with a text Editor (such as <i>Notepad</i>) and overwrite it only with the content from the box and save it.</p> <p>This is because the format used by Putty/MobaXterm for the public key is not the expected one</p>	 <pre>Key Public key for pasting into OpenSSH server (~/.ssh/authorized_keys file): ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOCRmpD2XrYIP57vJxtLe8AcMgBxSagzr58lbJmQuD TG ed25519-key-20201120</pre>
---	--

How to create a GPG key pair

This document guides you through the steps of creating a GPG key. A new GPG key pair should be created for LeoMed. This pair should only be used for LeoMed.

The GPG key pair is used to encrypt a QR code needed for generating the second factor. This encrypted code will be sent to you via email. You will be able to decrypt this file with your private GPG key.

- the ed25519 algorithm for generating GPG key-pairs is recommended
- the passphrase (non-empty, of minimum 16 characters) protection of the GPG key-pair is recommended
- format convention for sharing the public GPG key: as file named `<username>.gpg`

Terminal

To execute commands in the terminal

1. open your terminal application
2. type or paste the command into the terminal window
3. hit enter

Example:

```
[user@workstation]$ ls
```

1. open your terminal application
2. type "ls" into the terminal window (*[user@workstation]\$* indicates the machine you are currently working on and might look different depending on your settings)
3. hit enter

Linux / Unix (macOS)

1	<p>Install the GPG software using these terminal commands</p> <p>a. Linux</p> <pre>[user@workstation]\$ sudo apt-get install gnupg2</pre> <p>b. macOS</p> <p>After installing homebrew, you can use the `brew` command to install GPG</p> <pre>[user@workstation]\$ brew install gpg</pre>
2	<p>a. Open your terminal application of choice</p> <p>b. Generate the GPG key using the following command.</p> <pre>[user@workstation]\$ gpg --gen-key</pre>
3	<p>Fill in the required information including your real name and your work email</p> <p>Important</p> <p>Please avoid the use of special characters like umlauts</p>

Example

```
[user@workstation]$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

```
Real name: John Smith
Email address: john.smith@email.corp
You selected this USER-ID:
  "John Smith <john.smith@email.corp>"
```

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O

- 4 We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.
gpg: key 94F3B324F9CA5741 marked as ultimately trusted
gpg: revocation certificate stored as '/home/<username>/.gnupg/openpgp-revocs.d/21AF418DC988D4B85B81C05894F3B324F9CA5741.rev'
public and secret key created and signed.

