LABORATOIRE D'ELECTRONIQUE APPLIQUEE Q2

Cours destiné aux étudiants de première année en Technologie de l'Informatique et orientation sécurité des systèmes

Table des matières

Laboratoire sur Raspberry

Petits rappels (Raspberry) :

- Attention : La RPI[1] est alimentée en 5V par un adaptateur en 5V DC. **MAIS ATTENTION** : les composants (ce qui est connecté sur la GPIO) travaillent en 3,3V. Ceci signifie que c'est bien cette tension qu'il faut prendre en compte pour les calculs (de résistance sur une LED, par exemple)
- Attention : Une Led se raccorde **toujours** avec une résistance en série !
- La tension d'une Led varie en fonction de la couleur. (Rappel : Led verte = 2V, 20mA)
- La Raspberry est un SoC[2]. Il faut donc installer un OS[3] sur le support du système de fichiers.
- Sur la RPI, il n'y a pas, à priori, de disque dur. L'OS et tous les autres fichiers sont sur un support mémoire : la carte micro-SD.
- Pour l'installation de l'OS sur la carte mémoire, il faut graver une image.
- Installation d'une RPI **sans** intervention sur la **configuration** des **PCs du laboratoire.**
- L'installation d'une Raspberry peut se faire avec une intervention minimale sur une configuration quelle qu'elle fut. Plus qu'une contrainte, la démarche proposée ici offre un net avantage en terme de temps et de confort. Les seules connections à réaliser seront l'alimentation et le réseau. On peut même pousser la simplification à la seule connexion de l'alimentation, si on est dans un environnement wifi disponible.

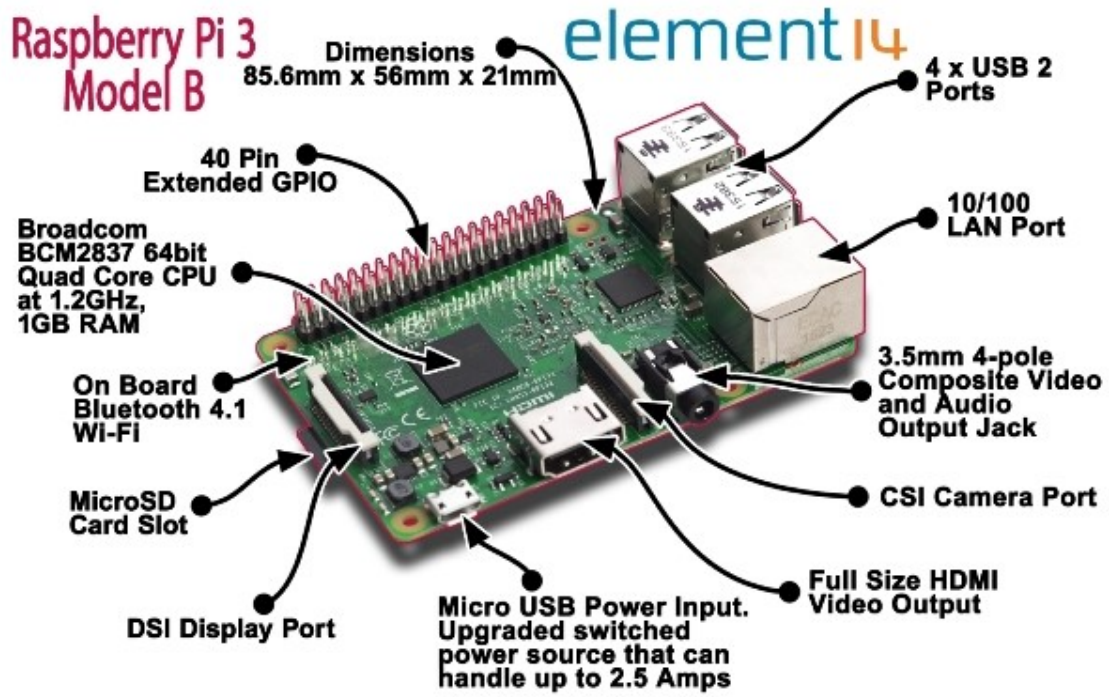


Image venant de : https://www.element14.com

[1] Raspberry
[2] SoC: System on Chip
[3] OS: Operating System

Manip 1 : configuration WiFi + ajout de sécurité sur la RPI

Configuration du WiFi

Plutôt que de connecter une Raspberry via câble (HDMI, RJ45, souris clavier) chez soi, il est plus facile d'utiliser le wifi quand on en a un. Il est possible de faire cette connexion en débranchant le matériel et de "chambouler" une installation audiovisuelle qui avait pourtant été réalisée avec grand soin (tant au laboratoire que chez soi).

