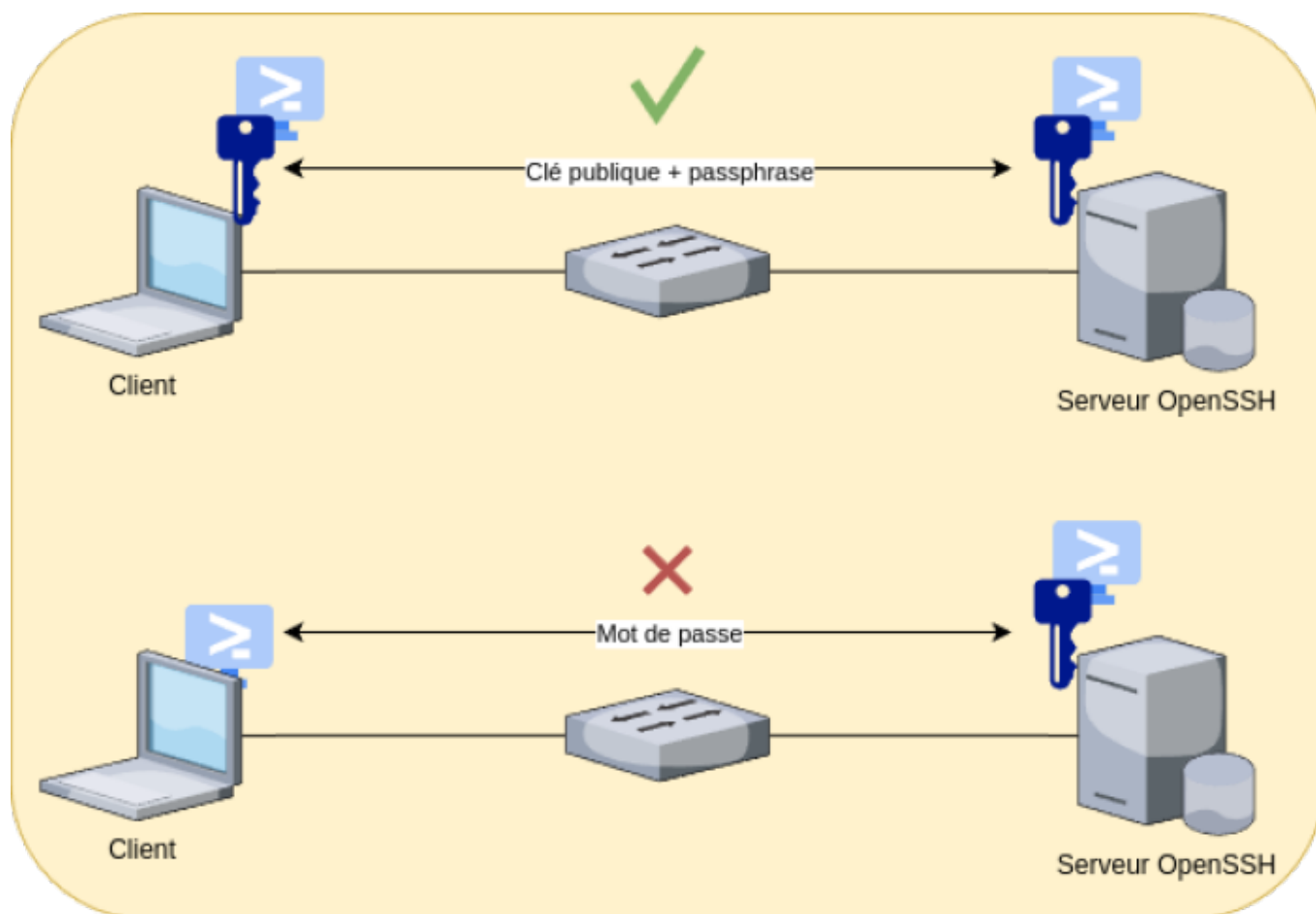


Authentification par Clé publique

L'authentification **SSH** par **clé** permet de choisir les clients **autorisés** à se connecter sur le serveur SSH. Mais l'intérêt est aussi de pouvoir tous les **désactiver** en retirant la clé de **confiance** sur le serveur.



Sources importantes



- [Article IT-Connect : Authentification SSH par clés](#)
- [Article DigitalOcean : Comment configurer une authentification par clé SSH sur un serveur Linux](#)

Générer la clé publique

La clé doit être générée par le client et par la suite transmise au serveur ssh.

installer le paquet OpenSSH si il ne l'est pas.

```
sudo apt install openssh-client
```

Puis générer la clé avec **ssh-keygen**.

La clé sera générée dans le dossier **/home/utilisateur/.ssh/**

```
ssh-keygen
```

