How to create a SSH keypair (a pair of public/private keys to use for SSH authentication)

Introduction

The most common way to authenticate to a UNIX-like system (*e.g.* Linux, Mac OS, *BSD, etc.) when using the SSH protocol is via standard credentials: the username and password associated to the system user.

This authentication method is not the best from the point of view of security, since an exposed password access can be the target of sophisticated attacks, and a password which is not strong enough can be easily cracked, and the system compromised.

Another (much more secure) authentication method, which makes no use of passwords, is based on a couple of *keys* (*i.e.* a key pair).

How it works

The aforementioned keys are:

- a **private** key, which **must never be shared with anyone, nor put on any storage support accessible via network**
- a **public** key, which can be known by anyone

Such keys work as a lock (the public key) and a key (the private key): whenever an account is secured with the lock (the public key), whoever owns the (private) key can enter.

Concretely, in a UNIX-like system (which we call *e.g.* server.example.com), the public keys are stored in the user's `$HOME/.ssh/authorized_keys` file, one per line. When one tries to connect to the system via SSH

How to create a key pair

A keypair can be created executing the `ssh-keygen` command in a BASH/ZSH (the standard shells in UNIX-like systems and in the Windows Subsystem for Linux (WSL) in Windows) **on the client**, answering to the prompted questions (the default answers, *i.e.* the one obtained by pressing the return button to all questions, are fine). The following is an example:

```
$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key ($HOME/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in $HOME/.ssh/id_ed25519.
Your public key has been saved in $HOME/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:YHUT239POA6FuRGK2rVBixAhKFfwTzZtkLunIRGM3Ps
<USERNAME>@server.example.com
The key's randomart image is:
+---[RSA 2048]----+
| ..Bo =+. =..    |
|. =.+..= = * +   |
| o  .o*.= * = .  |
|    o=.* . o = . |
|     oo.S . o + o|
|    . E .    o +.|
|     . +      . .|
|      .          |
|                 |
+----[SHA256]-----+
```

The above command (with the default answers to the prompted questions) generates two files in the `$HOME/.ssh`:

- `$HOME/.ssh/id_ed25519`, containing the **private** key
- `$HOME/.ssh/id_ed25519.pub`, containing the **public** key

One must then transfer the copy of the `$HOME/.ssh/id_ed25519.pub` (the public key) inside the `$HOME/.ssh/authorized_keys` file (in a single line) **on the server**.

The following is an example of a public key (`id_ed25519.pub`) generated with the above procedure (which can be visualized with the command `cat $HOME/.ssh/id_ed25519.pub`; please remember that the key must always be contained in a **single** line):

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIAD2c8813ECZvHDvVO7c7yidsrtMaomJ+JqqbCiYKr4Y
<USERNAME>@<CLIENT>
```

Where the corresponding private key (`id_ed25519`) is (which can be visualized with the command `cat $HOME/.ssh/id_ed25519`; please note that it is **not** contained in a single line):

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACAA9nPPNdxAmbxw71Tu3O8onbK7TGqJifiaqmwomCq+GAAAAJAIpvjKCKb4
ygAAAAtzc2gtZWQyNTUxOQAAACAA9nPPNdxAmbxw71Tu3O8onbK7TGqJifiaqmwomCq+GA
AAAEDwpLlBDxHysEZhOMtkVbIayhW5a/pmG/ZWQct+EdJk0QD2c8813ECZvHDvVO7c7yid
srtMaomJ+JqqbCiYKr4YAAAAC3N0ZUBrZXRjaHVwAQI=
-----END OPENSSH PRIVATE KEY-----
```

How to connect to a server via SSH using a key pair

One can connect to the server via SSH executin the following command **on the client**:

```
ssh -i <PATH_TO_PRIVATE_KEY> <USERNAME>@<SERVER>
```

replacing

- `<PATH_TO_PRIVATE_KEY>` with the (absolute or relative) path to the file containing the **private** key on the file system of the client
- `<USERNAME>` with the username of the user on the **target** system (*i.e.* **on the server**)
- `<SERVER>` with the *fully qualified domain name* (FQDN, *e.g.* server.example.org) or the IP address **of the server**

The system will then ask for the passphrase, if one has been set up when creating the key pair, otherwise access to the server will be granted directly.

Please notice that many UNIX-like systems automatically try to use the private keys stored in the user's `$HOME/.ssh` directory when connecting to a remote server via SSH.

Troubleshooting

It can happen that the server answers with a `Permission denied (publickey)` message when one tries to connect. This can happen for different reasons:

Wrong private key

If one tries to connect to the server using a private key not corresponding to the one copied inside the `$HOME/.ssh/authorized_keys` file, the system will deny access. If this happens, **one must make sure to use the correct key, specifying it with the `-i` flag of the `ssh` command**:

```
ssh -i <PATH_TO_PRIVATE_KEY> <USERNAME>@<SERVER>
```

replacing

- `<PATH_TO_PRIVATE_KEY>` with the (absolute or relative) path to the file containing the **private** key on the file system of the client
- `<USERNAME>` with the username of the user on the **target** system (*i.e.* **on the server**)
- `<SERVER>` with the *fully qualified domain name* (FQDN, *e.g.* server.example.org) or the IP address **of the server**

File and directory permission

Please notice that (on the **server**) the `$HOME/.ssh` (hidden, as it starts with a `.`) directory must have `drwx------` (or `700`) permissions, while the `$HOME/.ssh/authorized_keys` must have `-rw-------` (or `600`) permissions. These permissions can be set executing the following commands on the server (server.example.com in the previous example):

```
$ chmod 700 $HOME/.ssh
$ chmod 600 $HOME/.ssh/authorized_keys
```

Moreover, the `$HOME/.ssh/id_ed25519` **private** key (on the **client**) must have the `-rw-------` (or `600`) permissions, which can be set executing the following command:

```
$ chmod 600 $HOME/.ssh/id_ed25519
```