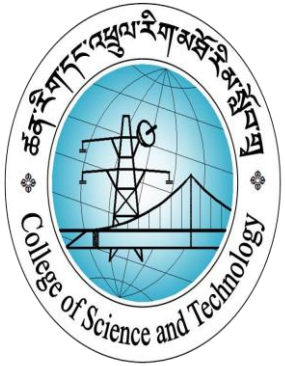


Royal University of Bhutan

LESSON – 17

SSH and Keys

LEARNING OUTCOMES

On Completion of this lesson, you will be able to:

- Use SSH and Keys based authentication
- Generate Asymmetric Keys
- Secure private key with passphrase
- Work with true password less authentication
- Restrict the access using password and root user

SSH Introduction

Password is Bad, Really Bad!:

1) Password is bad! A large proportion of security failures are due to passwords:
   - Users choose poor passwords
   - Users write them down or share them
   - Passwords can be guessed or brute-forced
   - Password can be sniffed or key-logged
   - People hate forced password changes and password complexity tests

SSH Introduction

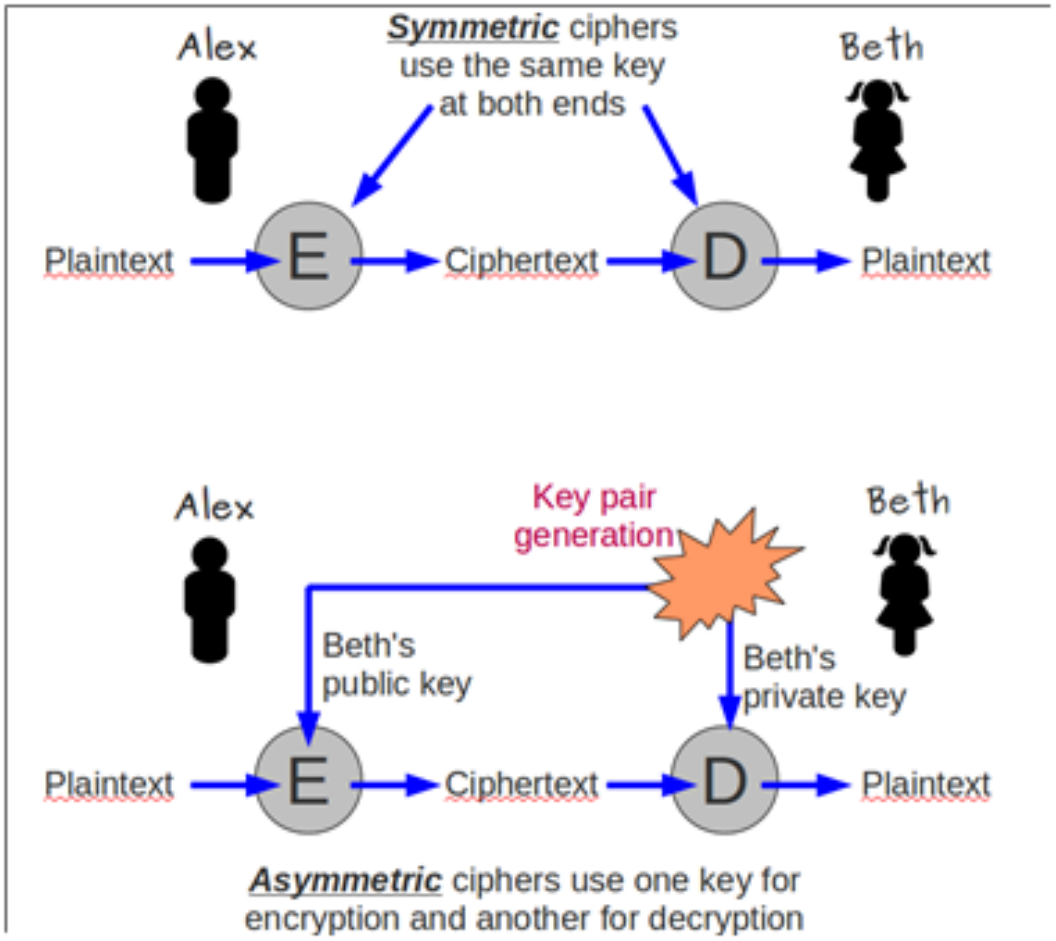
SSH and System Administration

1. SSH gives you remote command-line access to systems
2. Traffic is encrypted, which at least makes it hard to sniff passwords off the network
3. telnet uses plain text
4. Much better than telnet
5. SSH allows you to use cryptographic keys instead of password