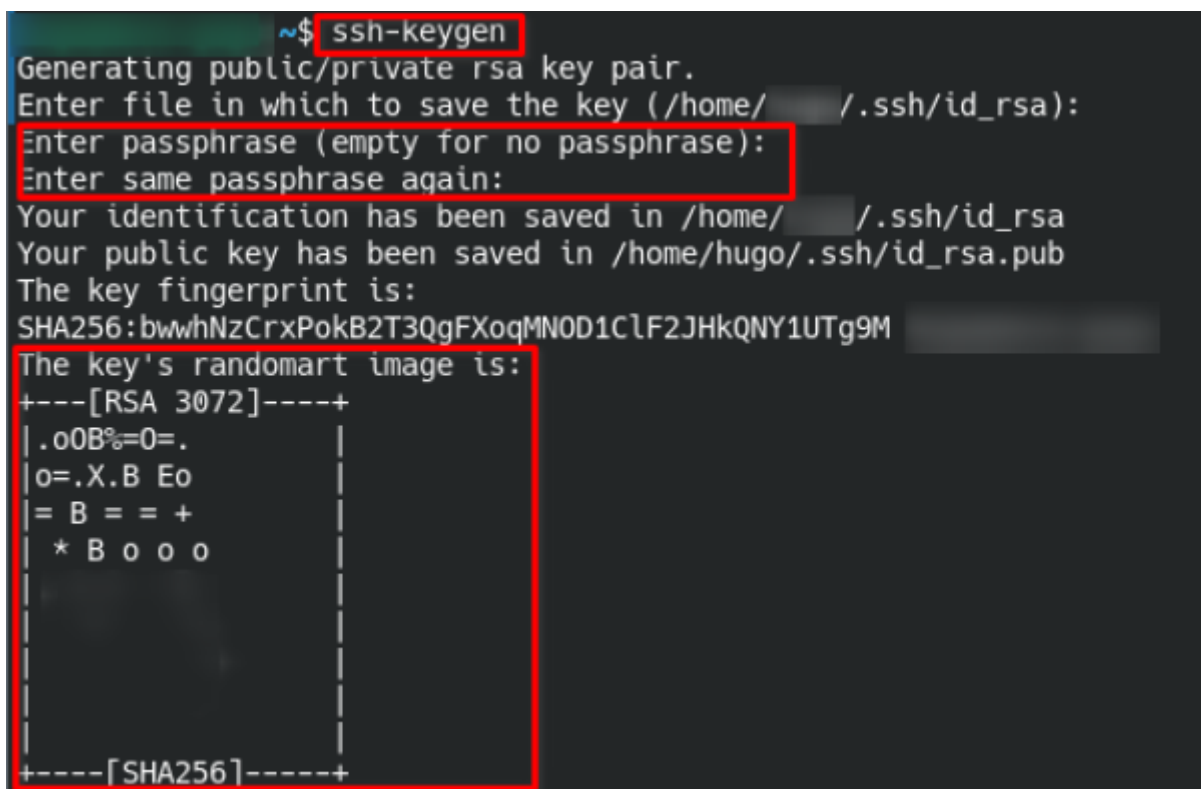
Si aucune option n'est spécifiée, une clé RSA de 2048 bits sera créée.

L'option **-b** permet de spécifier la taille.

```
ssh-keygen -b 4096
```

Afin de protéger la clé en cas de compromission, il faut définir une passphrase.

Son Fingerprint sera également affiché.

A terminal window showing the execution of the 'ssh-keygen' command. The prompt is '~\$'. The command 'ssh-keygen' is entered and highlighted with a red box. The output shows the generation of a public/private RSA key pair. It prompts for a file name (defaulting to '/home/.../.ssh/id_rsa'), a passphrase (empty), and confirmation of the passphrase. It then shows the key's identification path, public key path, and fingerprint (SHA256: bwwhNzCrXPokB2T3QgFXoqMNOD1ClF2JHkQNY1UTg9M). Finally, it displays the key's randomart image, which is also highlighted with a red box. The randomart image is a square graphic with a dashed line on the right side, containing the text '[RSA 3072]' at the top and '[SHA256]' at the bottom.

```
~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/.../.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/.../.ssh/id_rsa
Your public key has been saved in /home/hugo/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:bwwhNzCrXPokB2T3QgFXoqMNOD1ClF2JHkQNY1UTg9M
The key's randomart image is:
+---[RSA 3072]---+
|.o0B%0=.
|o=.X.B Eo
|= B = = +
* B o o o
+-----[SHA256]-----+
```

Configurer la clé sur le serveur

Avec **OpenSSH** on peut ajouter la clé avec **ssh-copy-id**.

```
ssh-copy-id utilisateur@serveur
```

-p Pour spécifier le port.

```
ssh-copy-id -p 500 utilisateur@serveur
```

```
~$ ssh-copy-id -p [redacted]@[redacted]
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: '/home/[redacted]/.ssh/id_rsa.pub'
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
[redacted]@[redacted]:~$
Number of key(s) added: 1
```

Il faudra spécifier le mot de passe une dernière fois.

La connexion avec ce client se fera désormais avec la passphrase.

```
~$ ssh [redacted]@[redacted] -p [redacted]
Enter passphrase for key '/home/[redacted]/.ssh/id_rsa':
Linux [redacted] 5.10.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian
```

Désactiver la connexion par mot de passe



Avant de désactiver la connexion par mot de passe, s'assurer qu'au moins un client peut se connecter par clé.

Éditer le fichier **/etc/ssh/sshd_config**.

```
sudo vim /etc/ssh/sshd_config
```

Éditer ces 2 lignes :

```
PasswordAuthentication no
PubkeyAuthentication yes
```

Puis relancer le service **sshd**.

```
sudo systemctl restart sshd
```

Debugger la connexion

Activez le mode verbose sur le client pour déboguer :

```
ssh -vv user@server
```

From:
<https://www.hugo-mattaliano.fr/wiki/> - Gogo Wiki

Permanent link:
https://www.hugo-mattaliano.fr/wiki/doku.php?id=wiki:ssh:cert_auth&rev=1721074046

Last update: **2024/08/21 14:20**



