Implementing SSH RSA Key Authentication

RSA Cryptosystem Background Information

Read the following article for more information: https://en.wikipedia.org/wiki/RSA_(cryptosystem)

The very high-level view is to generate a very large prime number. This number is called a "private key" and is to be kept secret. Then generate other prime number(s) that has mathematical relationship to the private key (via the RSA algorithms), called "public keys(s)". Anything encrypted using a public key can only be decrypted using the private key. Private key can also be used to generate "signatures" to verify the sender's identity.

In short, only the party that has access to the private key can possibly decrypt anything that is encrypted using the associated public key (and to generate a signature to verify its identity). Therefore, **keep your private keys private and never ever share them with anyone for any reason**.

Generating SSH RSA key pair

Simply use the command "`ssh-keygen`" and follow the prompts. Although optional, it is recommended that you use a passphrase when creating your key pair.

Note: It does not matter who/which executes this command, or on what computer the command is executed on. As long as you get the private and public keys you can use this anywhere. You can specify a different location if you are going to be using this key on another machine.

Note: many SSH client software such as Bitvise and PuTTY come with software or facilities to generate RSA keys on your computer. You do not have to do this on the server!

>>>>Take Screenshot 1.png<<<<

Example:

```
ali@ers20095559:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ali/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ali/.ssh/id_rsa
Your public key has been saved in /home/ali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:tBE4InireUklF/Zta7oYSxcdt+5/7qh1oXHhd3r7s6E ali@ers20095559
The key's randomart image is:
+---[RSA 3072]----+
| .  o. ..        |
|. +.+.o. .       |
| . * ...* .   .  |
|  o    + * . . . |
| + .  . S . . + o|
|o o    + .   + +.|
| .  o o   . o o..|
|   . = . . . oooo|
|    o .   oooEoo=|
+----[SHA256]-----+
```

The following files are generated in the specified location:

```
ali@ers20095559:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts
```

Here is the list of files you see:

- id_rsa is your **"Private Key"**
- id_rsa.pub is your **"Public key"**

Question 1: what are the default permissions for your private key? Answer using the following format:

- File mod (either numeric, or alphabetic format, ex 755 or -rwxr-xr-x), File owner, and File group

Note: if you are using a cloud server, or any internet connected server, use 600 (rw-------) as the file permission mode

Question 2: what will happen if you create a new user, generate a /home/user/.ssh/authorized_keys file, and then do a `sudo passwd -l user`? i.e. will the user be able to login to the system and how?

NOTE: do not run passwd -l command against your own username unless you have created an alternate account with sudoers group membership that can undo this. Otherwise you will have to recover your root account!

Using RSA key for SSH authentication

Authorize your Public Key on the server

In order to authenticate to a server via RSA, you need to provide your public key to the server and authorize that public key to login to shell via SSH. In case of cloud computing (aws, azure, etc) or if you are not the administrator of the server you are accessing, you can provide your **public key** to the server administrator (or via the web-interface provided by aws when launching the instance) and the administrator can put your key in a file called ~/.ssh/authorized_keys.

Remember: private key is PRIVATE, never give this out to anyone! Not to AWS, Azure, the hosting company, or whoever the administrator of your server is.

Since you are the administrator of your own server, go ahead and add your public key into the .ssh/authorized_keys file:

```
ali@ers20095559:~$ cp .ssh/id_rsa.pub authorized_keys
```

Now use any method to export your private key from the server. I used sftp to copy id_rsa file to my desktop.

Note: You should safeguard your private key. I suggest you delete your private key from the server once you have it in your possession elsewhere. Don't leave your private key lying around on the server!

Run the following commands in your console:

```
cd ~/.ssh
ls
cat authorized_keys
```