SSH and Keys

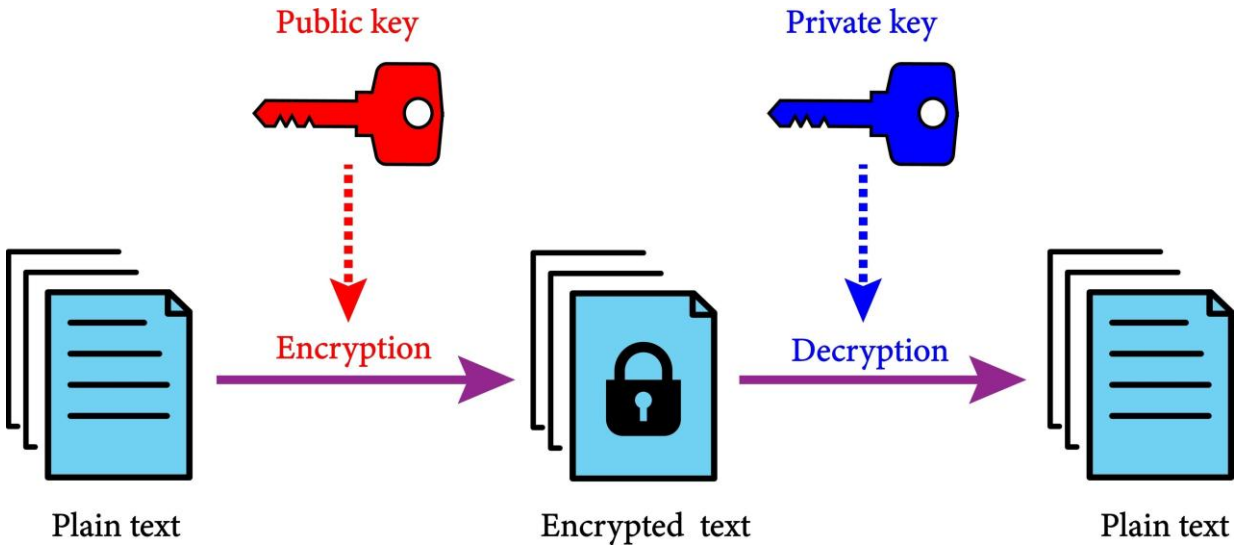
Cryptographic Keys:

1. Symmetric Key
2. Asymmetric keys



SSH and Keys

How do SSH and Keys Works?



SSH and Keys

- The public key is used to encrypt data that can only be decrypted with the private key. The public key can be freely shared, because, although it can encrypt for the private key, there is no method of deriving the private key from the public key.

How it works?

Authentication using SSH key pairs begins after the symmetric encryption has been established as described in the previous section.

The procedure happens as follows:

- The client begins by sending an ID for the key pair it would like to authenticate with to the server.
- The server checks the authorized_keys file of the account that the client is attempting to log into for the key ID.
- If a public key with a matching ID is found in the file, the server generates a random number and uses the public key to encrypt the number.
- The server sends the client this encrypted message.

How it works?

- If the client actually has the associated private key, it will be able to decrypt the message using that key, revealing the original number.
- The client combines the decrypted number with the shared session key that is being used to encrypt the communication, and calculates the *MD5 hash* of this value. MD5 is a message-digest algorithm that uses the hash function to generate a 128-bit hash value.
- The client then sends this MD5 hash back to the server as an answer to the encrypted number message.
- The server uses the same shared session key and the original number that it sent to the client to calculate the MD5 value on its own. It compares its own calculation to the one that the client sent back. If these two values match, it proves that the client was in possession of the private key and the client is authenticated.

