



## CREATING YOUR DIGITAL SIGNATURE

### Creating A Key :

When you want to use ssh with keys, the first thing that you will need is a key. To create the most simple key, with the default encryption, open up a console, and enter the following command :

```
$ ssh-keygen
```

The ssh-keygen program will now generate both your public key(identity.pub) and your private key(identity).

Your keys are stored in the .ssh/dir in your home directory

*This will output the following :*

Generating public/private rsa1 key pair.

Enter file in which to save the key : /home/abhijit/.ssh/identity

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/abhijit/.ssh/identity.

Your public key has been saved in /home/abhijit/.ssh/identity.pub.

The key fingerprint is:

```
22:bc:0b:fe:f5:06:1d:c0:05:ea:59:09:e3:07:8a:8c abhijit@HGDRD1
```

### Creating a version 2 keypair

In our example we will create a keypair using dsa encryption.

```
$ ssh-keygen -t dsa
```

The file identity2 contains your version 2 private key & the file identity2.pub contains your version 2 public key.

Placing the public key on the remote server

To be able to log in to remote systems using your pair of keys, you will first have to add your public key on the remote server to the authorized\_keys (for version 1) file, and the authorized\_keys2 (for version2) file in the .ssh/ directory in your home directory on the remote machine.

*Which will output the following :*

Generating public/private dsa key pair.

Enter file in which to save the key : /home/abhijit/.ssh/identity2

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/abhijit/.ssh/identity2

Your public key has been saved in /home/abhijit/.ssh/identity2.pub

The key fingerprint is:

```
7b:ab:75:32:9e:b6:6c:4b:29:dc:2a:2b:8c:2f:4e:37 abhijit@HGDRD1
```

In our example we will assume you don't have any keys in the `authorized_keys` files on the remote server. (Hint: If you do not have a remote shell, you can always use your own useraccount on your local machine as a remote shell (`ssh localhost`)).

Now, we will upload the public keys to the remote server

```
$ cd .ssh/
```

```
$ scp identity.pub abhijit@172.16.1.2:./identity.pub
```

```
$ scp id_dsa.pub abhijit@172.16.1.2:./identity2.pub
```

After that we will login on the remote server using `ssh` or `telnet` the conventional way... with a password.

When you are logged in you should create a `.ssh` directory, and inside the `.ssh/` directory create an `authorized_keys` and an `authorized_keys2` file and add the keys to the files.

### **Adding the public key for version 1 & version 2:**

```
$ mkdir .ssh
```

```
$ cd .ssh
```

```
$ touch authorized_keys
```

```
$ chmod 600 authorized_keys
```

```
$ cat ../identity.pub >> authorized_keys
```

```
$ touch authorized_keys2
```

```
$ chmod 600 authorized_keys2
```

```
$ cat ../identity2.pub >> authorized_keys2
```

Log in using your key :

To log in using your key use the `ssh` command.

```
$ ssh abhijit@172.16.1.2
```

### **SSH Keys with a passphrase :**

If I lost my key, the finder would be able to access every system on which I installed my public key.

To sort out this problem we can use a passphrase on our key. This does nothing more than configuring your key so that you have to enter a passphrase to use it.

### **Testing the setup (example)**

---

```
ssh abhijit@172.16.1.2
```

```
Enter passphrase for /home/abhijit/.ssh/identity:*****
```

```
abhijit@172.16.1.2's password:*****
```

After you enter your passphrase and password , it will load the key and use it to authenticate you using `ssh`.