CSE484 SSHing and SCPing Guide

Note: Anything written like `<a_thing>` is something you replace.

Generating Key Pairs

To generate a key-pair run the following in a terminal (it is strongly suggested to use a passphrase):

```
ssh-keygen -t ed25519 -f <file_name_for_key>
```

This will generate *two* files: `<file_name_for_key>` and `<file_name_for_key>.pub`
.pub is your public key, the other is your private key.

To use ssh (after we have created your account):

```
ssh -i <path_to_private_key> <username>@cse484.cs.washington.edu
```

A note for Windows:

We're assuming you are using the official "Windows Terminal" application (https://apps.microsoft.com/store/detail/windows-terminal/9N0DX20HK701)
You can also consider installing and using "windows subsystem for linux"

SSH Config Files (Recommended!)

You might want to make use of the `~/.ssh/config` file on your machine. That way, you can set up the log-in name for the cse484 server as your group name. An example would be:

```
Host cse484-lab1
    HostName cse484.cs.washington.edu
    User cse484-24sp-lab1-X
    IdentityFile <path to ssh private key>
```

Of course, replace cse484-24sp-lab1-X with the real name of your group and replace the path to your ssh private key. Then you should just be able to `ssh cse484-lab1` to get on cse484.
This will also transparently work for `scp`!

Using scp to copy files from cse484

SCP (Secure CoPy) is a tool that copies files, and understands how to do so over ssh.
Just like ssh, it can use either keys or a password to connect to remote machines. You don't have a password on cse484, so you'll need to give scp your ssh private key.

`scp -i <path_to_a_private_key> <username>@cse484.cs.washington.edu:turnins/<rest_of_the_path_to_your_file> <destination>`
Will copy the file from cse484 to `<destination>` on your local machine.

If you have an ssh config file setup as explained above, you can do
`scp cse484-lab1:turnins/<rest_of_the_path_to_your_file> <destination>`

If you generated your ssh keys on attu, and are using them there, you have two options:

- Scp the file from cse484->attu and then from attu->your local machine
- Scp your ssh private key from attu->your local machine, and then scp the file from cse484->your local machine.

An example workflow (from a terminal):

1. Join a lab1 premade group on Canvas. Let's assume you joined group Lab 1 X:
2. Open a terminal on the machine you will be sshing into cse484 from.
3. Move to a directory that you would like to save the SSH keys for this lab
4. `ssh-keygen -t ed25519 -f 484lab1` Will generate new ssh keys, using the ed25519 cryptosystem, and saved to two files:
    a. `484lab1` (the private key) and
    b. `484lab1.pub` (the public key).
5. It is good practice to set a pass**phrase** (even if it's short, see FAQ for details) for your private key when asked. Make sure you remember the passphrase for your private file.
6. `484lab1.pub` (your public ssh key) is what we need in the google form
7. Below is an example of what it looks like (it is all one line of text):
    a. `ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOHK0eNCjWCsVX/otyjCFadkLe89W9Ep4Mqk00R 0ODdp user@localhost`
8. Log in into your uw account (cse account won't work). Then submit your key to the form.
9. Wait 10-15 minutes.
10. Now you should be able to to connect:

`ssh -i <path_to_private_key> <username>@cse484.cs.washington.edu`
since you are in the same directory, `<path_to_private_key>` will just be ./484lab1.
and `<username>` will be cse484-24sp-lab1-X with X as the lab 1 group you joined at step 1.

Then you will see below message prompted in your terminal:
`Enter passphrase for key '484lab1':`
Now, enter the passphrase you created for your private key file in step 4.
You should now be logged into cse484 under your user account!

