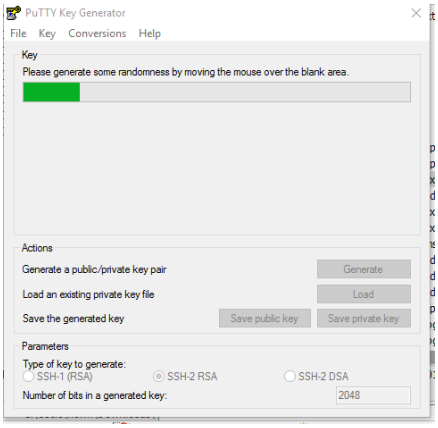
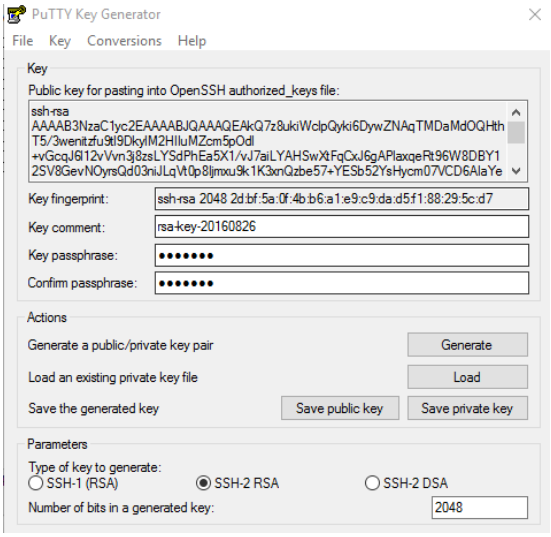
This explains how to create a key pair for connecting to the raspberrypi via ssh. This example is for a windows pc using **PuTTY** a remote terminal program. This example also uses **PuTTYgen** to generate the public and private keys. The download page for both of these tools can be found at this link: http://www.putty.org/

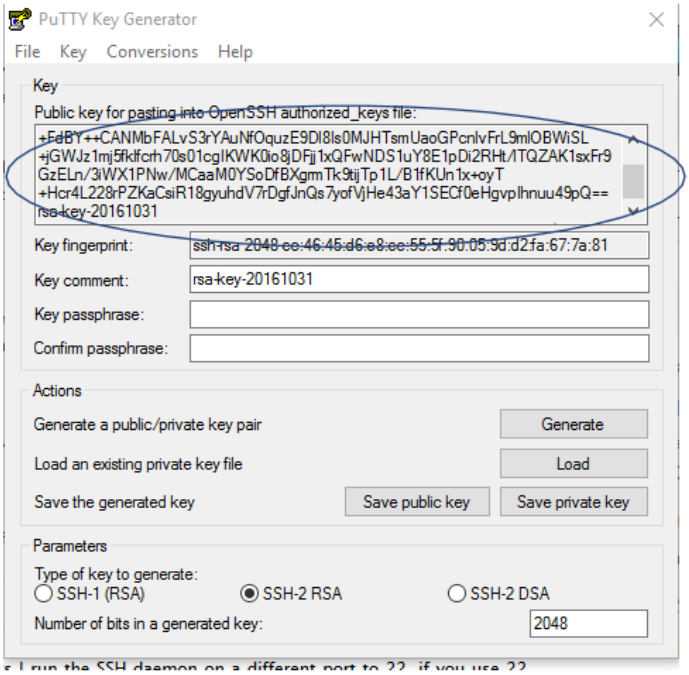
The first step with **PuTTYgen** is to generate a key pair. The program uses random movements of your mouse to generate the key.



