UNIVERSITY OF CYPRUS

Computer Science Department

ΕΠΛ 660 – Information Retrieval and Search Engines

A guide for connecting to LInC[1]-powered cloud-based Virtual Machines

Lab instructor: Pavlos Antoniou

Date: 11/09/2020

http://www.cs.ucy.ac.cy/courses/EPL660

I. Introduction

Secure Shell (**SSH**) is a cryptographic network protocol for operating network services securely over an unsecured network. The best known example application is for remote login to computer systems by users. **SFTP**, which stands for SSH File Transfer Protocol, or Secure File Transfer Protocol, is a separate protocol packaged with SSH for transferring files between two remote systems over a secure connection. The advantage is the ability to leverage a secure connection to transfer files and traverse the filesystem on both the local and remote system.

In order to be able to connect (for login or file transfer) to a Virtual Machine (VM) running on a remote machine we need:

a) the IP address of the VM,
b) an SSH client for secure remote login,
c) an SFTP client for secure remote file transfer.

An SSH client logins securely (i.e. over an encrypted connection) to port 22 of the VM given that a dedicated SSH server is installed and configured to accept new connections. This service is already installed on the given VMs. Therefore, we can login from the command line interface using PuTTY. The secure remote file transfer is also enabled by the SSH server running on the VM so we are able to transfer files from/to VM using an sftp client such as Filezilla, WinSCP (for windows only). In case we want to gain access to a (more appealing) graphical interface we may use, for example, the X2Go client. The latter option requires that the x2go server is additionally installed and configured on the VM (which is the case for your VMs).

Prior connecting to the VM using any of the aforementioned options, a secure, authenticated, channel must be established between our machine and the remote VM.

[1] The VMs provided for the lab of EPL660 run on the cloud infrastructure of the Laboratory for Internet Computing (LInC) of the Computer Science Department. LInC focuses its research activities on three important areas of Internet Computing, namely Cloud and Grid computing, Web data management and Vehicular computing. The person in charge of providing the cloud-based VMs is Mr. Athanasios Tryfonos (a.tryfonos@cs.ucy.ac.cy).

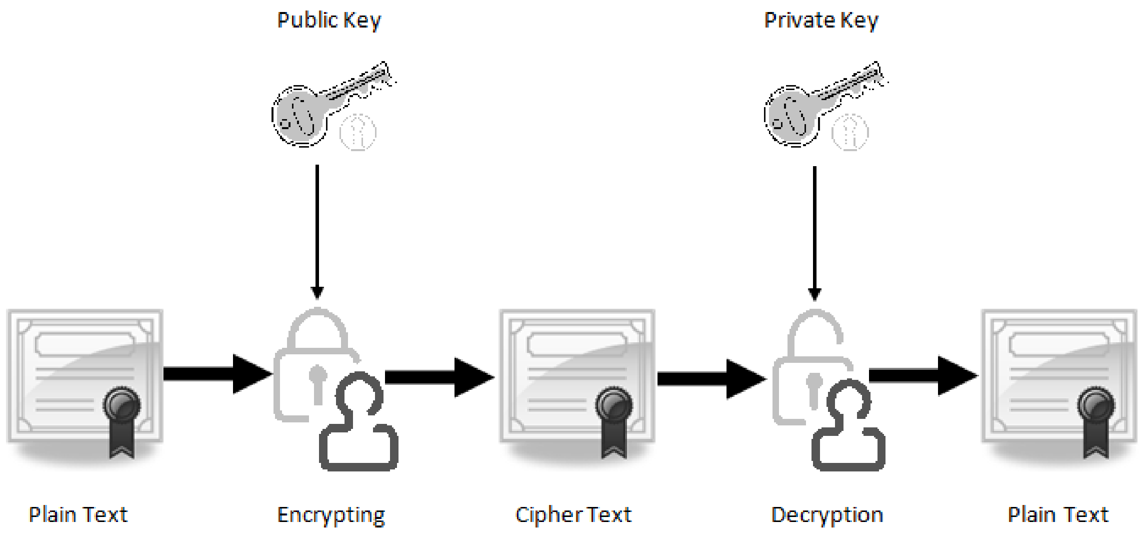
An SSH server can authenticate clients using a variety of different methods. The most basic of these is password authentication, which is easy to use, but not the most secure. Although passwords are sent to the server in a secure manner, they are generally not complex or long enough to be resistant to repeated, persistent attackers. Modern processing power combined with automated scripts make brute forcing a password-protected account very possible. Although there are other methods of adding additional security (fail2ban, etc.), SSH keys prove to be a reliable and secure alternative. The latter method will be used in our case.

SSH key pairs are two cryptographically secure keys that can be used to authenticate a client to an SSH server. Each key pair consists of a public key and a private key. This method is referred to as "public key cryptography". The public-private key pairs are generated for each user who is uniquely identified by his/her username. In our case, the username is the same for all teams, namely, **ubuntu**. A small description to the public key cryptography is given below in Section II. Section III describes the process for importing public-private keys into the client tools presented above, namely, PuTTY, Filezilla and X2Go client.

II. Public key cryptography

Public key cryptography, or asymmetrical cryptography, is any cryptographic system that uses pairs of keys: **public keys** which may be disseminated widely, and **private keys** which are known only to the owner. This accomplishes two functions: authentication, which is when the public key is used to verify that a holder of the paired private key sent the message, and encryption, whereby only the holder of the paired private key can decrypt the message encrypted with the public key. In our case, will we use the public-private key pair for encryption.