FAQ

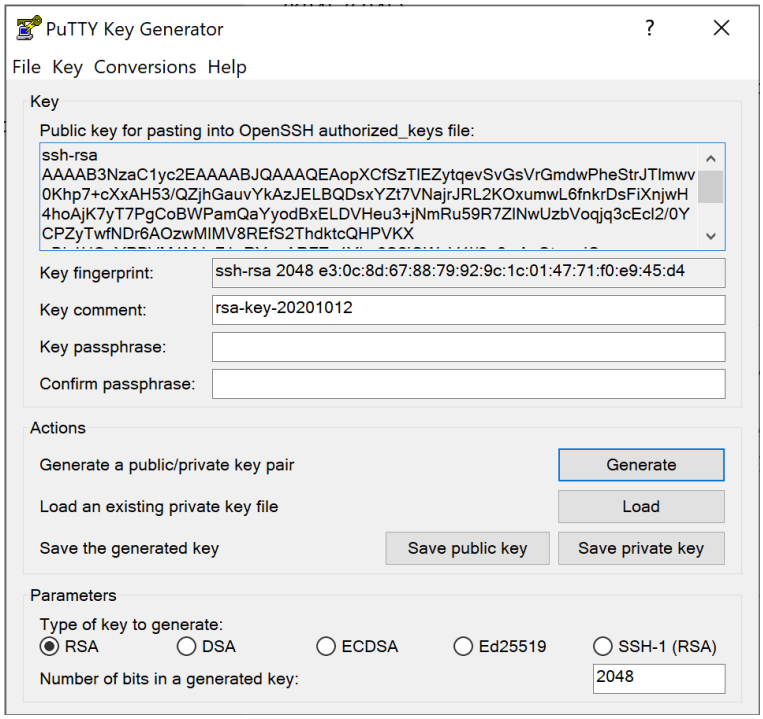
- What is a passphrase and how does it differ from a password for SSH?
    - A pass**phrase** is an optional password-like protection for your local private key file. It is never transmitted to the server, it is like a lock on the file itself.
    - A pass**word** is one way to login to an SSH server, and is the default way you connect to (for example) attu. Our server does not support password logins, on purpose.
- I'm getting asked for a password to login to cse484? What password?
    - Something didn't work right, and ssh is trying to fall back to password-based login
        - Your path to your private key may be specified incorrectly
        - Or, you didn't specify the right username.
        - Note that if you give an invalid username, it'll still ask for a password!
    - You don't have a password on cse484, so if you ever see a pass**word** prompt, something is wrong with your connection attempt, and nothing will succeed.
- Do I have to set a passphrase for my ssh key?
    - Nope. Just a good idea in general.
- I want to use VSCode or JetBrains or some other tool to edit code on cse484 directly, can I do that?
    - Sure! All those tools allow you to edit files over ssh, you'll need to look up instructions for your specific tool for how to do that. It'll need access to the ssh private key you generated.
    - A note that these tools accomplish this goal by automatically installing a (quite large, >1GB) server on the remote machine you are connecting to. This may cause them to run somewhat slowly.
    - If you encounter problems, you may want to try sshing from a terminal first to isolate if the problem is with VSCode/JetBrians configuration or your ssh key.
- It seems like nothing works and ssh won't accept my private key!
    - One thing to check is if the permissions on your ssh private key file got changed from the defaults. SSH *requires* that your private key file not be world-readable or writable for safety reasons. It will explicitly tell you this if you try to ssh from the terminal and the file permissions are not reasonable.
    - You can check permissions with `ls -l`
    - chmod is the linux tool for changing file permissions. Specifically, you can remove (r)ead, (w)rite, and e(x)ecute permissions from all (o)ther users on the system with:
        - `chmod og-rwx <filename>`
- I see a big warning when I try to ssh to attu that complains about keys not matching and doesn't let me ssh into attu?
    - Attu changed their ssh _server_ keys back in Sept 2022 (there were some emails about it, but you may not have realized the implications)
    - That is fine, follow the instructions the error message gives you to remove the old attu host/server key. An easy way is to use ssh-keygen to remove any host keys belonging to attu it has seen (`ssh-keygen -R attu.cs.washington.edu`) Then

try sshing in again and accept the new key (expected thing attu tells you in Apr 2023 is:

```
The fingerprint for the ED25519 key sent by the remote host is
SHA256:3hMf5WNtVf6ppUEVahHXD8x538yE6YX3S/w8hrd1KxM
```

Legacy notes for using PuTTY

1) Open PuTTYgen
2) Select the type of key as EdDSA (or RSA)
3) Click Generate and move the mouse around to generate entropy
4) (Optional but recommended) Enter a Passphrase + Confirmation of Passphrase
5) Click save the private key
6) Copy the text of the public key to post/email from the box at the top



Format conversion:

If the public key you get looks like the SSH2 format shown below, you will need to convert both public and private keys to the OpenSSH format.

---- BEGIN SSH2 PUBLIC KEY ----
Comment: " user@localhost"
AAAAB3NzaC1yc2EAAAADAQABAAABAQDTKPi45wxeSezgO5JmG8HiuAQH6R3kqQTeOeT
bntWxliiClrahwlnkv26PAIaQKNdRbVH1fgX9kyUfsdj5JAvvNFuxpfY+GVVZKFI5M3CuzAynIy
mBjqnDn6Auq+tuSl8O4osb/0L9zDeQzOxQ+ed6iVDuPPkBLoX+XyuNUyYKV46xCIHOS6ao+
6CkZXhp4VTz4LUvb1s8DIUcaD8/bbigxxZH3eKRQH2arV9AqP1LoC2T3azLTkHvCrcImpjVW
/pxf5+nbkRb1SSkkHFvFPdd+0us12yGOp1xBbo2kuKWSdcBgd4eiGHQsO+VWi23R92bcOh
/DxRZumdMyaDBMGY/
---- END SSH2 PUBLIC KEY ----

To convert the private key: go to the "Conversions" menu in PuTTYgen, and click on "Import key." Select your .ppk private key file. Then go to "Conversions" again, and click on "Export OpenSSH key." Save the exported private key file and you should be able to log in with it.

To convert the public key: use command ssh-keygen -i -f *ssh2.pub* > *openssh.pub*

To ssh (with PuTTY):

On the left side, select Connection->SSH->Auth. In this pane, browse to your private key, and then login as usual.

Enter <username>@cse484.cs.washington.edu in the Host Name field (where <username> should resemble cse484-23sp-lab1-X), and 22 for the Port field. For Connection Type, make sure you select SSH.

You may want to save the session for a quicker login next time. (Note, if you generated your ssh key pairs using Linux and you want to use it in windows, you will need to use PuTTYgen to convert it from .pem to .ppk before using it)

How do I use PuTTY to log in to cse484?

Under Session, specify cse484.cs.washington.edu for the Hostname. Leave the port as is (was 22 for me). Connection type should be SSH by default.
(Optional) Under Connection >> Data, specify your group name in Auto-login Username, so it won't prompt you each time you log in.
Under Connection >> SSH >> Auth, click Browse for Private key file for authentication, and navigate to your private key (a .ppk file).
Go back to Session, and under Saved Sessions, give your cse484 settings a name. Click Save. Now you can quickly load cse484 when you start up PuTTY!