SSH and Keys

Using crypto keys with SSH:

- This is a one-time operation
- For Windows/putty: use **puttygen.exe**
- For Linux and OSX: use **ssh-keygen**
- There are four different key types
  - rsa (v1 obsolete), rsa (v2), dsa, ecdsa
- Recommended to use RSA version 2 with a key length of 2048 bits (-t rsa -b 2048)
- You get a private key and a related public key

SSH and Keys

OpenSSH public key looks like this :

- One very long line of text

ssh-rsa AAAAB3NzaC1..... you@yourmachine

- Safe for copy-paste (but beware line wrap)
- puttygen has a different native format but can also export the above format

SSH and Keys

Understand the difference!:

- Your **private key** is like the Crown Jewels
- Your **public key** is like a photograph of the Crown Jewels
- Which of these would you be happy to send via the postal service? :-)
- Never give your private key to anyone else Never send your private key via E-mail
- Should you need to transfer it, do so via a secure channel like scp or sftp

SSH and Keys

Activity:

Using crypto keys with SSH:

- Generate a Private/Public key pair
- Copy the public key onto the systems you want to be able to log into
- Log in with ssh, using your private key to prove your identity to the other system, instead of a password

SSH and Keys

Linux and SSH:

1. `ssh-keygen`
2. `ssh-keygen`

[OR]

`ssh-keygen -t rsa|dsa -b 4096`

1. `ssh-copy-id`
2. `eval $(ssh-agent)`

   [or] `ssh-agent`

1. `ssh-add`

Example:

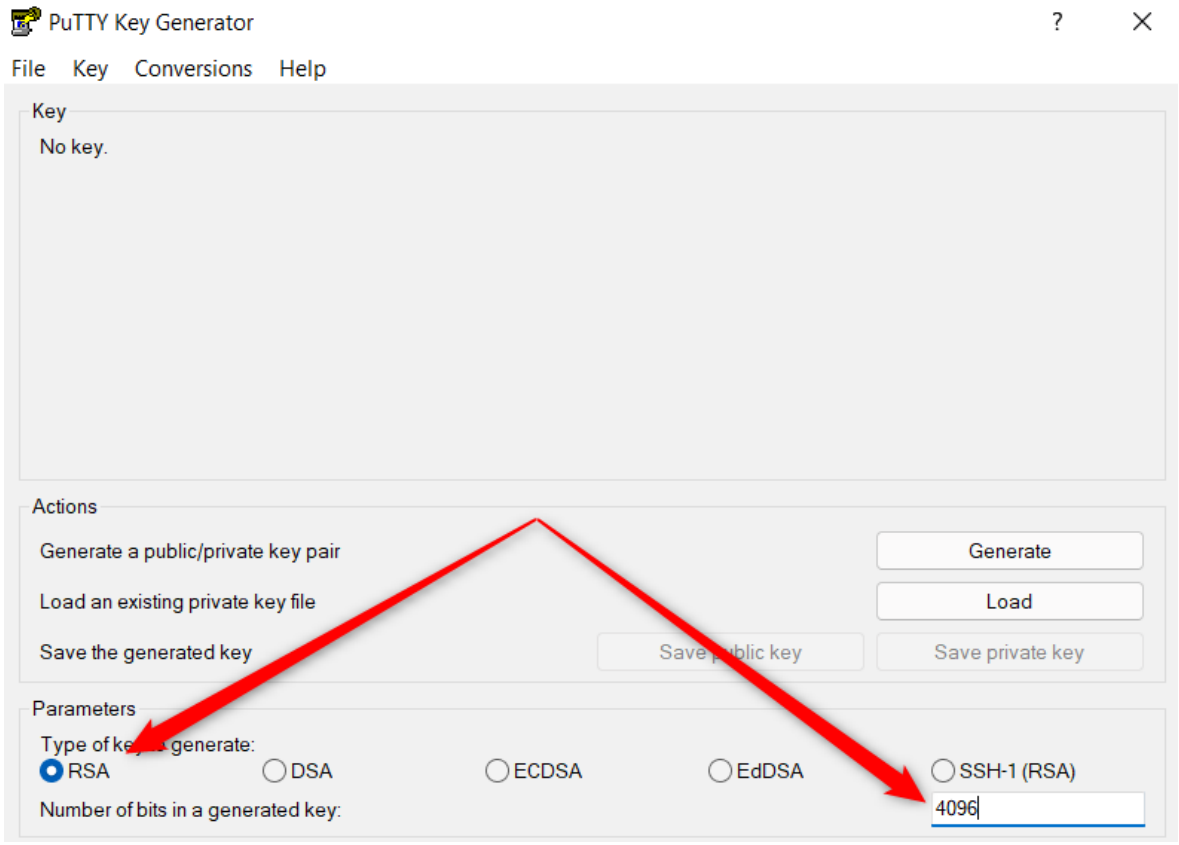
```
#ssh-keygen | ssh-keygen -t rsa -b 4096
    ~/.ssh/id_rsa
    ~/.ssh/id_rsa.pub
#ssh-keygen -f .ssh/key-with-pass
    ~/.ssh/key-with-pass
    ~/.ssh/key-with-pass.pub
#eval $(ssh-agent)
or #ssh-agent bash
#ssh-add

#scp .ssh/id_rsa.pub
sysname@IP:.ssh/authorized_keys
```

