

# Public Key Authentication

The access.eps.surrey.ac.uk service requires that you either have Multi Factor Authentication setup on your Surrey account or use Public Key authentication. For more information on MFA and how to set it up on your Surrey account please see [here](#).

## Public Key Authentication

Key pairs work on the basis that a username or password can be stolen, but a token is something you have, and thus harder to steal. The key pair consists of two elements, both of which are files stored on computers:

**Public Key** – This is what you distribute to the systems you wish to log in to. This part of the key is considered public, and thus can be stolen, transferred, or copied without any fears.

**Private Key** – This part of the key pair is stored by you on the system you wish to connect from. If it is stolen then it is considered compromised, and both the Private Key and Public Key's must be replaced. To help protect from theft / misuse of a Private Key, it should always be protected by a passphrase.

Once the key pair is in place, logging in will prompt you for your username. Once entered, you will be prompted for the passphrase for your Private Key (stored on your local computer). Once entered, you will be successfully logged in to the remote compute.

## Setting Up Public Key Authentication

### Step 1 – Setting up the .ssh folder:

These instructions should be carried out on a University Linux machine, either in person or via the Global Protect VPN service.

First we need to check if the .ssh directory exists in your home directory. Your Linux home path is denoted by the ~ (tilda) symbol. Please be sure to type all commands exactly and double check before you press Enter.

From the command line or terminal, run the following command to create your SSH directory and press Enter (take special note of the . ahead of ssh):

```
mkdir ~/.ssh
```

Now run the following command to change directory (cd) into the newly created (or existing) .ssh directory

```
cd ~/.ssh
```

Now that you should be in your `.ssh` directory, we can check the contents of this to see if any existing key pairs exist. Run the following command to print a file list:

```
ls -lah
```

If your `.ssh` directory already existed, you should see output similar to the following:

```
drwx----- 1 <user> itsstaff 296 Oct 23 11:34 .
drwx----- 1 <user> itsstaff 8.0K Nov 7 12:19 ..
-rw----- 1 <user> itsstaff 394 Apr 7 2017 authorized_keys
-rw----- 1 <user> itsstaff 1.4K Oct 23 11:34 config
-rw----- 1 <user> itsstaff 1.8K Mar 1 2017 id_rsa
-rw----- 1 <user> itsstaff 394 Mar 1 2017 id_rsa.pub
-rw-r--r-- 1 <user> itsstaff 16K Nov 7 12:19 known_hosts
```

In this example we can see the `.ssh` directory has existed for some time and contains files. We can also see the existence of the `id_rsa` and `id_rsa.pub` files. If you have not previously set up a key pair then these two files will not exist yet. If you already see them, you already have a key pair created. `id_rsa` is your private key, while `id_rsa.pub` is your public key.

## Step 2 – Creating your key pair:

Now that the relevant `.ssh` folder exists and we've confirmed that it does not contain existing keys, it is time to generate a new key pair.

WARNING: The following commands will delete any existing RSA keys.

From the command line, run the following command to begin your RSA key pair generation.

```
ssh-keygen
```

The following prompt will be displayed:

```
[1:38pm] <computername>: > ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/user/<path>/<user>/.ssh/ id_rsa):
```

The value described in brackets (`/user/<path>/<user>/.ssh/id_rsa`) is the default path. Press Enter to accept this path. Moving this to a different path may render the key pair unusable.

Now you will be prompted to enter a passphrase. This secures your private key.

WARNING: Do not leave this blank otherwise if your private key is stolen it could be used by the malicious party to gain entry to your account or any other computer containing your public key.

Please enter your passphrase now, then press enter.  
Now you will be prompted to enter the passphrase again.