Un technicien, un vrai ☺, n'aura pas peur de la ligne de commande. Pour ce faire, nous allons éditer un fichier (Id_str n'est pas obligatoire mais peut être utile pour avoir un nom plus parlant que le SSID) :

"**sudo cat /etc/wpa_supplicant/wpa_supplicant.conf**"

Dans lequel on ajoutera les éléments suivants :

network={

ssid="<le ssid du wifi auquel vous voulez vous connecter>"

psk="<le mot de passe>"

}

network={

ssid="<le ssid d'un autre wifi auquel ... >"

psk="<le mot de passe ... de l'autre>"

}

(On en met autant qu'il n'en faut (Wifi labo, Wifi maison, etc..))

Attention, il est déconseillé de mettre le mot de passe de connexion (en clair) directement dans le fichier.

Nous allons donc crypter le mot de passe :

La commande **wpa_passphrase** permet cela.

La commande a comme arguments : le SSID et la clé d'accès.

Exemple : si le SSID est « toto » et la clé d'accès « Tigrou007» la commande s'écrit :

wpa_passphrase "toto" "Tigrou007". La sortie de la fonction peut être écrite directement dans le fichier wpa_supplicant.conf dans psk. Attention, il ne faut **plus de guillemets sur le psk** !

(psk =894bbb4830fdb8594fac9e66add31c1152b8e70c1dfb916a0cc54cc1430afeef)

… et voilà ! Au bénéfice d'un reboot, vous aurez votre adresse IP (par mail). Vous ne saurez-vous connecter à votre Raspberry **QUE** si vous êtes dans le même sous-réseau. Vous pourriez également vérifier la connexion via une des commandes :

- ifconfig wlan0
- Ip –br addr show wlano

Via votre Raspberry, il est désormais possible de naviguer sur internet, si votre router est lui-même connecté à internet. Pour pouvoir le faire de l'extérieur de votre réseau privé, il faut rediriger le port du serveur ssh (de la RPI) sur l'adresse publique du routeur.

Sécurisation de la Raspberry

Avant-propos :

Si votre routeur n'était pas accessible de l'extérieur, maintenant, il l'est (via votre RPI) ! Il s'agit évidemment d'une grande faille de sécurité. (Il faut toutefois connaître le mot de passe de votre router.) Nous allons nous attacher à la colmater. Voici comment nous allons procéder :

- Personnaliser le mot de passe de la Raspberry
- Changer le port du serveur SSH
- Supprimer la possibilité de connexion SSH via mot de passe, et le remplacer par une clé (elle-même protégée par mot de passe … ou pas).

Personnaliser le mot de passe de la Rasp.

La modification du mot de passe est assez simple et se fait avec la commande "passwd". A noter que la distribution de Raspbian est un peu différente de Debian. Sur la Raspberry, l'utilisateur "pi" est un pseudo-administrateur, et a tous les droits. La personnalisation du mot de passe de "pi" est donc très importante.

Changer le port du serveur SSH

Il faut éditer le fichier de configuration du serveur SSH :

sudo nano /etc/ssh/sshd_config

Dans ce fichier, on cherche la ligne "# port 22". Il faut décommenter la ligne et mettre une autre valeur pour le port. Attention à ne pas mettre une valeur qui entrerait en conflit avec un autre service.

Il va sans dire que c'est-ce nouveau port qu'il faut rediriger sur le routeur.

Supprimer le mot de passe "SSH"

Il s'agit ici de remplacer le mot de passe par une clef. Ces clefs vont par 2 : une privée, que l'on ne communique à personne ; et une clef publique, que l'on donne à tous ceux qui sont d'accord de nous accorder un accès à un dispositif[4].

[4] Imaginez une chaîne avec plusieurs cadenas. Un des cadenas est à vous, et vous êtes le seul à en avoir la clef.