```
pub  rsa3072 2020-11-20 [SC] [expires: 2022-11-20]
     21AF418DC988D4B85B81C05894F3B324F9CA5741
uid           John Smith <john.smith@email.corp>
sub  rsa3072 2020-11-20 [E] [expires: 2022-11-20]
```

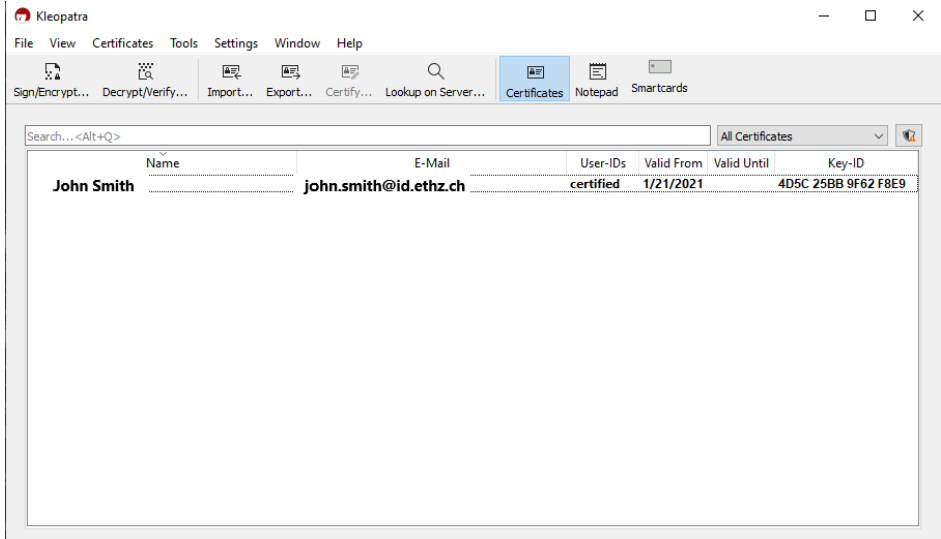
Export public key

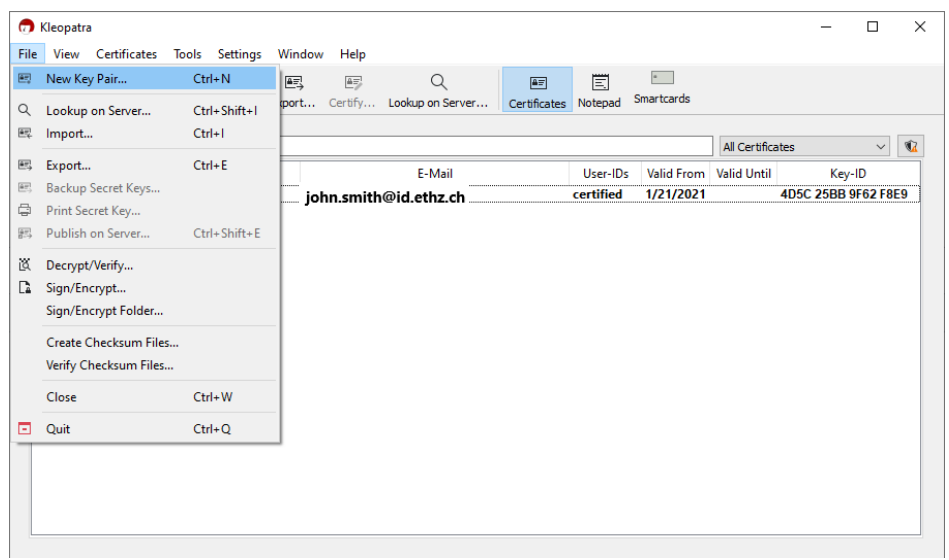
To export the public key into a file (for instance, on your Desktop), run the following command:

5 [user@workstation]\$ gpg -a --output ~/Desktop/<username>.gpg --export <key_ID>

- <username> should be replaced by your LeoMed username
- <key_ID> should be replaced by the GPG key's ID (in the above example, "21AF418DC988D4B85B81C05894F3B324F9CA5741"; the `gpg --list-key` command can also be used to retrieve the key ID.)

Windows

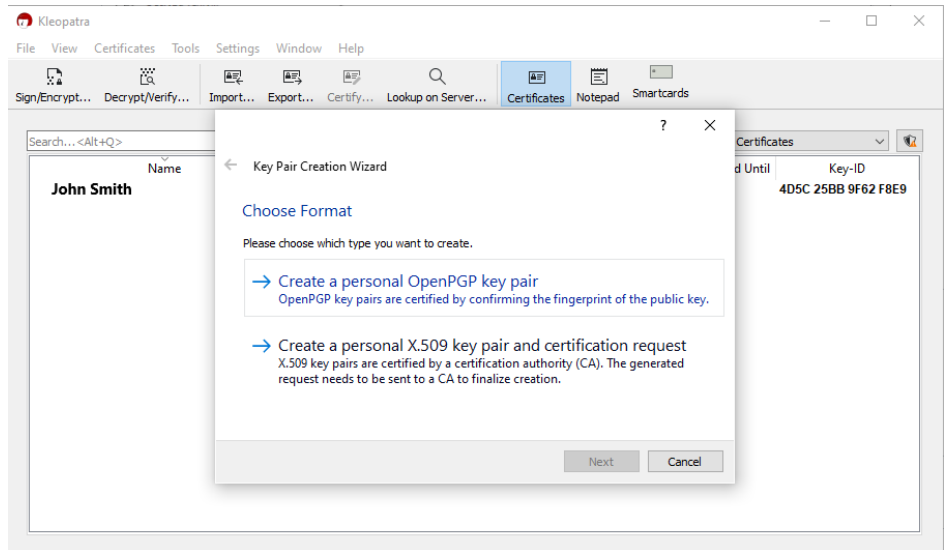
<p>a. Install GPG4Win (also known as Kleopatra)</p> <p>b. Open Kleopatra</p> <p>1 c. Create a new GPG key pair</p> <p>d. Select Create a personal OpenPGP key pair</p>	<p>b. Kleopatra UI</p>  <p>The screenshot shows the Kleopatra application window. The title bar reads 'Kleopatra'. The menu bar includes 'File', 'View', 'Certificates', 'Tools', 'Settings', 'Window', and 'Help'. The toolbar contains icons for 'Sign/Encrypt...', 'Decrypt/Verify...', 'Import...', 'Export...', 'Certify...', 'Lookup on Server...', 'Certificates', 'Notepad', and 'Smartcards'. Below the toolbar is a search bar with the text 'Search... <Alt+Q>'. A table displays a list of certificates:</p> <table border="1"><thead><tr><th>Name</th><th>E-Mail</th><th>User-IDs</th><th>Valid From</th><th>Valid Until</th><th>Key-ID</th></tr></thead><tbody><tr><td>John Smith</td><td>john.smith@id.ethz.ch</td><td>certified</td><td>1/21/2021</td><td></td><td>4D5C 25BB 9F62 F8E9</td></tr></tbody></table>	Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID	John Smith	john.smith@id.ethz.ch	certified	1/21/2021		4D5C 25BB 9F62 F8E9
Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID								
John Smith	john.smith@id.ethz.ch	certified	1/21/2021		4D5C 25BB 9F62 F8E9								



The screenshot shows the Kleopatra application window. The 'File' menu is open, displaying options such as 'New Key Pair...', 'Import...', 'Export...', and 'Decrypt/Verify...'. The main window displays a table of certificates:

E-Mail	User-IDs	Valid From	Valid Until	Key-ID
john.smith@id.ethz.ch	certified	1/21/2021		4D5C 25BB 9F62 F8E9

d. Key Pair creation



The screenshot shows the 'Key Pair Creation Wizard' dialog box. It prompts the user to 'Choose Format' and offers two options:

- Create a personal OpenPGP key pair
OpenPGP key pairs are certified by confirming the fingerprint of the public key.
- Create a personal X.509 key pair and certification request
X.509 key pairs are certified by a certification authority (CA). The generated request needs to be sent to a CA to finalize creation.

The dialog box includes 'Next' and 'Cancel' buttons.

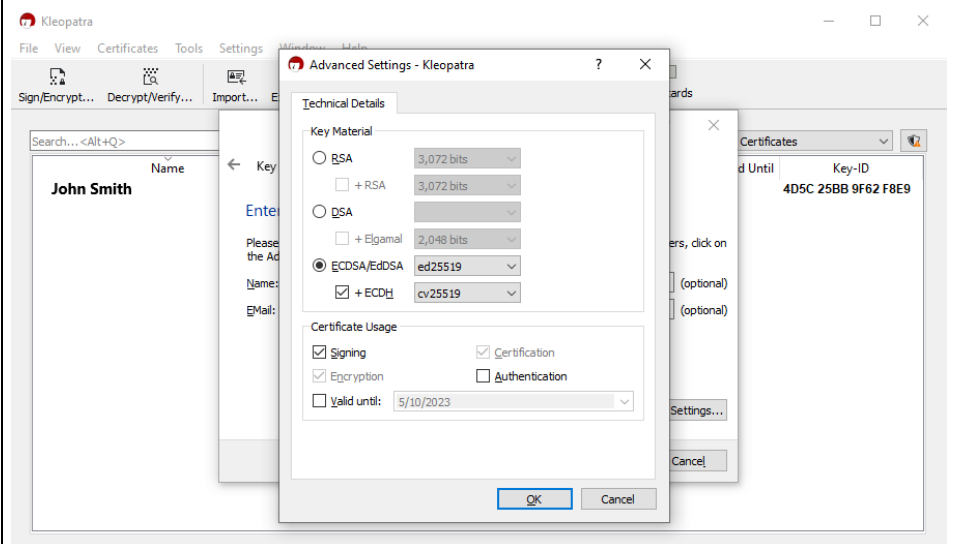
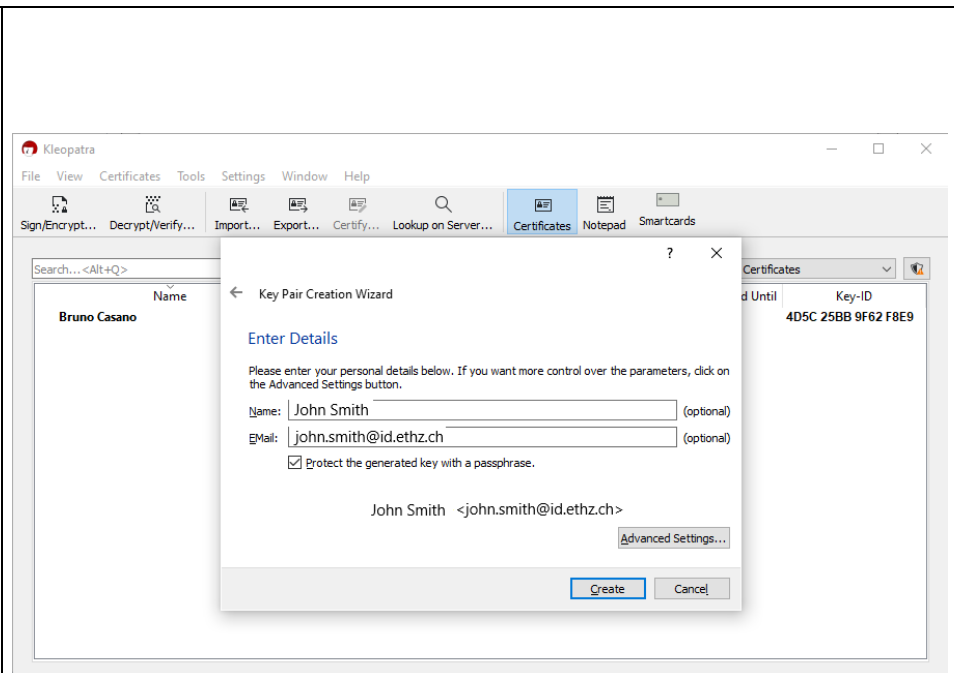
a. Fill in the required information including your real name and your work email

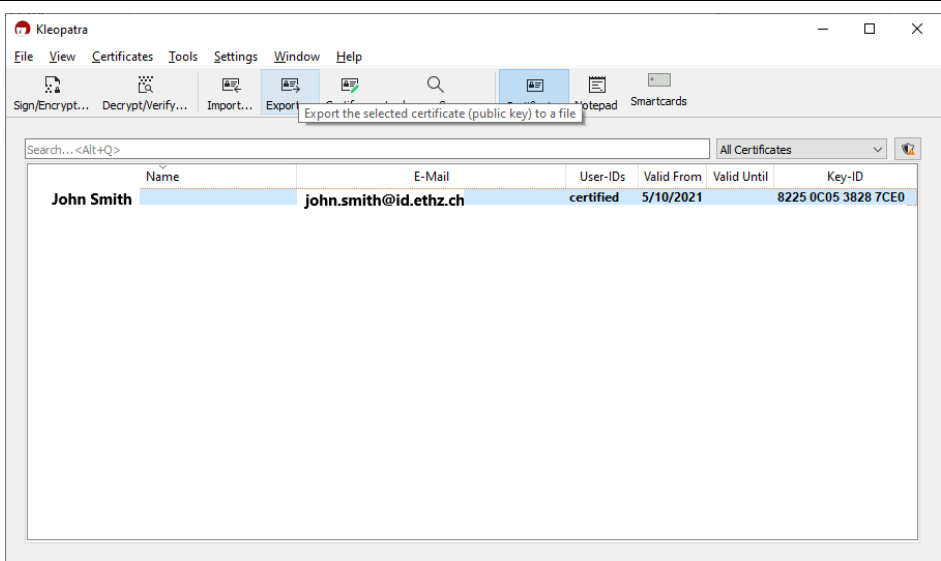
b. Make sure you protect the pair with a passphrase by checking the *Protect the generated key with a passphrase* checkbox.

2 c. Click the *Advanced Settings* button and update the *Key Material* to ECDSA/EdDSA and un-select the *Valid until* checkbox, as shown in the screenshot

Important

