

# Générer une clé SSH

Au BIK'LAB, nous utilisons l'authentification via SSH, pour administrer à distance nos serveurs ou publier des commit git sans (re)taper de mot de passe.



Ce tuto vous montre simplement comment **générer une paire de clé** sur une station GNU/Linux ou Unix (ex: Ubuntu ou MacOSX) et **partager la clé publique**.

## SSH : des clés asymétriques

SSH (Secure Shell) est un protocole de communication sécurisé fondé sur le principe de clé asymétrique :

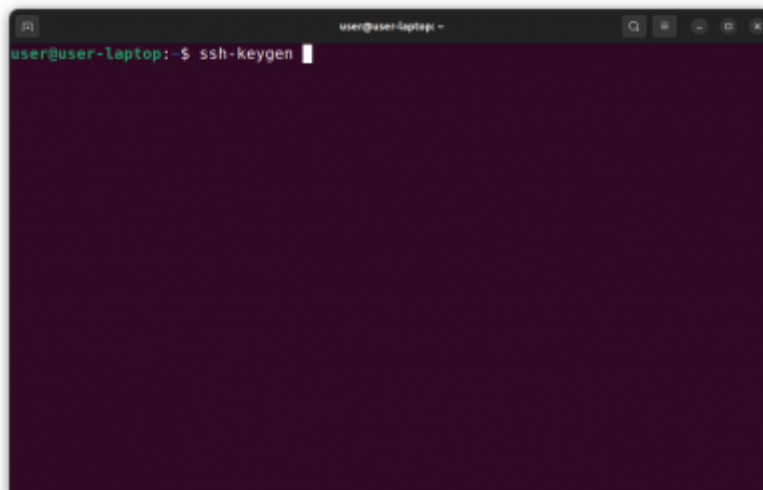
1. une clé publique ;
2. une clé privée.

Ce système de clé asymétrique se matérialise donc par 2 fichiers, pour chacune des clés. La clé publique peut être diffusée librement sur les réseaux même publics et non sécurisés alors que **la clé privée doit absolument rester secrète sur votre disque dur**.

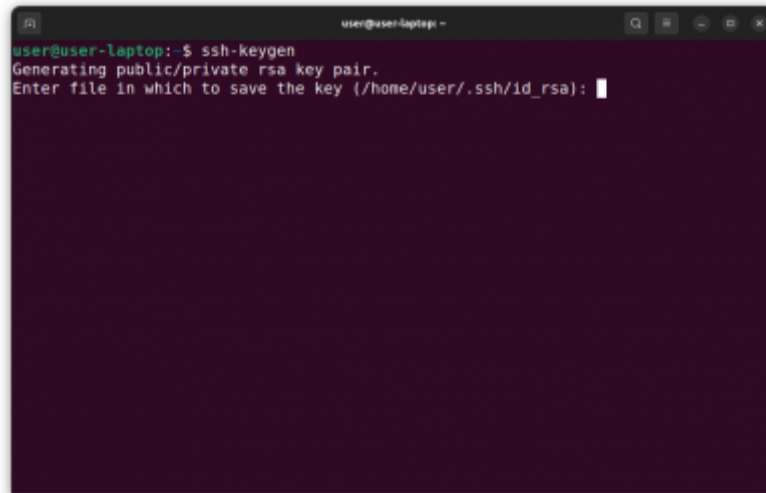
## ssh-keygen pour la génération d'une clé SSH

Pour des questions de rapidité d'authentification, nous choisissons par défaut le chiffrement RSA.