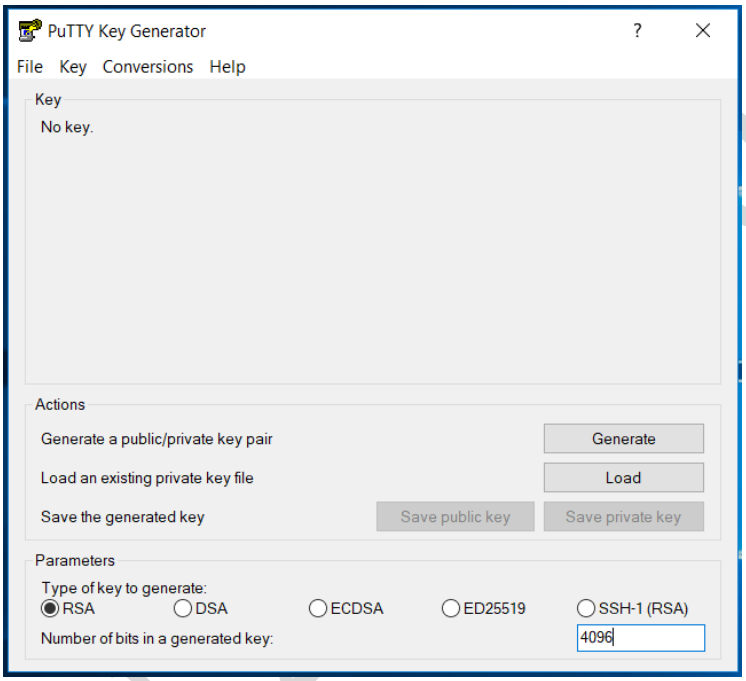
Ici, nous avons plusieurs façons de procéder suivant que nous sommes sous Linux, sous Windows ou sous Mac.

Traitons le premier cas : sous Windows, interface graphique.

Sous Windows, nous utilisons[5] le programme "PuTTy" pour se connecter à la Raspberry. Le plus simple étant de créer un profil que l'on rappellera à chaque instance de connexion avec la Rasp.

Avant d'utiliser une clef, il faut … la créer ! Procédure :