Please avoid the use of special characters like umlaut



<p>3 Export public key to a file</p>	 <p>The screenshot shows the Kleopatra application window. The 'Export' menu item is highlighted, and a tooltip reads 'Export the selected certificate (public key) to a file'. Below the menu, a table lists certificates. The first entry is selected:</p> <table border="1"><thead><tr><th>Name</th><th>E-Mail</th><th>User-IDs</th><th>Valid From</th><th>Valid Until</th><th>Key-ID</th></tr></thead><tbody><tr><td>John Smith</td><td>john.smith@id.ethz.ch</td><td>certified</td><td>5/10/2021</td><td></td><td>8225 0C05 3828 7CE0</td></tr></tbody></table>	Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID	John Smith	john.smith@id.ethz.ch	certified	5/10/2021		8225 0C05 3828 7CE0
Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID								
John Smith	john.smith@id.ethz.ch	certified	5/10/2021		8225 0C05 3828 7CE0								



How to assemble the required information in the correct format

1. Make sure your public GPG key file is called `<username>.gpg`, where `<username>` is to be replaced by your LeoMed user name.
2. Make sure your public SSH key file is called `<username>.pub`. Only required if you request command-line access to LeoMed.
3. Prepare a text file named `<username>.addr` containing your work-related contact information.

Dos and don'ts

These are some examples of the things we expect from the `.addr` file, and some things we don't support.

- Avoid the use of umlaut or special characters
- Avoid the use of "field: value" format, i.e., Name: Andrea Mueller

Good

```
Andrea Mueller  
Zuerich  
andrea.mueller@foo.ethz.ch
```

Bad

```
Andrea Müller  
Zürich  
email: andrea.mueller@foo.ethz.ch
```