Once you press Enter your key will be generated and the following information displayed to you:

```
Your identification has been saved in /user/<path>/<user>/.ssh/id_rsa.  
  
Your public key has been saved in /user/<path>/<user>/.ssh/id_rsa.pub.  
  
The key fingerprint is: SHA256:9J5U1UjIYX+TL1gEvXkrKCCynM61aIM5qpDZ3AuR8T0  
<user>@<computer>  
  
The key's randomart image is:
```

Your keys have now been successfully generated.

Under the directory `~/.ssh` or `/user/<path>/<user>/.ssh` you will have created the following files

- `id_rsa`

Your private key, protected by your passphrase. This is stored on the machine you are connecting from.

- `id_rsa.pub`

Your public key. This is stored in the account you are connecting to.

It is now important to ensure the permissions are correctly set for the `.ssh` folder that you created. To do this, run the following commands:

```
chmod 700 ~/.ssh
```

```
chmod -R 600 ~/.ssh/*
```

The first command will set the correct permissions for the `.ssh` directory, while the second will set the correct permissions for all files in the directory.

If you attempt to use an private key with an insecure `.ssh` directory you will receive an error stating that the key is not secure.

## Step 4 – Deploying your keys:

In order to use this key pair, you will need to move the private key to a computer you wish to connect from, and the public key must be located on the computer you wish to connect to in the 'authorized\_keys' file.

### Private Key

On Linux or Mac machines, your private key (id\_rsa) needs to be stored in the correct folder so that it can be accessed by the operating system. By default, this is the ~/.ssh directory (~/.ssh/id\_rsa), just like on the Surrey Linux machines. If you have previously used SSH on that account, the .ssh directory should already exist, but if it doesn't you can follow the steps in this guide to create it.

On Windows Machines - Your private key file (id\_rsa) can be stored anywhere on the system, but usually somewhere inside of your Home Directory. When establishing an SSH connection using PuTTY, WinSCP or FileZilla you can specify the path to your id\_rsa file.

Note: For PuTTY / WinSCP you will be required to convert your keys to the .ppk format using PuTTYgen (part of the PuTTY suite).

### Public Key

Your public key (id\_rsa.pub) can be deployed to any account on any system that you wish to establish an SSH connection to. As SSH is exclusively a Unix/Linux service (including MacOS), the location for the public key should always be inside the ~/.ssh directory, in the 'authorized\_keys' file (~/.ssh/authorized\_keys). To add your key to this file, run the following command to append the key to the end of an existing authorized\_keys file, or to generate a new file if it does not already exist

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

## Step 5 - Using your key pair:

You can test if your key pair is working by establishing an SSH connection to access.eps.surrey.ac.uk from any computer containing your private key. As your public key is located in the .ssh directory your Surrey home area, logging onto any Linux machine will use the key pair over your password.

During login, you should be prompted for:

- Username
- Private Key passphrase

Now that you have successfully tested your key pair, you are able to use the public key on any system that you wish to SSH into, while the private key must be on the machine you wish to connect from.

## **Fallback**

SSH will always prefer to use key pairs over username and password as it is considered more secure, but should your key pair not be available on the system it will fall back to password authentication.

Please note that fallback to username and password on [access.eps.surrey.ac.uk](https://access.eps.surrey.ac.uk) is not available externally. If a key pair is not available, then the system will fall back to username, password and MFA.

It is recommended that all key pair users still set up MFA.

## **Additional Information**

### **If you occasionally get disconnected when using SSH, resulting in the program running to stop working.**

Using the command 'screen' followed by the command you wish to run will detach the command from your active session. This means that if your connection is lost, the screen / session containing your application is able to be restored.

### **If you need to run computational applications from outside the University.**

If you wish to run computational applications, please establish your connection to [access.eps.surrey.ac.uk](https://access.eps.surrey.ac.uk) before then creating another SSH connection from there to the machine you wish to run the application on (i.e. your own desktop or dedicated application or departmental server).

### **Are other key types supported apart from RSA?**

Surrey Ubuntu 18.04 and Ubuntu 20.04 machines support RSA, ECDSA and Ed25519 key algorithms.