In a public key encryption system, any person can encrypt a message using the public key of the receiver, but such a message can be decrypted only with the receiver's private key. For this to work it must be computationally easy for a user to generate a public and private key-pair to be used for encryption and decryption. The strength of a public key cryptography system relies on the degree of difficulty (computational impracticality) for a properly generated private key to be determined from its corresponding public key. Security then depends only on keeping the private key private, and the public key may be published without compromising security.



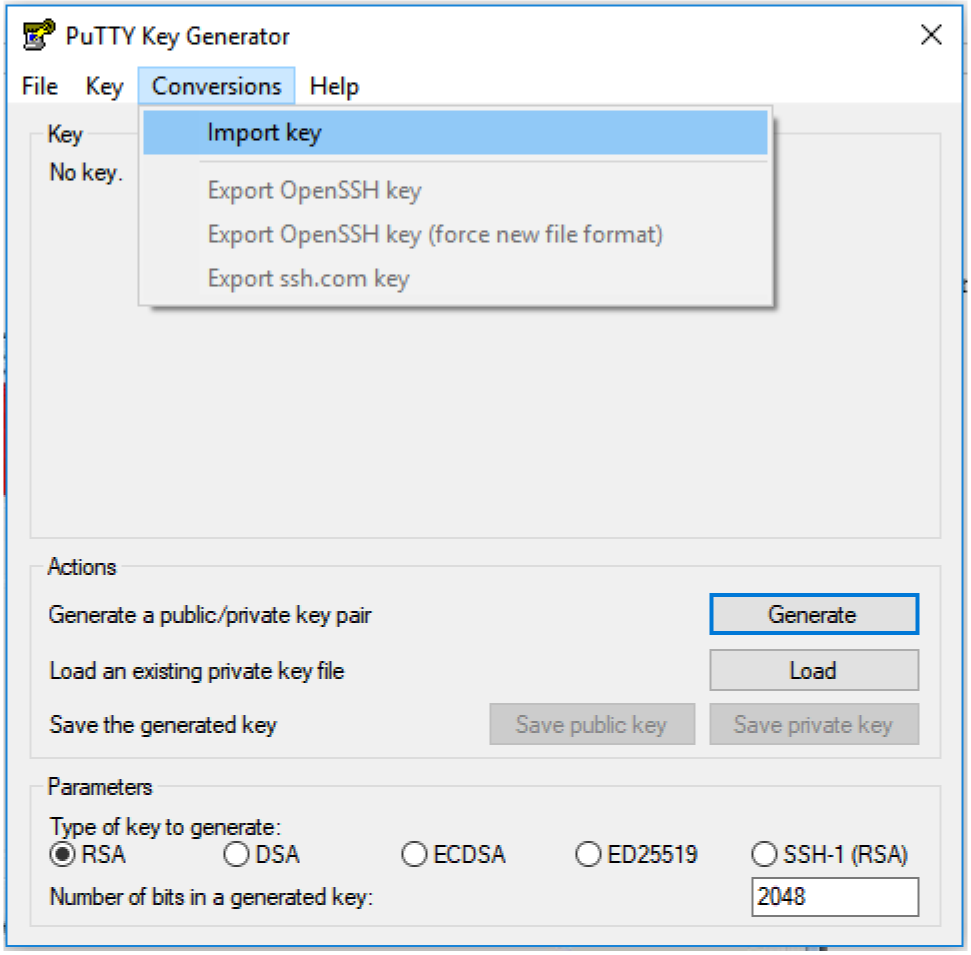
In order to establish a two-way (bidirectional) secure channel between 2 parties (i.e. our machine and the remote VM), both parties must generate[2] a public-private key pair and make their public key available to the other party. In our case, to make things less complicated, all public-private key pairs were generated by the cloud owner (LiNC). **The private key for each person was exported to a dedicated .pem file** and was sent to you via email. This key must be imported to the all client tools that will be used to connect to your VM (see next section). During the secure channel establishment, the public key of the VM will be sent to connecting client tool. The VM public key will be used by the client tool to encrypt data sent to VM.

III. Connect to VM using PuTTY, X2Go, Filezilla

PuTTY

PuTTY cannot import files in the PEM format, instead it uses its own file format called PPK. Therefore, you have to convert the given .pem file to .ppk file using the PuTTYgen tool.

After downloading and installing PuTTYgen tool, the .pem file is imported as shown in the next image.



No key.