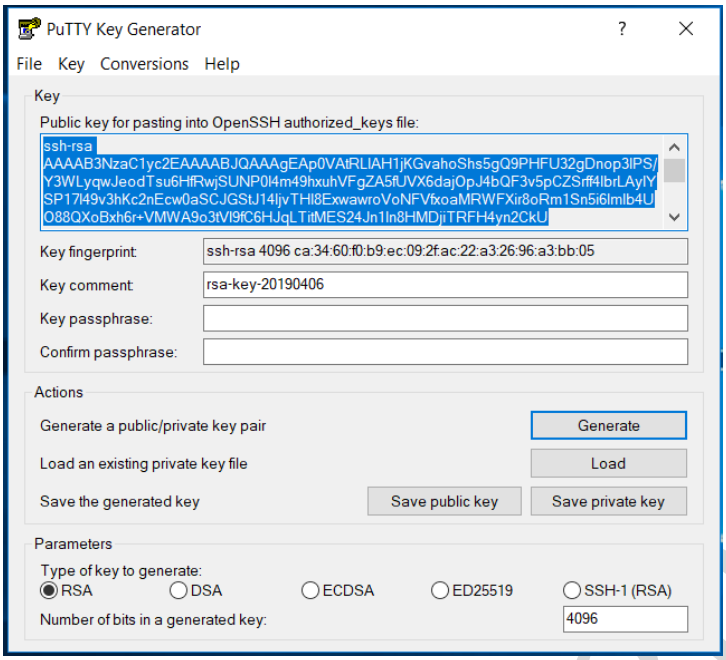
On lance la commande puttygen.exe



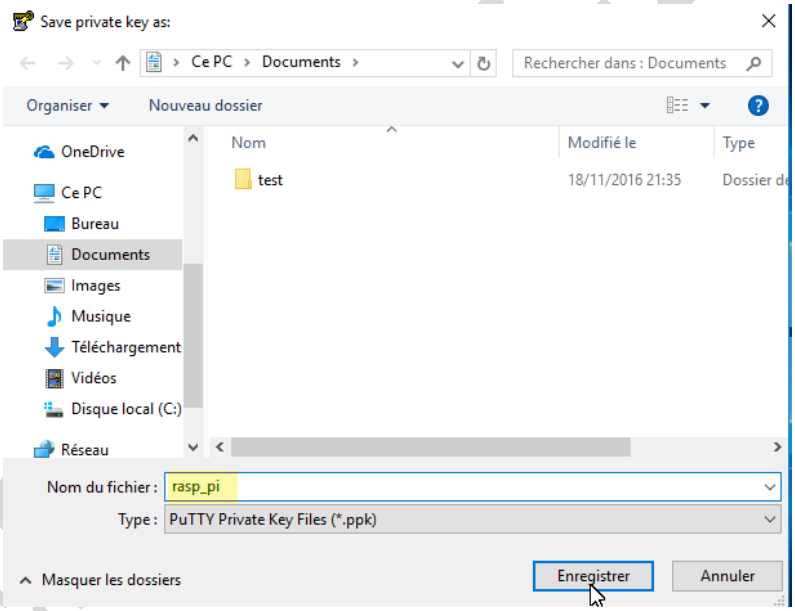
On change la taille de la clef pour "4096 bits"

On lance la génération de la clef par "Generate"[6]. Un des caractères aléatoire va être obtenu par un déplacement de la souris … alors, allez-y!

[5] suivant les consignes du labo ...
[6] Je sais, cela surprend toujours un peu … au début!



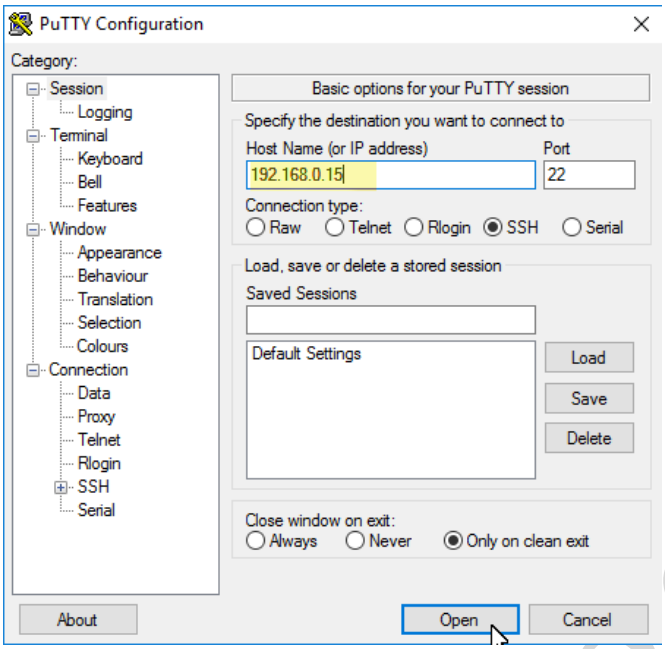
Je vous conseille, là, de mettre un mot de passe pour protéger votre clef[7]



Sauvegardez votre clef là où vous voulez avec le nom que vous voulez.

Copiez la clef publique sur le serveur. Connectez-vous avec PuTTy :

[7] entrez le dans "key parsephrase"



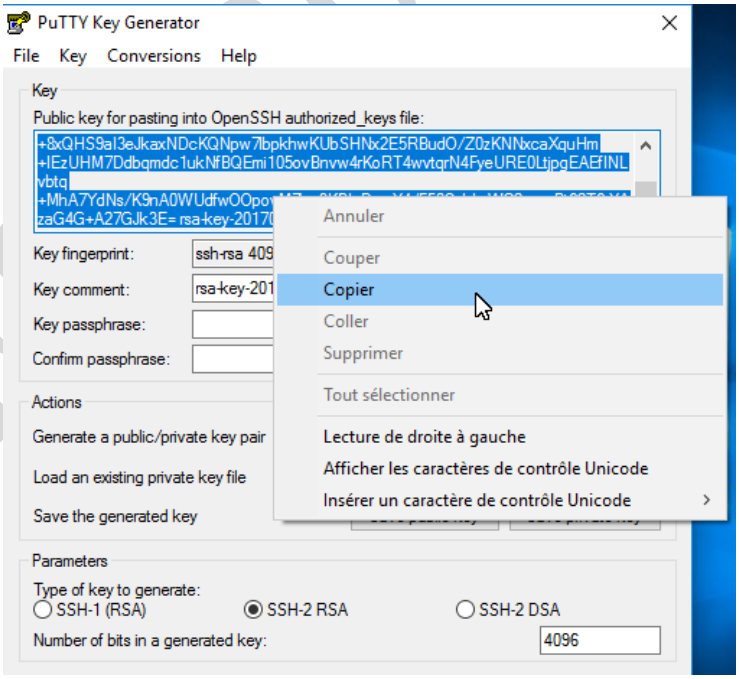
Sur la Rasp, entrez :

