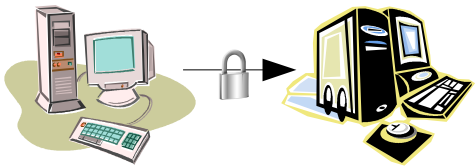
SSH : Secure SHell

Pour l'utilisateur Windows

Version décembre 2011

F. Bongat

Présentation



- Sécuriser des connexions à distance : *Secure Shell*
  - SSH permet de sécuriser les communications des réseaux
    - Utilise pour cela de la cryptographie
    - SSH est composé d'un ensemble d'outils permettant des connexions sécurisées entre des machines. Ces outils ont pour but de remplacer les utilitaires de connexions classiques n'utilisant pas de chiffrage.
      - Remplace : rcp, rlogin, rsh, telnet (ftp par sftp en SSH V2)
    - SSH chiffre et compresse un canal de communication qui sécurise les données transmises (permet d'éviter les sniffers réseaux)
    - Non seulement le mot de passe est chiffré lors de la connexion mais les informations circulant sur le réseau entre les deux machines le sont aussi.

SSH : Les outils à avoir sous windows

- Putty client ssh
- **http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.62-installer.exe**
- Xming permet l'affichage de graphique monde unix
- ftp://ftp.aero.jussieu.fr/pub/Windows/X_server/Xming-6-9-0-31-setup.exe
- ftp://ftp.aero.jussieu.fr/pub/Windows/X_server/Xming-fonts-7-3-0-22-setup.exe
- Filezilla client de transfert de fichiers supportant ssh
- http://downloads.sourceforge.net/filezilla/FileZilla_3.5.3_win32-setup.exe

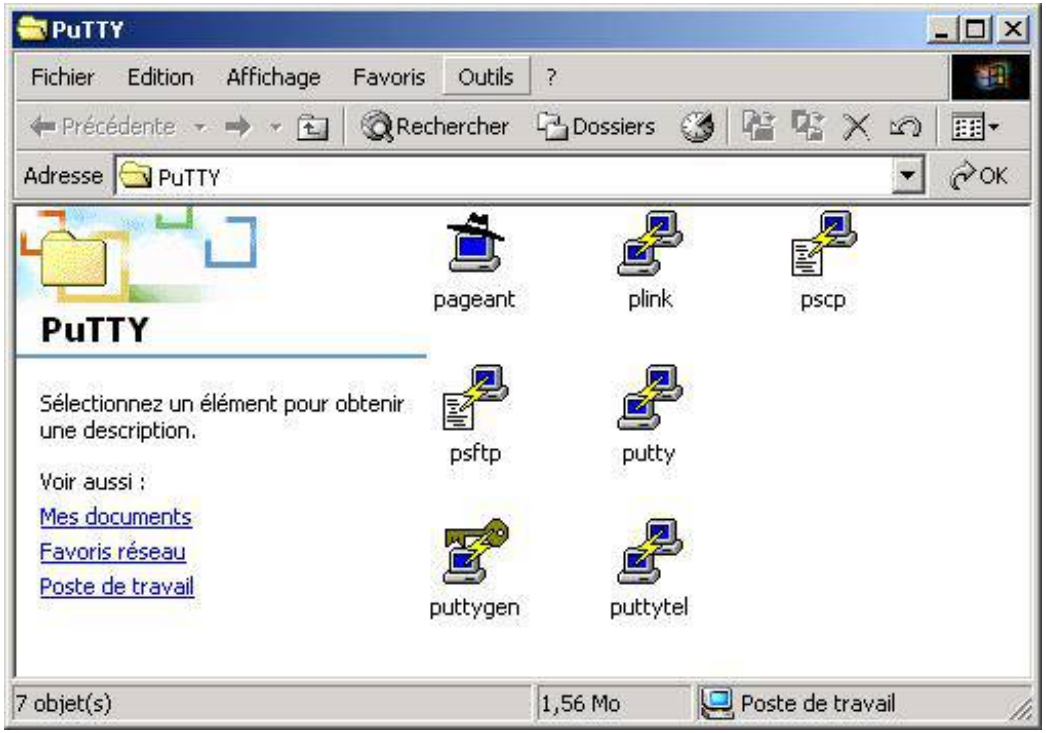
SSH : Les outils à avoir sous windows

- Putty client ssh
- **http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.62-installer.exe**
- Xming permet l'affichage de graphique monde unix
- ftp://ftp.aero.jussieu.fr/pub/Windows/X_server/Xming-6-9-0-31-setup.exe
- ftp://ftp.aero.jussieu.fr/pub/Windows/X_server/Xming-fonts-7-3-0-22-setup.exe
- Filezilla client de transfert de fichiers supportant ssh
- http://downloads.sourceforge.net/filezilla/FileZilla_3.5.3_win32-setup.exe