>>>Take Screensot2.png<<<

Example:

```
ali@ers20095559:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
ali@ers20095559:~/.ssh$ cat authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDlRKOPQo+VdJ4YtLLOE1iqlmSdoOjGH/qDh2689vMyHptHuWY6+N7
DevrAGtjqYZ5n5g6zpIDQqXAwcbL66pHMLmOetouqdO9DDvxTm1Mt04EmYGPeZ9Teo++b72cNISkKFATYtO
EAgGjQBycGl7MspWL41rHQRL7FbJkOlEHyMicSIBLYiHoqLn1S+H/lofaI96ERE/qw2jEQavtnvpUk4f83a
2tSRB4ZwFACz6GpyXP4moppAAVpYpLCmUWh+DyCg0Q3cKrfyjNjZVMUJBT+1PWNGrnlzFERN+tH3VW3CfnT
11rjzIeC+tWP4YbiuRgjVQT3SpT8EKANLU3i7ggWglyZwZaRjHtLfiRSCq7lonZQWGUDgZ9Pmm1ZkcOfShP
fjeV1qkRxxHojTU1Qo3ibdkjxM20Yjvv8Nrkv52wL1+XAAyP6xJbrmoDZwthjOPr2aEXfC69M/9cHOBU3zv
QlskY51psc8EaasDVqbuZCPChpE1HsYvjxMdtBxkCCcVU= ali@ers20095559
```

Configure your client with your private key

This step may vary depending on what SSH client you are using. Every client has a different way of implementing this. Read your SSH client documentation or refer to the Appendix at the end of this document.

Assignment Submission

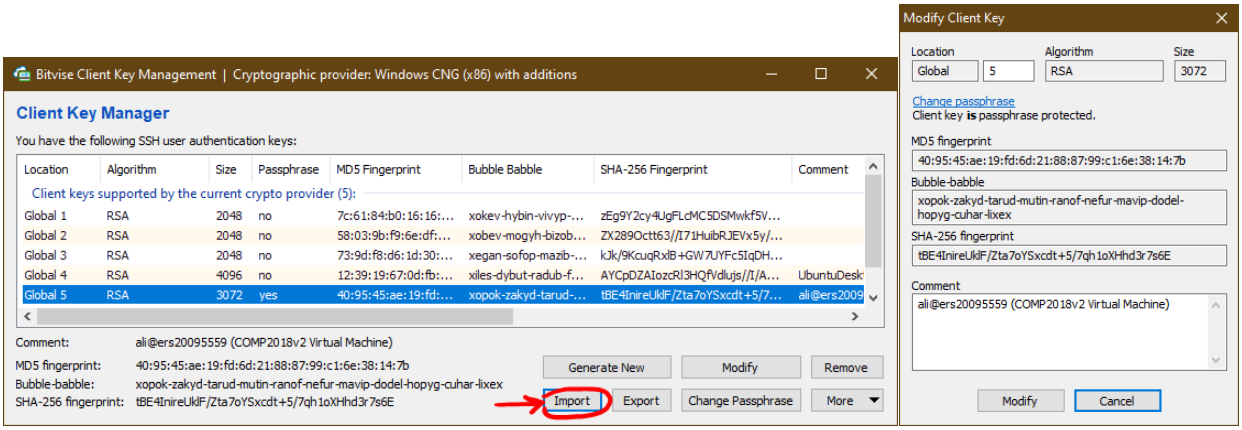
1. Answer questions 1 and 2 in the text submissions.
2. Attach the screenshot1.png from your RSA key generation step in the attachments.
3. Attach the screenshot2.png