- $ cd ~/.ssh

Si le répertoire n'existe pas encore, créez-le

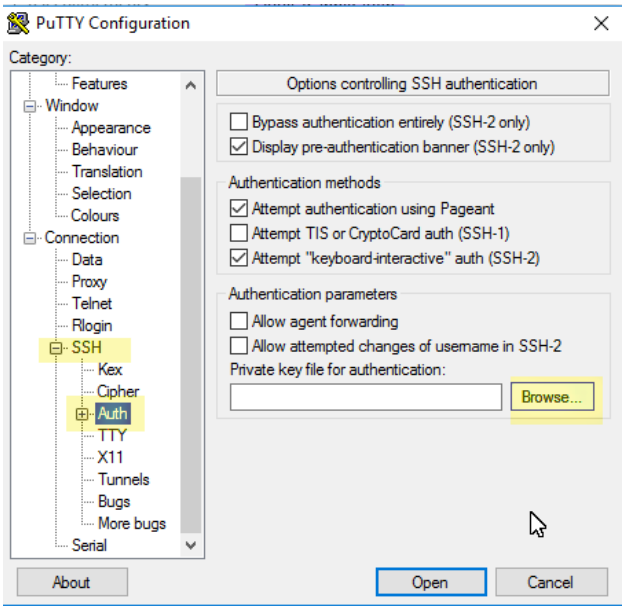
- $ nano authorized_keys

On copie la clef publique de puttykeygen.

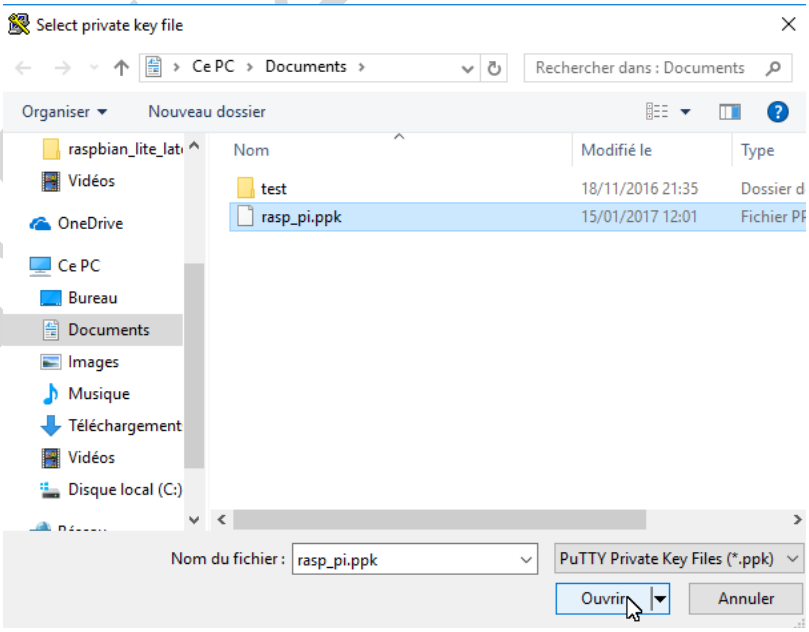


On colle dans nano et on sauve.