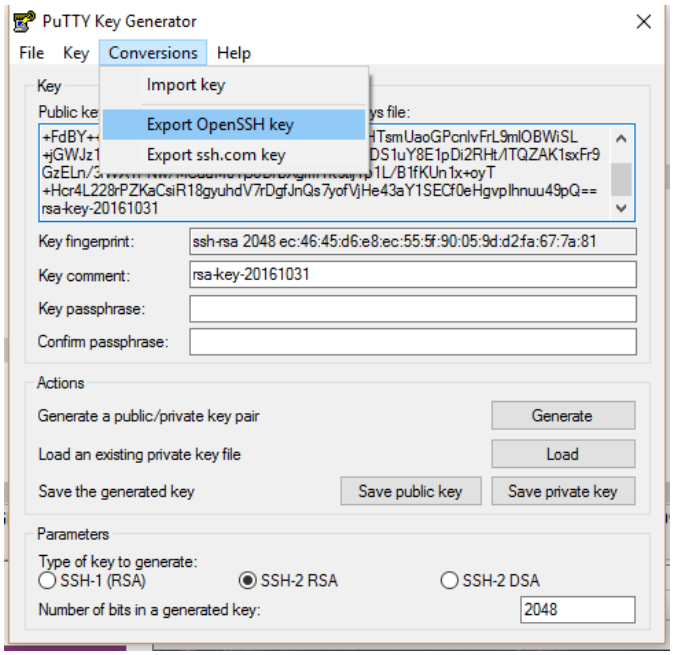
Once complete the key is displayed as well as other information including a comment that can be added to the private key file. At this point you can also enter in a passphrase that will be needed to log in as an additional security feature. If you do not enter anything here, no passphrase will be required. Press the buttons to save the public and private keys. Only the private key is needed for Windows. It is important that you keep the private key secure as it enables access to the pi for anyone who has the key. How the private key on Windows is used by **PuTTY** is described in the next section.



To generate keys for the pi and mac. The public key for the pi is in the window in puttygen that you can cut/paste into a text file. This file can be copied to the .ssh directory in the pi home account. The example script is included in this document and on the github account.

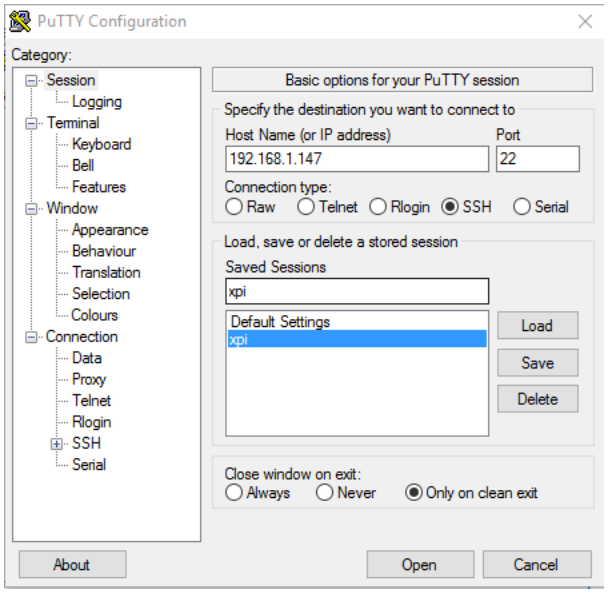


To generate the private key for the MAC, use the Conversion menu and export an open ssh key. Put this key file on the MAC in a location that can be accessed when you use ssh to login to the pi.

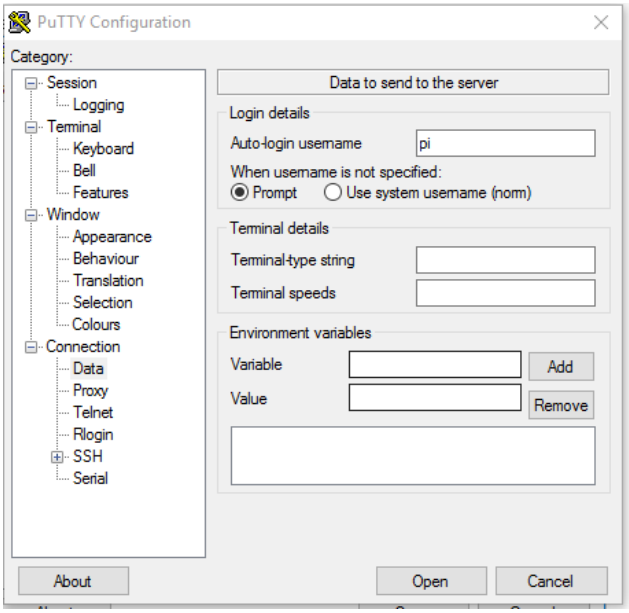


These are the steps to use the private key on windows in the **PuTTY** tool.