SSH and Keys

Activity (Windows):

Generate Keys With PuTTY Key Generator:

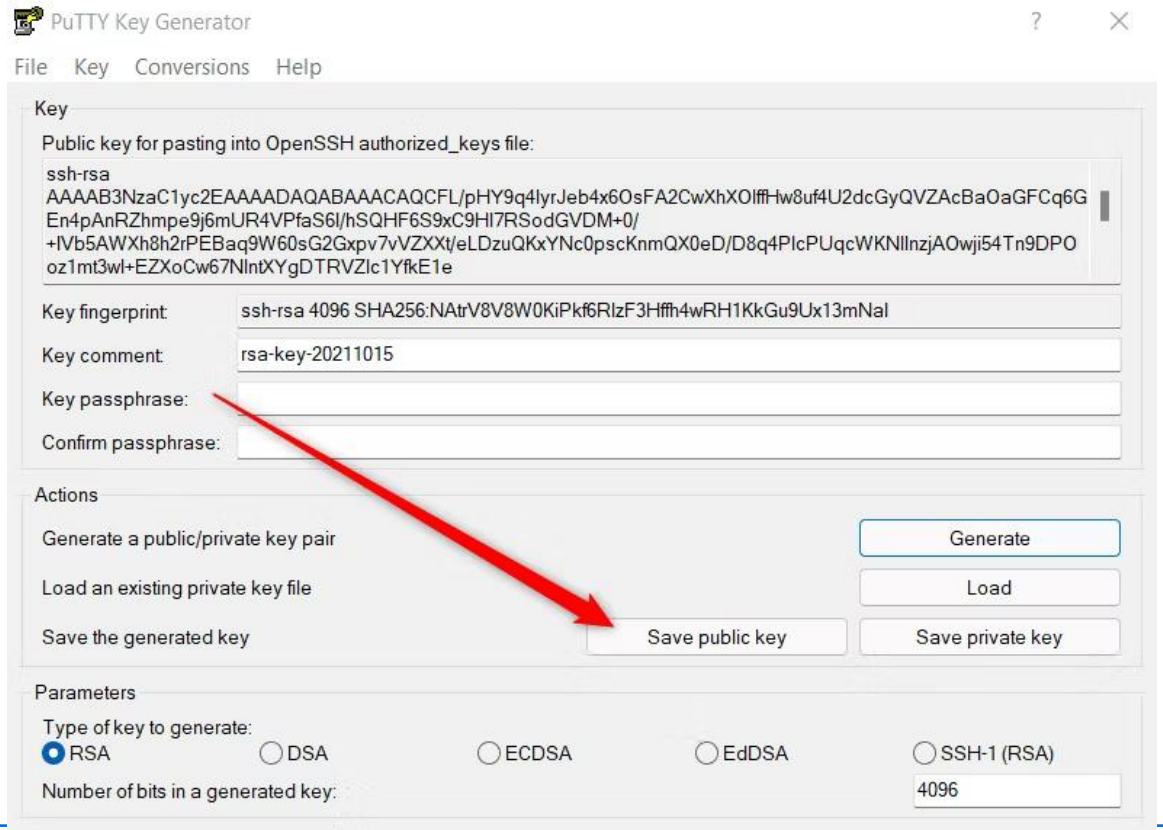


Update the parameters and click "**Generate**"