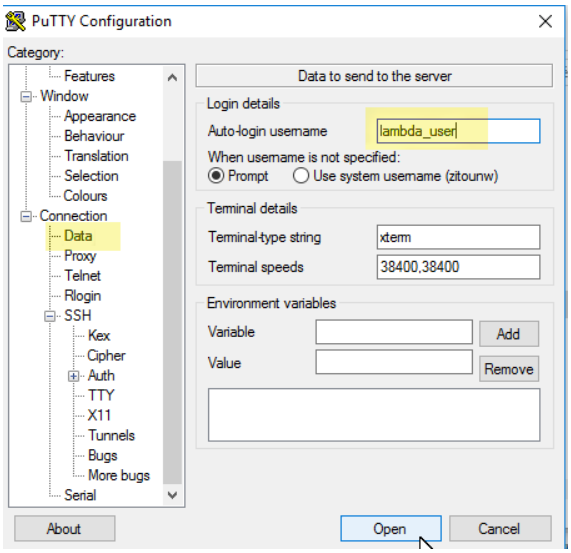
On relance PuTTy et on configure l'utilisation de la clef privée :



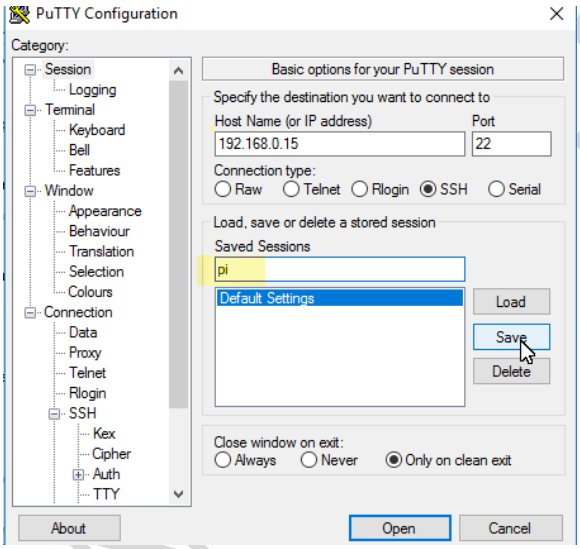
On sélectionne la clef qui vient d'être créée :



Dans PuTTy, on configure un utilisateur pour la connexion :



Et on sauvegarde le nouveau profil :



Désactiver la connexion par mot de passe :

Éditez le fichier de configuration du serveur ssh:

$ sudo nano /etc/ssh/sshd_config

Dans lequel il faut changer la ligne ad-ok en :

PasswordAuthentication no

PubkeyAuthentication yes

Relancer le service ssh, et l'affaire est faite.

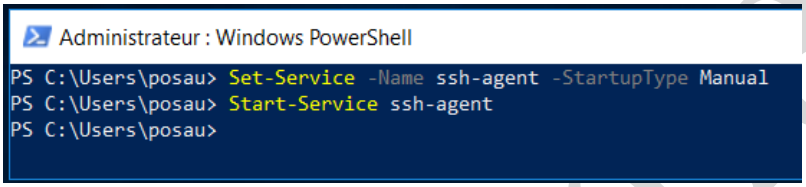
sudo service ssh restart

Traitons le premier cas : sous Windows, en Powershell.

Il faut d'abord, lancer un Powershell en administrateur.

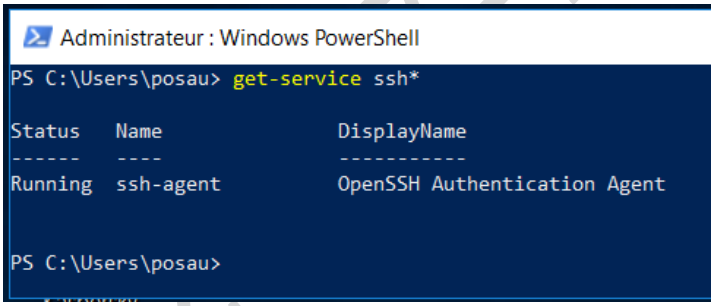
Depuis peu, Microsoft propose un service "openssh" pour son environnement de ligne de commande. Traitons ici le cas du Powershell:

Il faut, tout d'abord, mettre le service en route :



Notez au passage, que si vous remplacez "Manual" par "Automatic", vous ne devrez faire l'opération qu'une seule fois.

