SSH keys

Dr Peadar Grant

February 8, 2024

1 Key-based authentication

SSH key pairs are an alternative to a username/password. They consist of:

Private key kept on the client and securely stored.

Public key on the server(s) you want to log in to. (The public key can be freely shared around, even put up in public.)

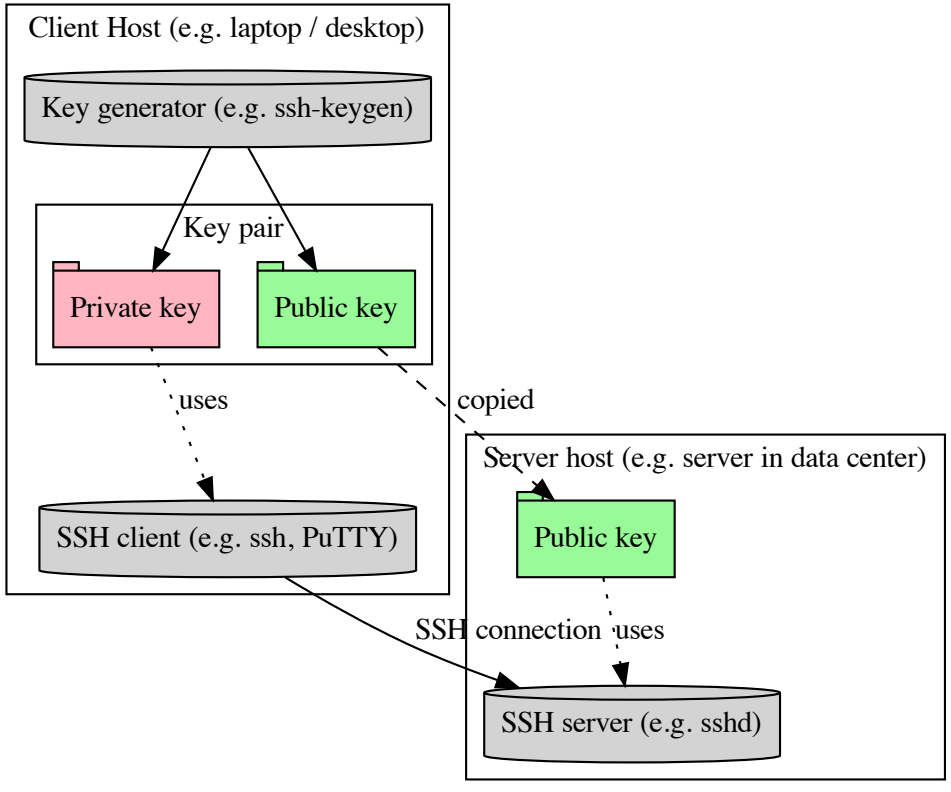

Figure 1: Key-based authentication

1.1 Creating key pair (mac, linux, windows 10 onwards)

Key pairs are normally and ideally created on your own local client computer. Key pairs only need to be generated once. (If you already have a key pair created, you can skip on ahead...)

To create an ED25519 key pair, in Powershell/Bash type:

```
ssh-keygen -t ed25519
```

You can optionally use a passphrase to encrypt the key pair or leave it blank for easier usage.

1.2 Default key locations

The key pair is then stored in two files in your home directory (same for Mac, Linux, Windows). You can find them by changing into the `.ssh` directory and listing the contents of it:

```
cd .ssh
dir
```

From the directory listing:

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        16/10/2020     15:19           3243 id_ed25519
-a----        16/10/2020     15:19            749 id_ed25519.pub
```

The public key is stored in `id_ed25519.pub`. The private key is stored in `id_ed25519`.

You can of course copy these files to/from a memory stick or online storage. Remember though that if your private key is compromised, anybody can use it. Best to protect it with a passphrase!

1.3 Connecting over SSH with keys

In PowerShell/Bash we can use the SSH command to connect to the SSH server on a remote host:

- This will then present us with a new shell on the remote computer (Bash for Linux/UNIX, PowerShell for Windows).
- By default, SSH will try all private keys in `.ssh` so we don't need to specify which.

```
ssh student@$publicIp
```

1.4 Using specific key

We can force the use of a specific key using the `-i` option:

```
ssh -i private_key_file_name_here username@host
```