1. Ouvrez un terminal et tapez la commande `ssh-keygen` et suivez les instructions affichées

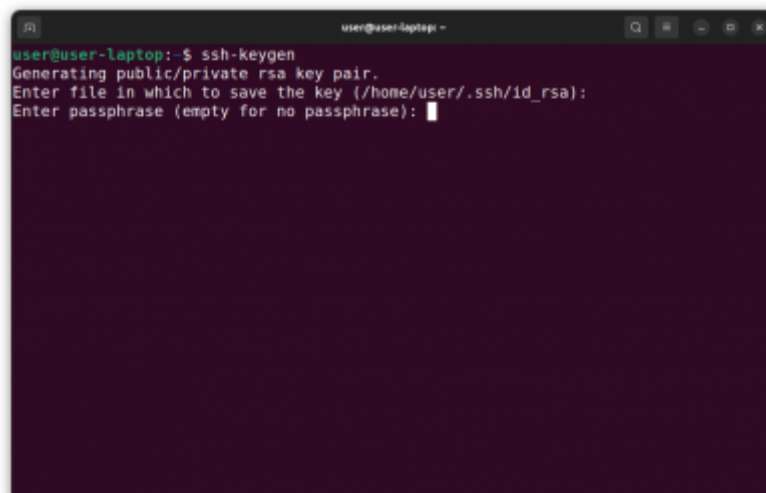


2. Validez le chemin proposé par défaut



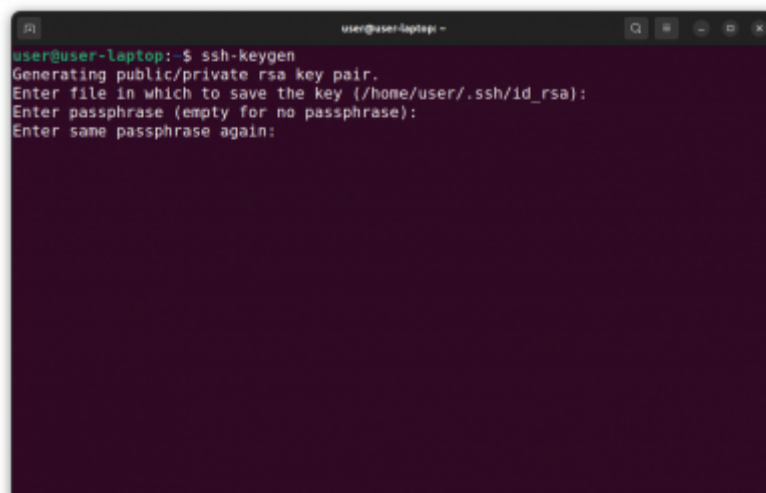
```
user@user-laptop: ~  
user@user-laptop:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/user/.ssh/id_rsa):
```

3. Validez pour une passe phrase vide



```
user@user-laptop: ~  
user@user-laptop:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/user/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):
```

4. Revalidez pour confirmer que vous voulez une passe phrase vide



```
user@user-laptop: ~  
user@user-laptop:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/user/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

5. Validez pour générer la paire de fichiers correspondants à votre clé

```
user@user-laptop:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa
Your public key has been saved in /home/user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:WLRxR0gpFj8AtLyiu9PYMhm/3p2a1YjStLwYYRala4 user@user-laptop
The key's randomart image is:
+---[RSA 3072]-----+
|  o.o.==+o+o      |
|  o o.oo+o.      |
|  o o +. .o       |
|  .+.+. .        |
|  +..S           |
|  E.++. o        |
|  o.o. o .       |
|  .+.+.         |
|  =X+.          |
+---[SHA256]-----+
user@user-laptop:~$
```

Et voilà, vous avez généré une clé SSH, ce qui se matérialise par 2 fichiers dans votre repertoire utilisateur.

La commande `ls ~/.ssh/id*` permet de lister les fichiers commençant par `id` dans le repertoire `.ssh`, repertoire utilisé pour stocker vos clés SSH.

```
user@user-laptop:~$ ls ~/.ssh/id*
/home/user/.ssh/id_rsa
/home/user/.ssh/id_rsa.pub
```

- Le fichier `id_rsa` est votre clé secrète que vous devez impérativement garder confidentielle ;
- le fichier `id_rsa.pub` est **vosre clé publique que vous pouvez diffuser sans modération ni risque**, tant que votre clé secrète reste secrète.

## Protection d'une clé SSH

Au niveau système, la clé secrète doit donc être inaccessible aux autres utilisateurs, tandis que votre clé publique devrait être accessible en lecture à tout le monde.

### Droits d'accès aux fichiers de la clé

La commande `ls -la ~/.ssh/id*` permet de lister les fichiers commençant par `id` dans le repertoire `.ssh`, et d'afficher en plus (option `-la`), les droits d'accès des utilisateurs et groupes à vos clés.