Generate a public/private key pair

Load an existing private key file

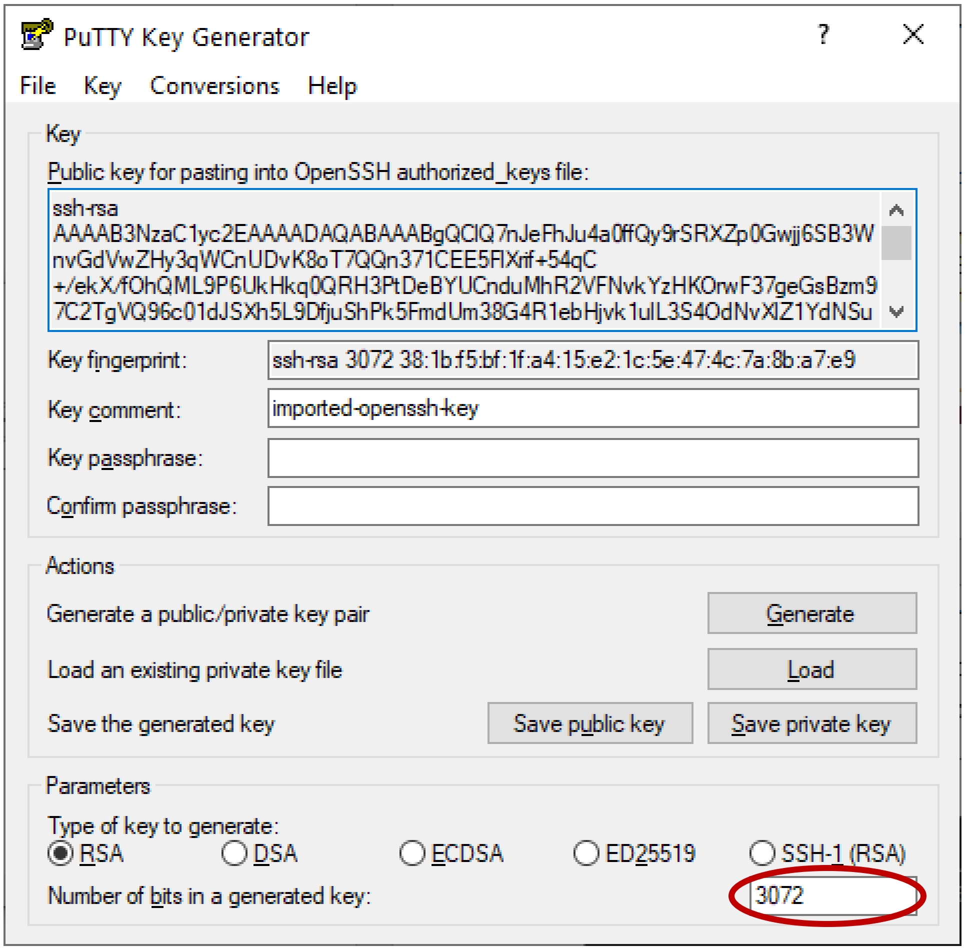
Save the generated key

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

If the .pem file is successfully imported, you will see something similar to the next image.

[2] Public-private key pairs can be generated in various ways. You may follow that link to see how you can generate key pairs from the (unix) command line.