Côté client : PuTTY - ssh

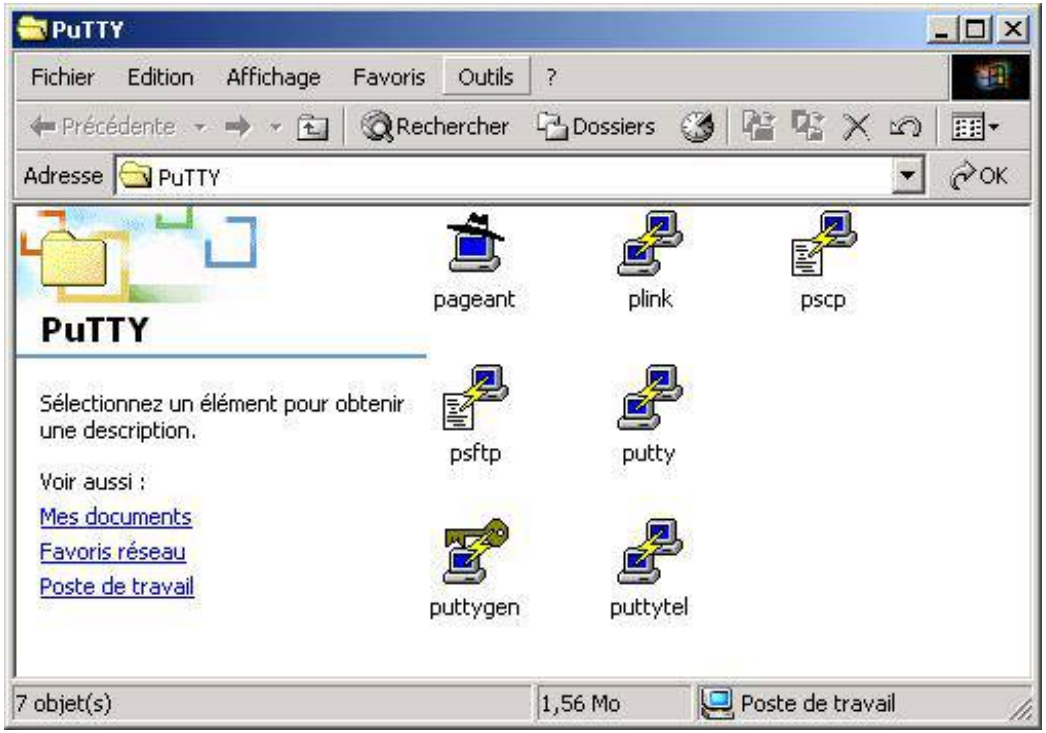
- SSH et Windows : PuTTY
  - Implémentation libre
  - Proche d'OpenSSH
    - Boîte à outils qui comprend un ssh, sftp, scp, ssh-agent et utilise des clés (compatibles avec les clés OpenSSH)
    - 7 binaires (ou un fichier zip) dont 5 indispensables (pageant, pscp, psftp,putty et enfin puttygen) à copier (ou décompresser) dans le dossier :

      C:\Program Files\PuTTY



Côté client : PuTTY - ssh

- SSH et Windows : PuTTY
  - ***pageant*** : agent d'authentification (*voir chapitre authentification forte*)
  - ***plink*** : ssh en mode commande dans une console (~ Invite de Commande)
  - **pscp** : scp en mode console
  - ***psftp*** : sftp en mode console
  - ***putty*** : ssh en mode graphique
  - ***puttygen*** : gestion des clés en mode graphique
  - ***puttytel*** : telnet en mode graphique (pas besoin !)