```
user@user-laptop:~$ ls -la ~/.ssh/id*
-rw----- 1 user user 2602 janv. 22 09:54 /home/user/.ssh/id_rsa
-rw-r--r-- 1 user user  570 janv. 22 09:54 /home/user/.ssh/id_rsa.pub
```

# Diffusion d'une clé SSH

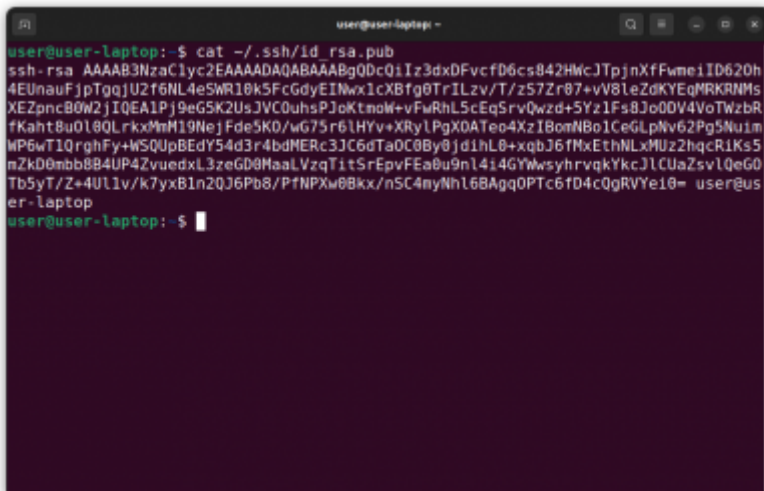
L'architecture clé asymétrique permet une large diffusion de votre clé publique sans l'exposer.

Idéalement, vous devez donc rajouter cette clé sur votre page de profil, ou dans votre signature, pour permettre aux admins de l'ajouter aux différents services auxquels vous pouvez accéder.

## Partager la clé en mode texte

Une clé est en fait une paire de fichiers texte (ASCII). Il est donc possible de partager sa clé soit via le fichier, soit en partageant la chaîne de caractères du fichier `id_rsa.pub`.

La commande `cat` permet de lire le contenu d'un fichier dans le terminal. Vous pouvez aussi utiliser votre éditeur préféré (certainement, `codium` ?)



Dans ce cas, il suffit de copier cette chaîne de caractères sur votre profil

```
ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQGDcQiIz3dxDFvcfD6cs842HwcJTpnXfFwmeiID620h4EU  
auFjpTgqjU2f6NL4eSWR10k5FcGdyEINwx1cXBfg0TrILzv/T/zS7Zr07+vV81eZdKYEqMRKR  
NMsXEZpncB0W2jIQEA1Pj9eG5K2UsJVC0uhsPJoKtmoW+vFwRhL5cEqSrvQwzd+5Yz1F  
s8JoODV4VoTWzbrfKaht8u0l0QLrkkMm19NejFde5K0/wG75r6lHYv+XRylPgX0ATeo4Xz  
IBomNBo1CeGLpNv62Pg5NuimWP6wT1QrghFy+WSQUpBEdY54d3r4bdMERC3JC6dT  
a0C0By0jdiHl0+xbJ6fMxEthNLxMUz2hqcRiKs5mZkD0mbb8B4UP4ZvuedxL3zeGD  
0MaaLVzqTitSrEpvFEa0u9n14i4GYWwsyhrvqkYkcJlCUaZsvlQeG0Tb5yT/Z+4U1  
lv/k7yxB1n2QJ6Pb8/PfNPXw0Bkx/nSC4myNh16BAgqOPTc6fD4cQgRVYe10= user@  
user-laptop
```



## Ajouter sa clé à un utilisateur Linux

Todo

## Ajouter sa clé SSH pour l'utilisateur pi d'un Raspberry Pi

Todo

## Ajouter sa clé SSH sur le GitLab

Todo

[[linux](#), [ssh](#)]

From:  
<https://wiki.lebiklab.fr/> - Wiki Le BIK'LAB

Permanent link:  
[https://wiki.lebiklab.fr/doku.php?id=tutos:hacking:generer\\_cle\\_ssh:start&rev=1683298376](https://wiki.lebiklab.fr/doku.php?id=tutos:hacking:generer_cle_ssh:start&rev=1683298376)

Last update: **04/04/2024 15:35**