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQClQ7nJeFhJu4a0ffQy9rSRXZp0Gwjj6SB3W
nvGdVwZHy3qWCnUDvK8oT7QQn371CEE5FlXrif+54qC
+/ekX/fOhQML9P6UkHkq0QRH3PtDeBYUCnduMhR2VFNvkYzHKOrwF37geGsBzm9
7C2TgVQ96c01dJSXh5L9DfjuShPk5FmdUm38G4R1ebHjvk1ulL3S4OdNvXlZ1YdNSu
```

Key fingerprint: ssh-rsa 3072 38:1b:f5:bf:1f:a4:15:e2:1c:5e:47:4c:7a:8b:a7:e9

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Parameters

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

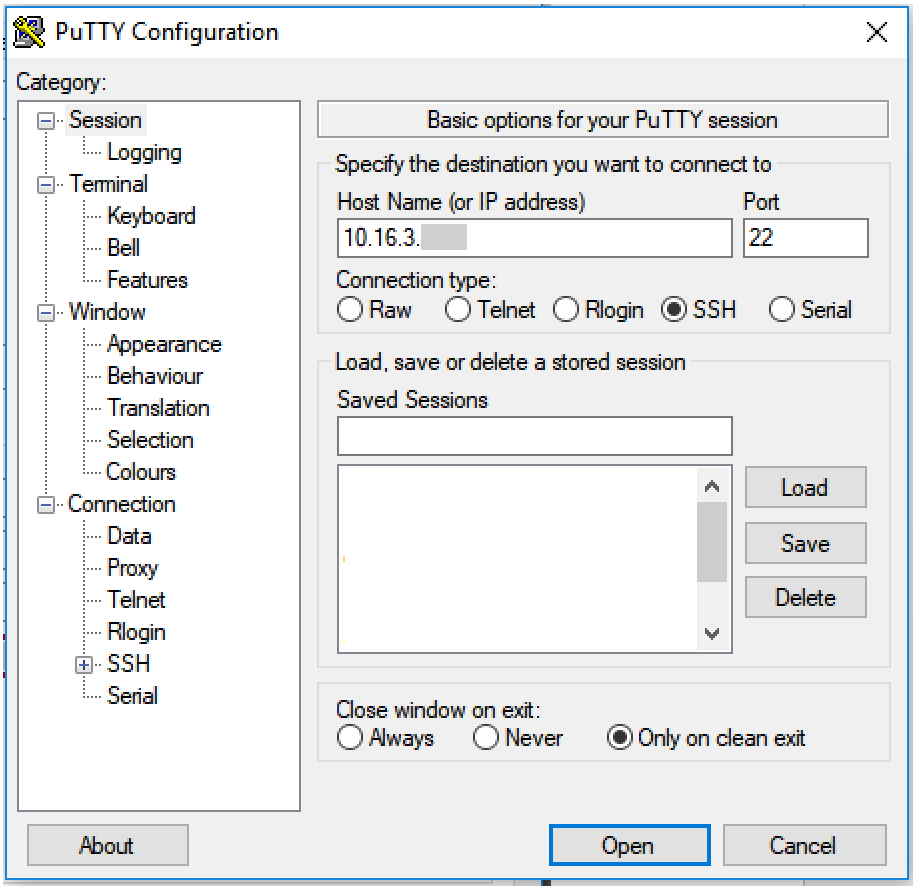
Number of bits in a generated key: 3072

At this stage, you may provide a passphrase (twice, for confirmation). This passphrase will be used to encrypt the key on disk, so you will not be able to use the key without first entering the passphrase. More specifically, the password will be required during the login process as the client tools (described in this guide) make use of the private key stored in the .ppk file to decrypt data sent from the VM.

If you leave the passphrase fields blank, the key will be saved unencrypted. You should not do this without good reason; if you do, your private key file on disk will be all an attacker needs to gain access to any machine configured to accept that key.

After passphrase is entered (twice), click the Save private key button and save the .ppk extension file by giving name to it.

In order to connect to VM using PuTTY, firstly, you enter the VM IP address into the Host Name (or IP address) field as shown in the following image. You can save the session by entering a name in the Saved Sessions box and then clicking the Save button.