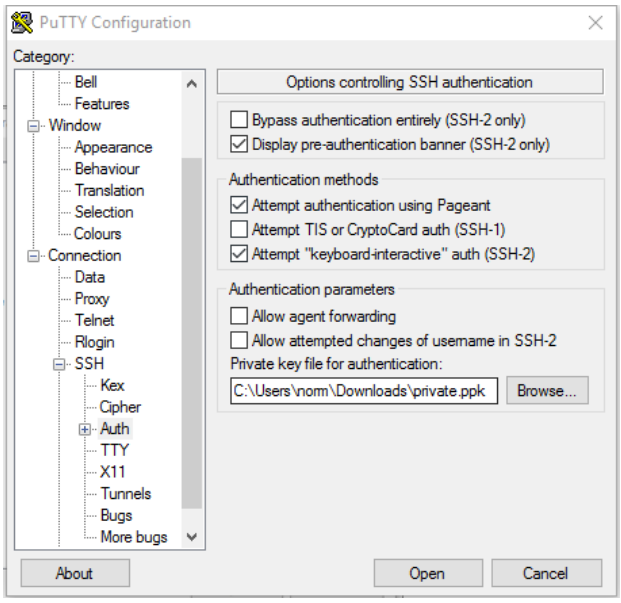
Step 1: Enter the "Session" dialog, enter in the IP address of the pi, port 22, SSH, only on clean exit. You can save the information in a session for reuse (this one is called xpi).



Step 2: under Connection->Data enter the user name (pi) for auto login.



Step 3: select Connection->SSH->Auth to bring up the authentication dialog. Match the checked and unchecked boxes. Then Browse for the private key file you created with **PuTTYgen**. Select Open and you should see a window. IF you entered in a passphrase, this will be required at this time, otherwise if the keys match, you will be logged in without any password or passphrase. IF you enter in the correct passphrase, you will be logged in without a password.



This section describes how to install the public key on the pi. The following is the bash script used to install the key. This is also on the github account. You also need the sshd_config file on the github account.

```
#!/bin/bash
# this script installs the public key for the user to allow secure ssh
# the script can also be run to install a new key
# /etc/ssh/sshd_config is also updated to restrict logins
#
D=     # set to echo for debug w/o executing the script
G=https://raw.githubusercontent.com/norm42/SecureSSH/master/  #location of public key and config
file
USR=pi  # this should be set to the user name
#get files from github
$D curl -o authorized_keys
https://raw.githubusercontent.com/norm42/SecureSSH/master/authorized_keysOCT
$D curl -o sshd_config https://raw.githubusercontent.com/norm42/SecureSSH/master/sshd_config

USRHOME=/home/$USR
$D sudo cp sshd_config /etc/ssh      # update sshd config file

if  [ -d "$USRHOME/.ssh" ]
then
   $D sudo rm -rf $USRHOME/.ssh              # remove if exists to clear out any history
fi

# make the .ssh dir, copy the public key to .ssh, set owner/group
# set modes to protect the files (600,700) from being overwritten by malicious user.
#
$D sudo mkdir $USRHOME/.ssh
$D sudo cp authorized_keys $USRHOME/.ssh
$D sudo chown $USR $USRHOME/.ssh $USRHOME/.ssh/authorized_keys
$D sudo chgrp $USR $USRHOME/.ssh $USRHOME/.ssh/authorized_keys
$D sudo chmod 600 $USRHOME/.ssh/authorized_keys
$D sudo chmod 700 $USRHOME/.ssh
```

On the MAC, to access the pi with the private key that is located on the MAC, use this command:

```
ssh  -i private_key_filename   username@hostname_or_ip_address
```