Appendix

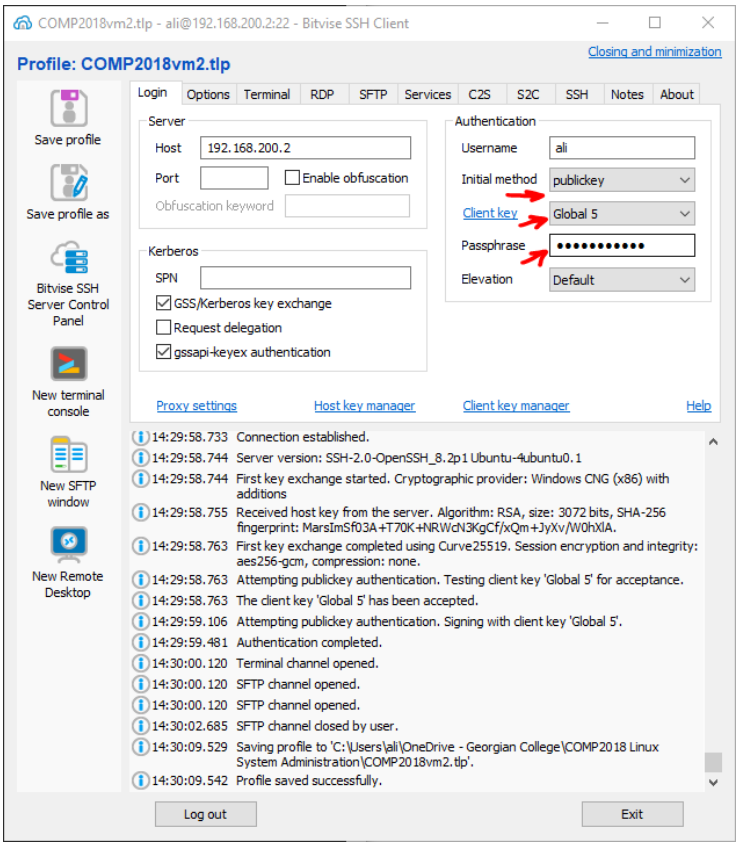
Bitvise SSH Client in Windows

open Start > Bitvise SSH Client > Client Key Manager

Click on “Import”, navigate to your file (you may want to select “All Files *.*” in the file type to see your id_rsa file). Follow on screen instructions.

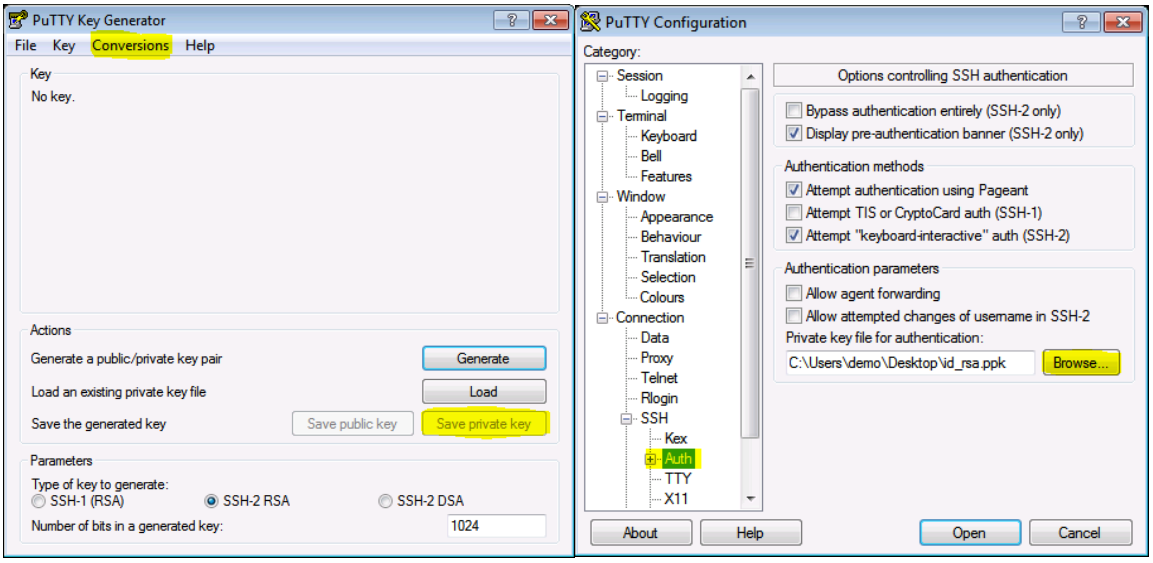


Modify your SSH profile or create a new SSH profile in Bitvise. Under Authentication enter your username and select the initial method “publickey” from the drop-down menu. Then select the client key to be the one you imported earlier. Here is what mine looks like after the modifications:



PuTTY in Windows

From start menu under PuTTY open “PuTTY Key Generator”. Click “Conversions” and then “Import Key” and navigate to your private key (“id_rsa” file). Click on “Save Private Key” button and save the file as “id_rsa.ppk”.



To connect using the key, scroll down to SSH > Auth under Category. Click on the Browse button and select the id_rsa.ppk file from earlier. Now you should be able to click on Open to connect to server using your key.

MacOS or other terminal using ssh command

You only have to put your private key in any location and use the following syntax to open a SSH connection:

ssh -i </path/to/private_key> <username>@<ip_address>

Example: ssh -i id_rsa ali@192.168.200.2