Specify the destination you want to connect to

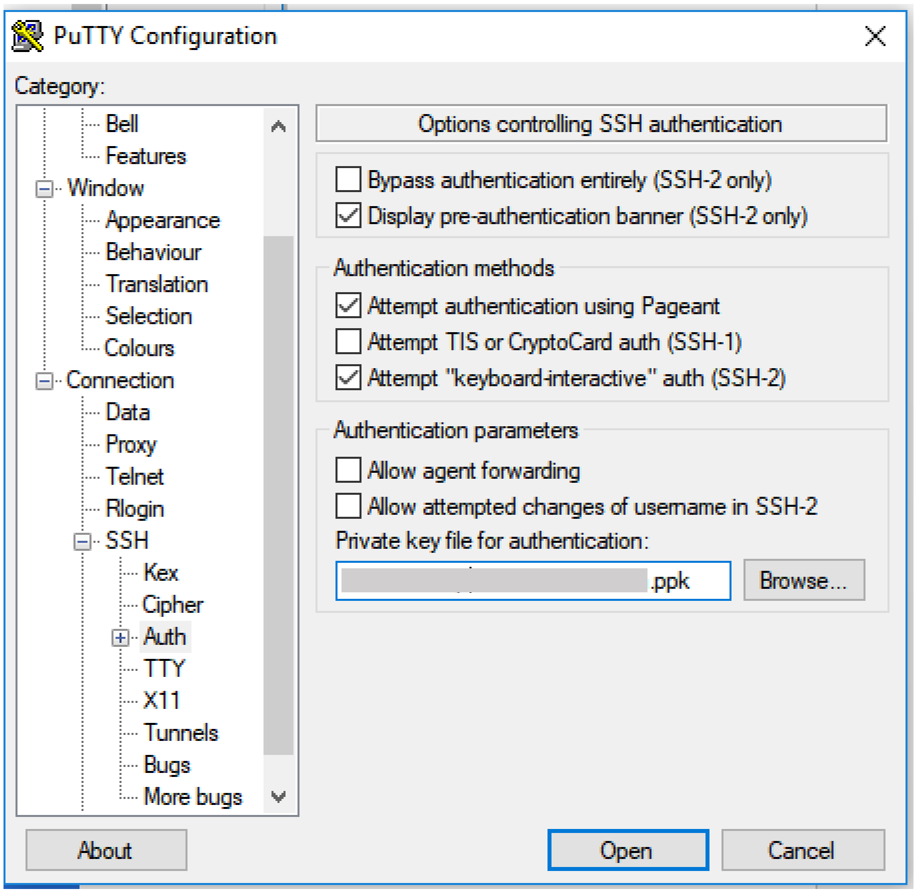
Host Name (or IP address): 10.16.3.[redacted]

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

Then you have to import the .ppk file as shown below:



□ Bypass authentication entirely (SSH-2 only)

☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

□ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

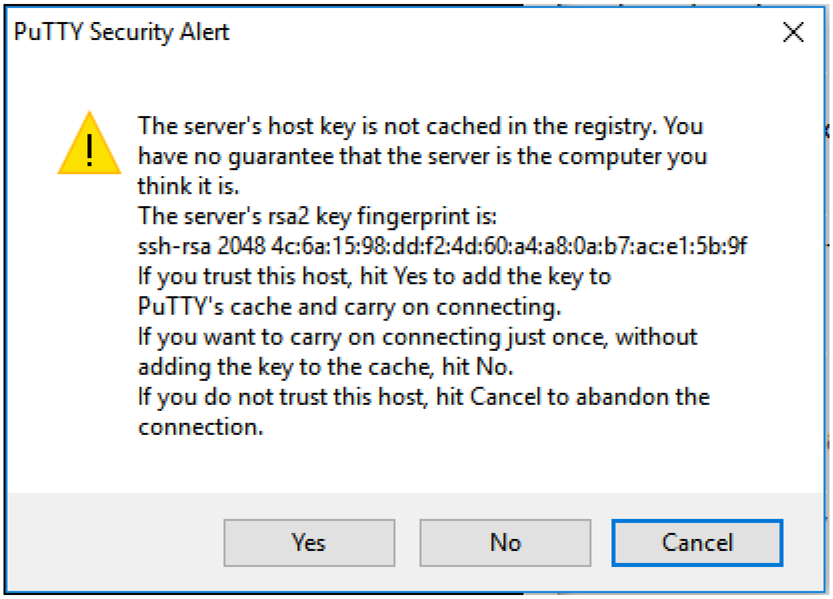
□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication: [redacted].ppk

You can save the .ppk file as part of the session (otherwise you will have to import the key anytime you try to connect to the VM via PuTTY) by going back to the Session screen and clicking Save. You can connect to the VM by clicking the Open button.

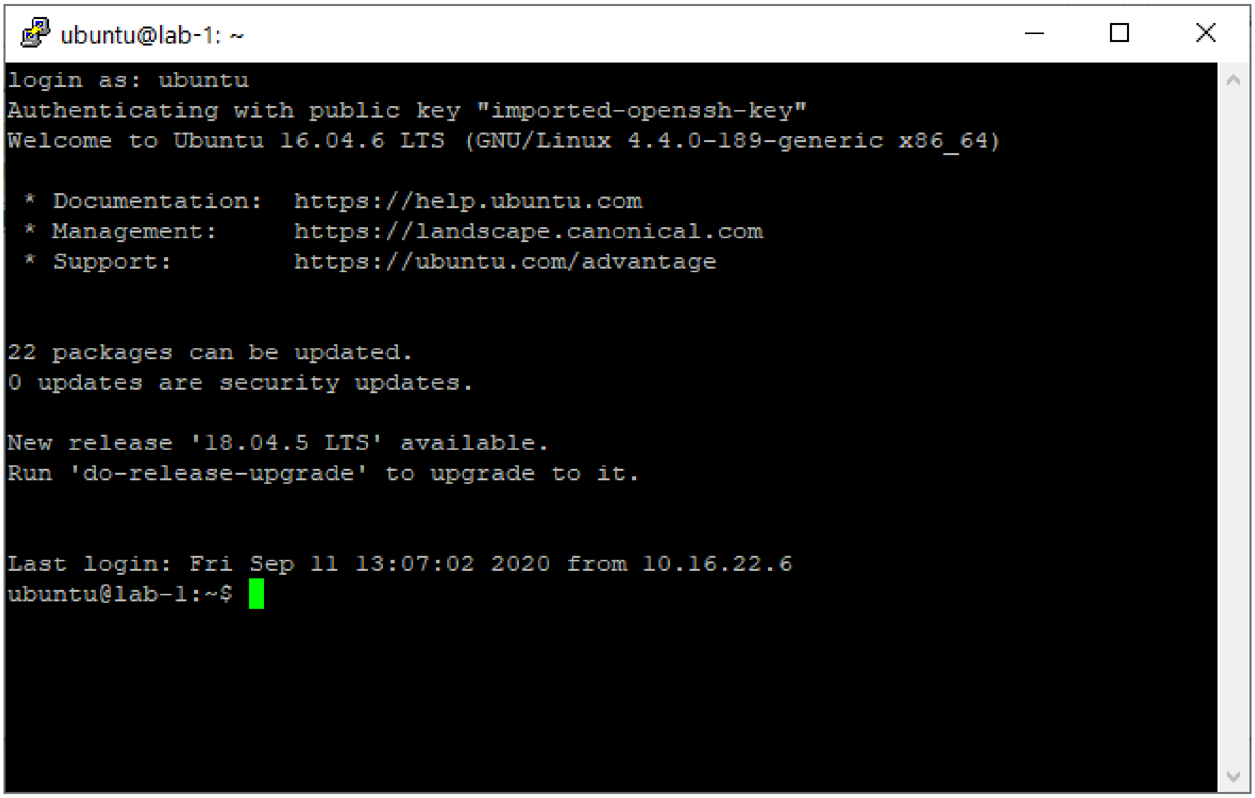
The first time you attempt to connect to the VM, the SSH server (on the VM) will sent its own public key (as discussed in the end of Section I) as shown below, and you will have to accept it. Otherwise, the secure connection will not be established and you will not be able to gain access to the VM.



PuTTY Security Alert

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 4c:6a:15:98:dd:f2:4d:60:a4:a8:0a:b7:ac:e1:5b:9f
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

When you accept the server's public key, you will be prompted for the username which is ubuntu and then you have to provide the passphrase that was used to encrypt the .ppk on disk.

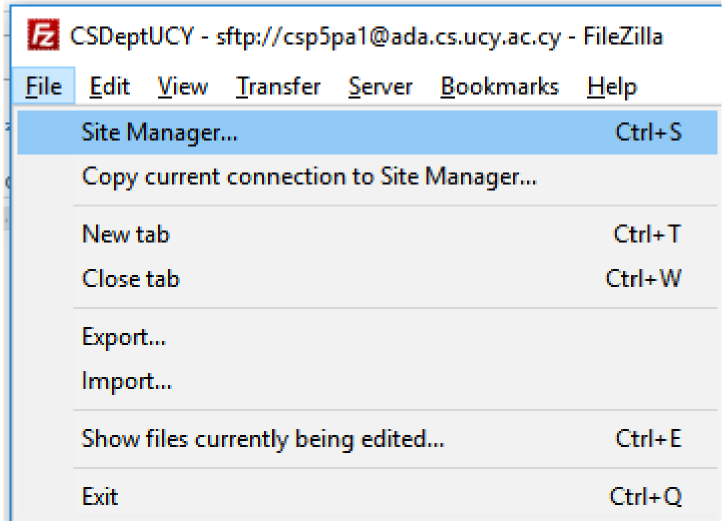