Pour vérifier que le service fonctionne :



Les commandes suivantes sont maintenant accessibles :

- ssh
- ssh-add
- ssh-keygen
- ssh-keyscan
- scp
- sftp

Création des clefs :

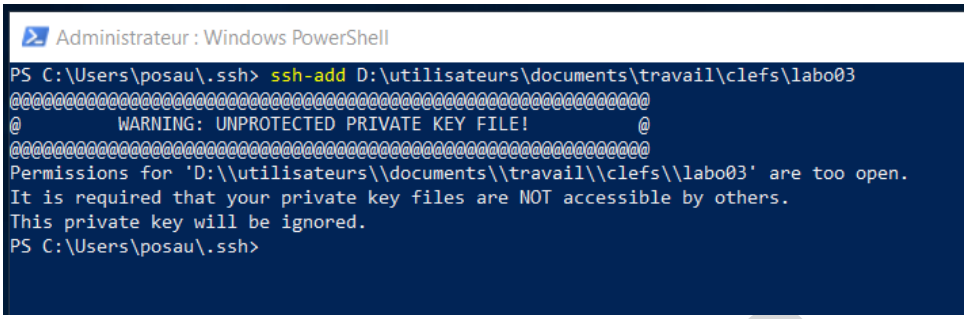
```
cd ~\.ssh

ssh-keygen –b 4096 –t rsa –f  id_rsa
```

Enregistrement de la clef pour le client SSH :

Pour enregistrer la clef et pouvoir vous connecter sans devoir la spécifier à chaque lancement de "ssh", entrez la commande:

```
ssh-add <répertoire et nom du fichier clef>
```

À la suite de quoi, vous aurez certainement l'erreur :



Pour protéger la clef, il faut restreindre les accès au fichiers avec une commande Windows (ma cela, vous savez déjà le faire !).

Pour se connecter :

ssh pi@<ip de votre raspberry>

Cas sous Linux:

On génère d'abord une clef dans un terminal :

cd /home/pi/.ssh

ssh-keygen –b 4096 –t rsa

Suivez les informations et au terme du processus, vous aurez la création de deux clefs. Une publique : id_rsa.pub, et une privée : id_rsa. La clef privée restera sur l'ordinateur avec lequel vous voulez vous connecter. La clef publique est à transférer sur la Raspberry :

Via la connexion ssh, sur la Raspberry :

cd /home/pi/.ssh

touch authorized_keys

chmod 600 authorized_keys

Sur l'ordinateur avec lequel on a créé les clefs :

cd ~/.ssh

cat id_ras.pub | ssh –p 22 pi@<ip de votre Rasp> 'cat >> .ssh/authorized_keys

Éditez le fichier de configuration du serveur ssh:

$ sudo nano /etc/ssh/sshd_config

Dans lequel il faut changer la ligne ad-ok en :

PasswordAuthentication no

PubkeyAuthentication yes

Relancer le service ssh, et l'affaire est faite.

```
sudo service ssh restart
```

Testez ...

Pense-bête :

Commandes utiles :

- "sudo" : pour exécuter une commande avec des droits d'administrateur.
- "nano" : pour éditer un fichier en mode texte. (Permet également de créer un fichier)
- "geany" : pour éditer un fichier en mode graphique. (Permet également de créer un fichier voire un dossier et de l'exécuter)
- "rm" : supprime un fichier
- "rmdir" : supprime un dossier vide
- "rm -r" : supprime un dossier non vide
- "mkdir" : créé un dossier
- "cd" : voyage dans les dossiers
- "ls" : liste les fichiers et dossiers
- "reboot" : redémarre la RPI
- "shutdown" : éteint la RPI
- "python" ou "python3" : Lance le programme python (avec la version 2 ou 3)
- "curl www.icanhazip.com" pour connaitre son adresse IP externe (IPv4)
- “Start-Service –Name ssh-agent –StartupType Manual” réglage du mode de démarrage de l’agent ssh sous Powershell (marche aussi avec Auto”
- “Start-service ssh-agent" démarre le service en question sous Powershell
- “get-service ssh*” liste les service actifs (sous powershell)
- “ssh” connexion à la Raspberry, ou envoie d’une commande
- “ssh-add” (powershell) enregistrement d’une clef privée dans un client ssh
- "scp" : transfert de fichier vers/de un ordinateur ayant un serveur SSH actif (linux)
- "winscp" : … devinez?
- "touch" : créé un fichier vide ou met à jour la date de modification du fichier
- "iwlist wlan0 scan" : liste les réseaux Wifi détecté par la RPI
- "wpa_passphrase" : permet de crypter le mot de passe du Wifi
- "soffice --calc VotreFichierALire" : permet de lire un fichier CSV avec calc