SSH and Keys

Activity (Windows):

Generate Keys with PuTTY Key Generator:



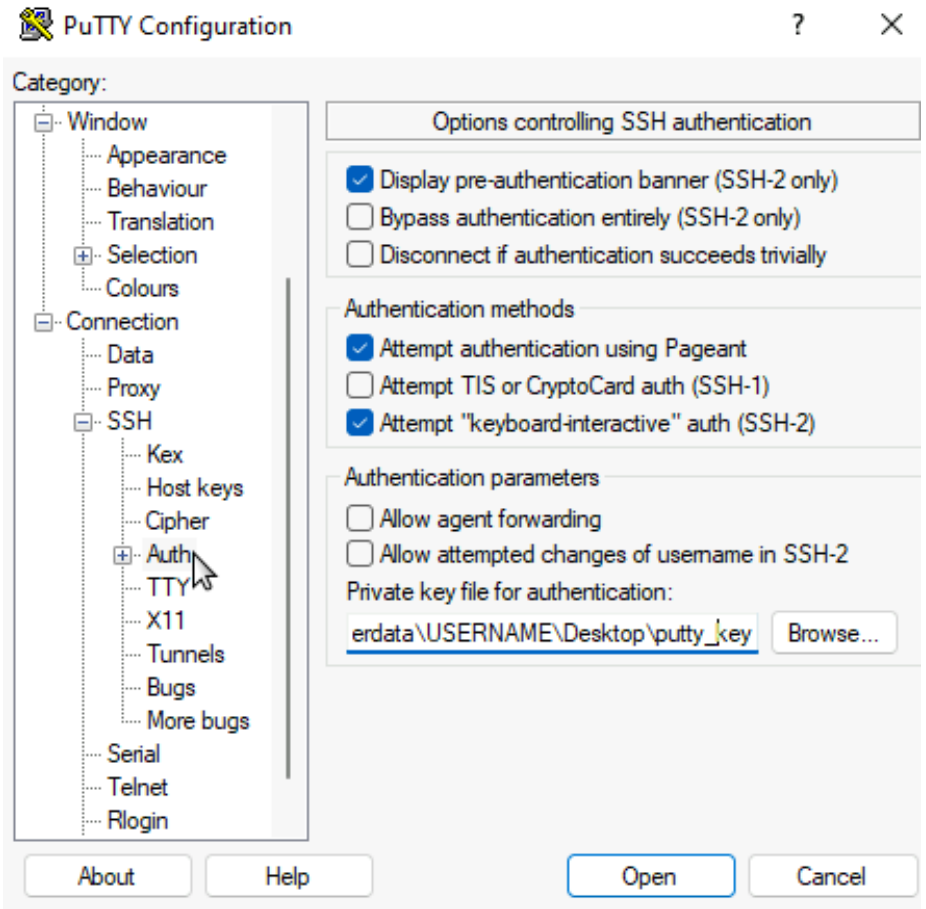
Save the public key with the extension ".pub" eg. id_rsa.pub

Save the private key as well

SSH and Keys

Activity (Windows):

Using PuTTY to access system:



SSH and Keys

Copying your public key to a host:

- Safe for copy-paste (but beware line wrap)
- Can use SCP (Secure Copy Protocol)
- public key needs to be placed into a file called **~/.ssh/authorized_keys** file
- There can be multiple public keys in the authorized_keys file.
- If this file does not exist it needs to be created (owner read/write only (mode 600)).

SSH and Keys

Copying your public key to a host:

- Safe for copy-paste (but beware line wrap)
- Can use SCP (Secure Copy Protocol)
- public key needs to be placed into a file called **~/.ssh/authorized_keys** file
- There can be multiple public keys in the authorized_keys file.
- If this file does not exist it needs to be created (owner read/write only (mode 600)).
  - mkdir -p ~/.ssh

SSH and Keys

Remote Access:

Using putty/terminal/cmd (windows)

- ssh username@remote_host

SUMMARY

In this lesson, you have learnt that:

- SSH with keys are more secure
- Keys with passphrase gives more security
- remote hosts can securely connect through ssh