```
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-189-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


22 packages can be updated.
0 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Sep 11 13:07:02 2020 from 10.16.22.6
ubuntu@lab-1:~$
```

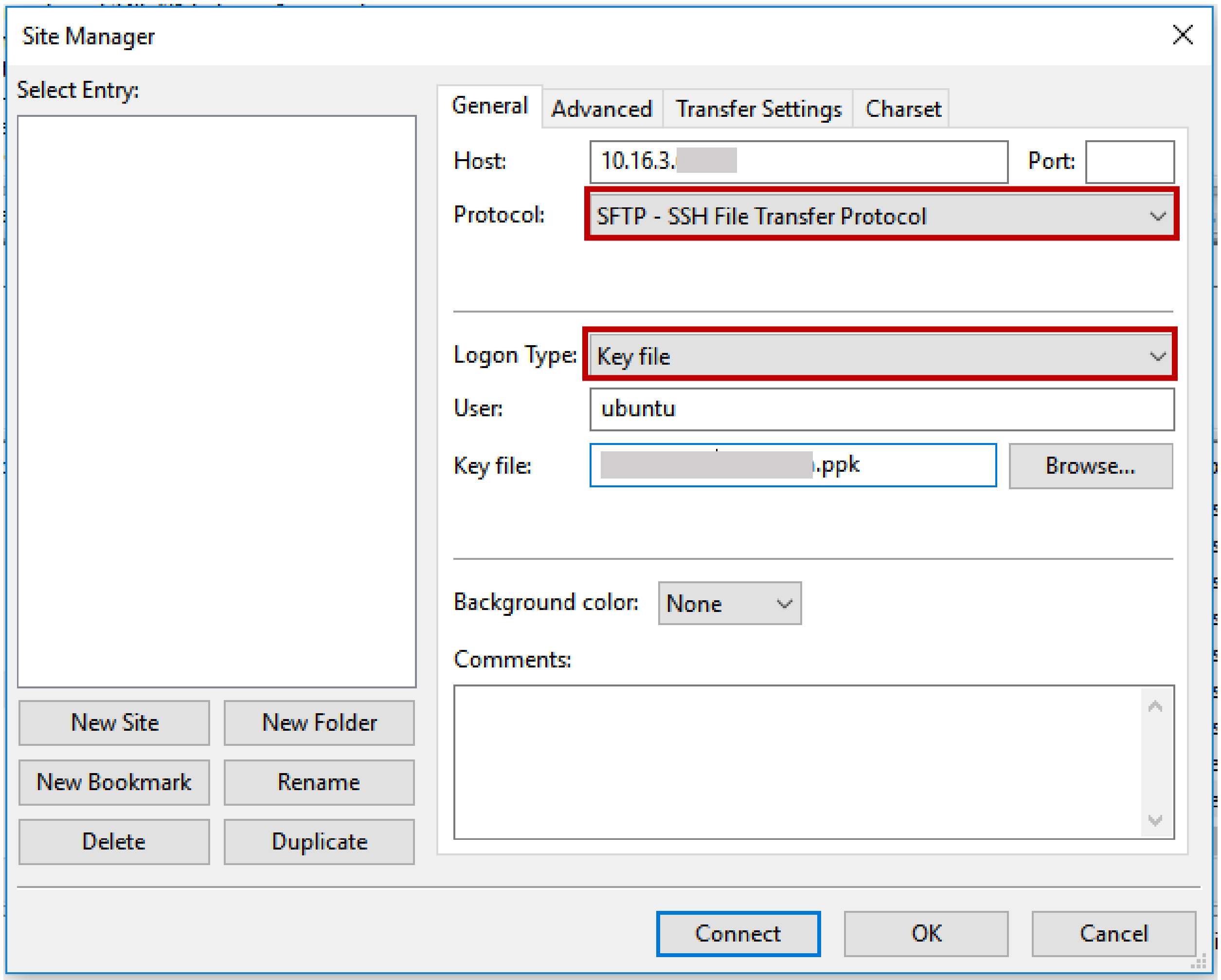
Upon successful completion of the process, you will manage to login to the VM via command line.

FileZilla

After installing FileZilla, you have to create a new site from File → Site Manager … as shown below.

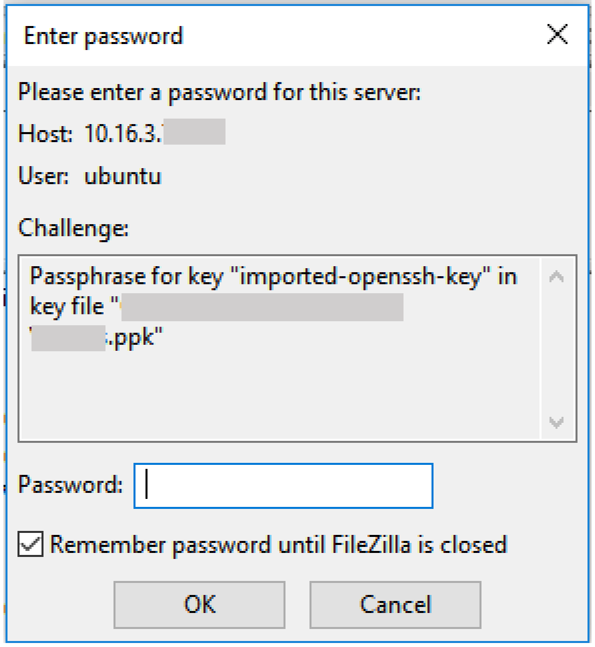

CSDeptUCY - sftp://csp5pa1@ada.cs.ucy.ac.cy - FileZilla

You need to go through the following: (a) fill in the VM IP address into the Host field, (b) choose the SFTP protocol, (c) choose Key file as Logon Type, (d) enter Ubuntu into the User field, and (e) import the Key file (.ppk).


Site Manager

Host: 10.16.3.[redacted]
Port:
Protocol: SFTP - SSH File Transfer Protocol
Logon Type: Key file
User: ubuntu
Key file: [redacted].ppk
Background color: None
Comments:

When you attempt to connect, you have to enter the passphrase as shown below. You may choose to make FileZilla remember the passphrase for any subsequent connection.



Enter password

Please enter a password for this server:

Host: 10.16.3.[redacted]

User: ubuntu

Challenge:

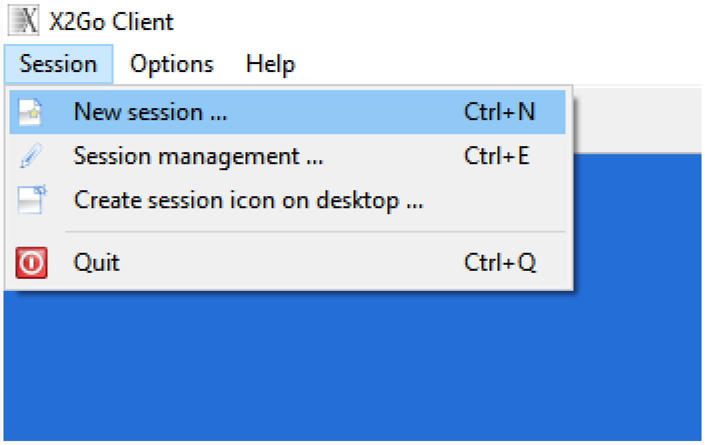
Passphrase for key "imported-openssh-key" in key file "[redacted] [redacted].ppk"

Password:

☑ Remember password until FileZilla is closed

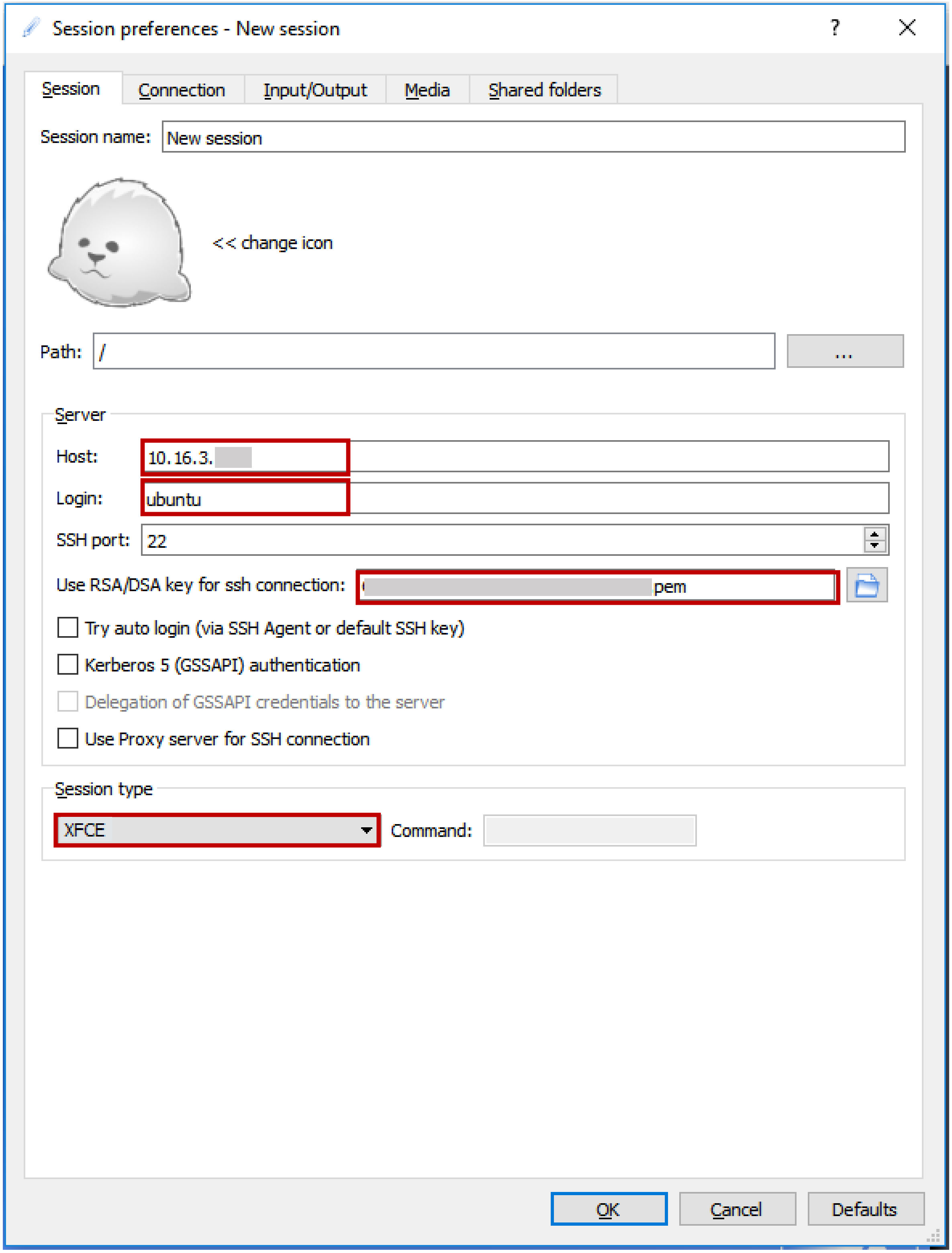
X2Go client

After installing X2Go client you have to create a new session as shown below.



X2Go Client
Session
Options
Help
New session ... Ctrl+N
Session management ... Ctrl+E
Create session icon on desktop ...
Quit Ctrl+Q

In the new windows that appears in your screen, go through the following: (a) type the IP address of your VM in the Host field, (b) type the username ubuntu in the Login field, (c) import the given PEM file (X2Go client can use the PEM file) and (d) choose MATE in session type.



Session name: New session

<< change icon

Path: /

Server

Host: 10.16.3.[redacted]

Login: ubuntu

SSH port: 22

Use RSA/DSA key for ssh connection: [redacted]pem

☐ Try auto login (via SSH Agent or default SSH key)

☐ Kerberos 5 (GSSAPI) authentication

☐ Delegation of GSSAPI credentials to the server

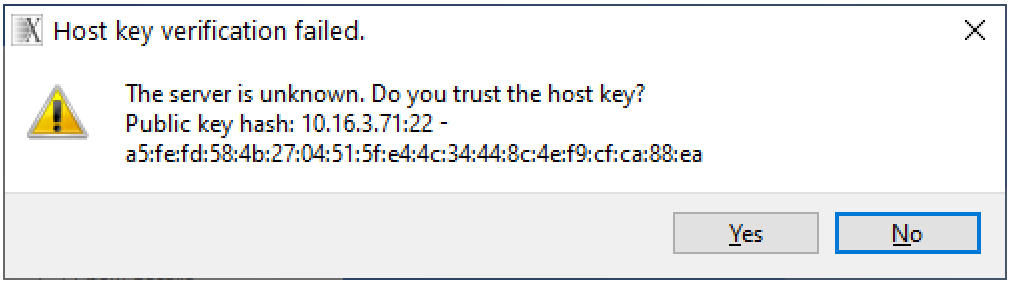
☐ Use Proxy server for SSH connection

Session type

XFCE

Command:

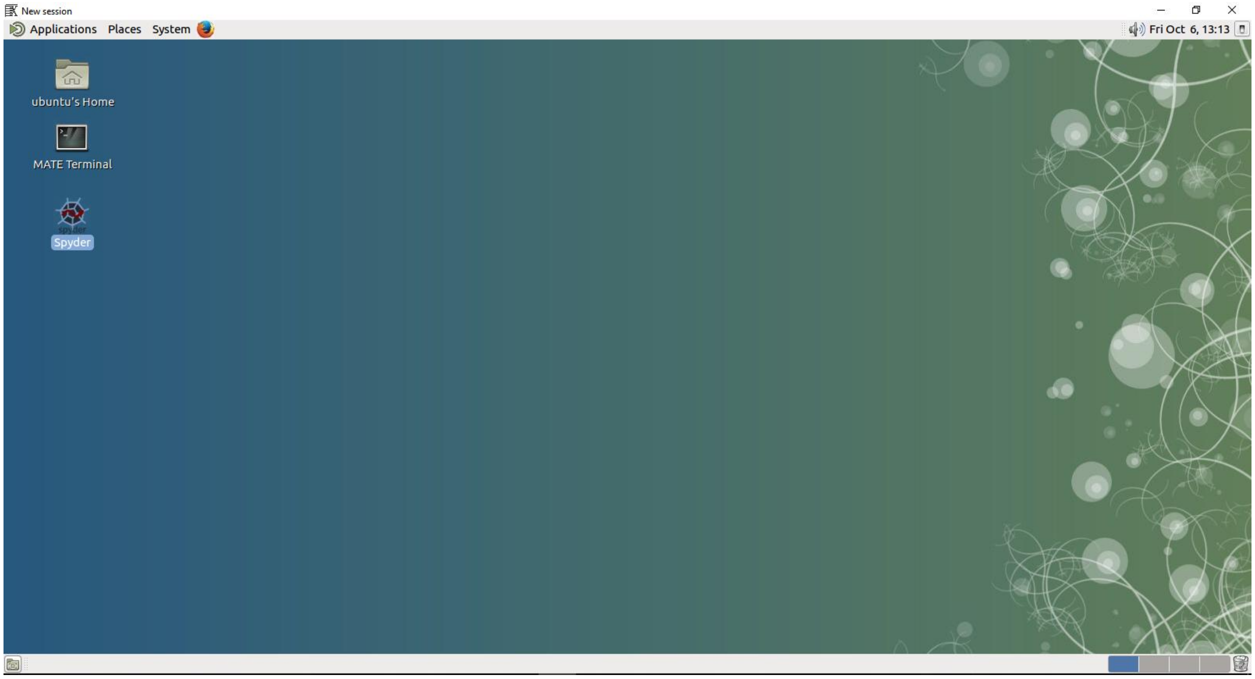
The first time you attempt to connect to the VM, the SSH server (on the VM) will sent its own public key (as discussed in the end of Section I) as shown below, and you will have to accept it. Otherwise, the secure connection will not be established and you will not be able to gain access to the VM using the graphical user interface provided by X2Go.



Host key verification failed.

The server is unknown. Do you trust the host key?
Public key hash: 10.16.3.71:22 -
a5:fe:fd:58:4b:27:04:51:5f:e4:4c:34:44:8c:4e:f9:cf:ca:88:ea

Upon successful completion of the process, you will see the following (MATE) desktop environment:



IMPORTANT NOTE: The VMs are only visible from the internal network of the Computer Science Department. Therefore, if you want to access them while being outside the CS network, you need to connect to CS VPN in advance.