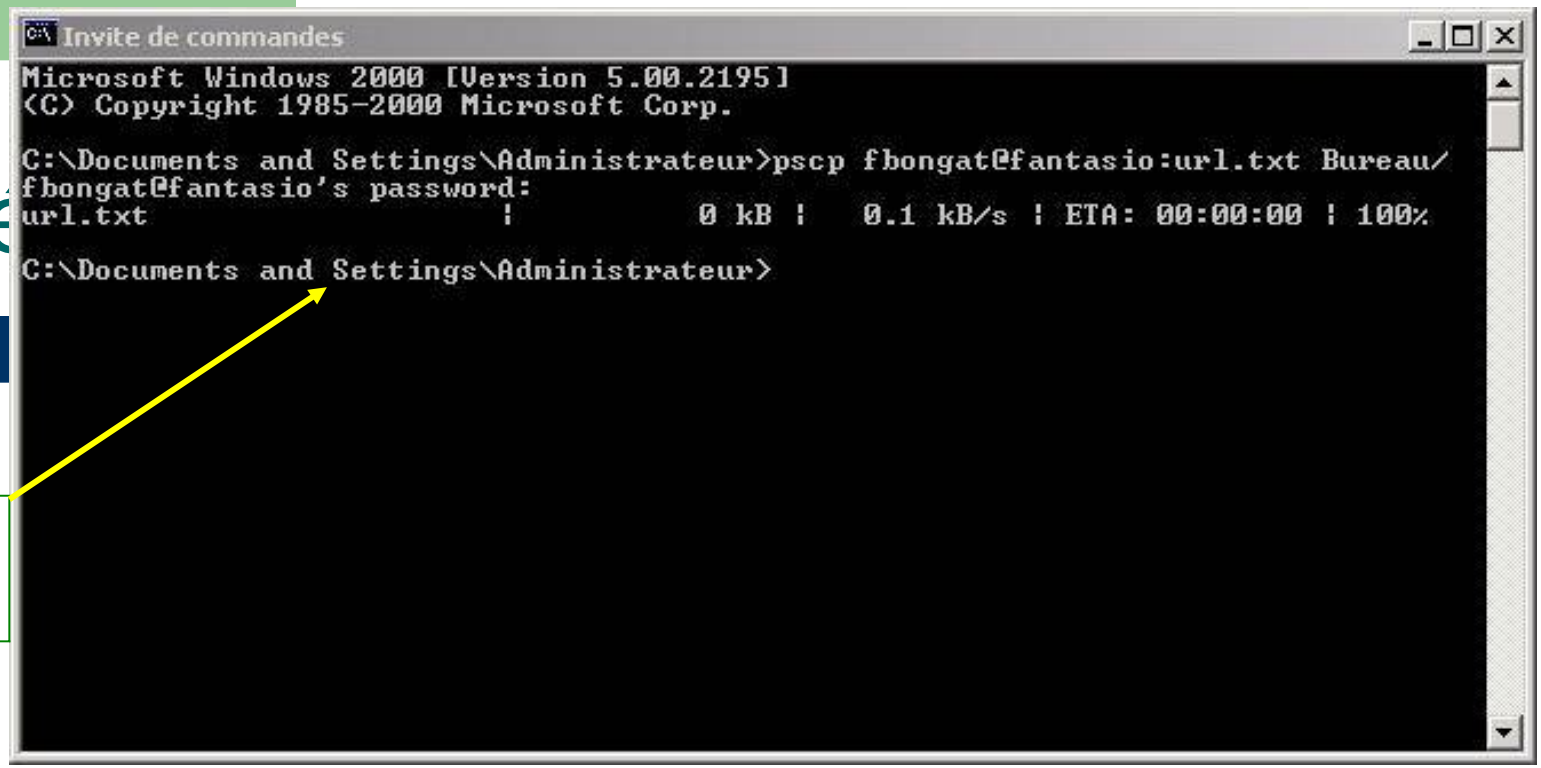
Côté client : PuTTY - ssh

- SSH et Windows : PuTTY
  - SSH
    - connexion rapide avec putty

Ajout de la clé publique dans la base de registres (équivalent au fichier known_hosts)

Répondre « oui » pour passer à la suite !

Côté



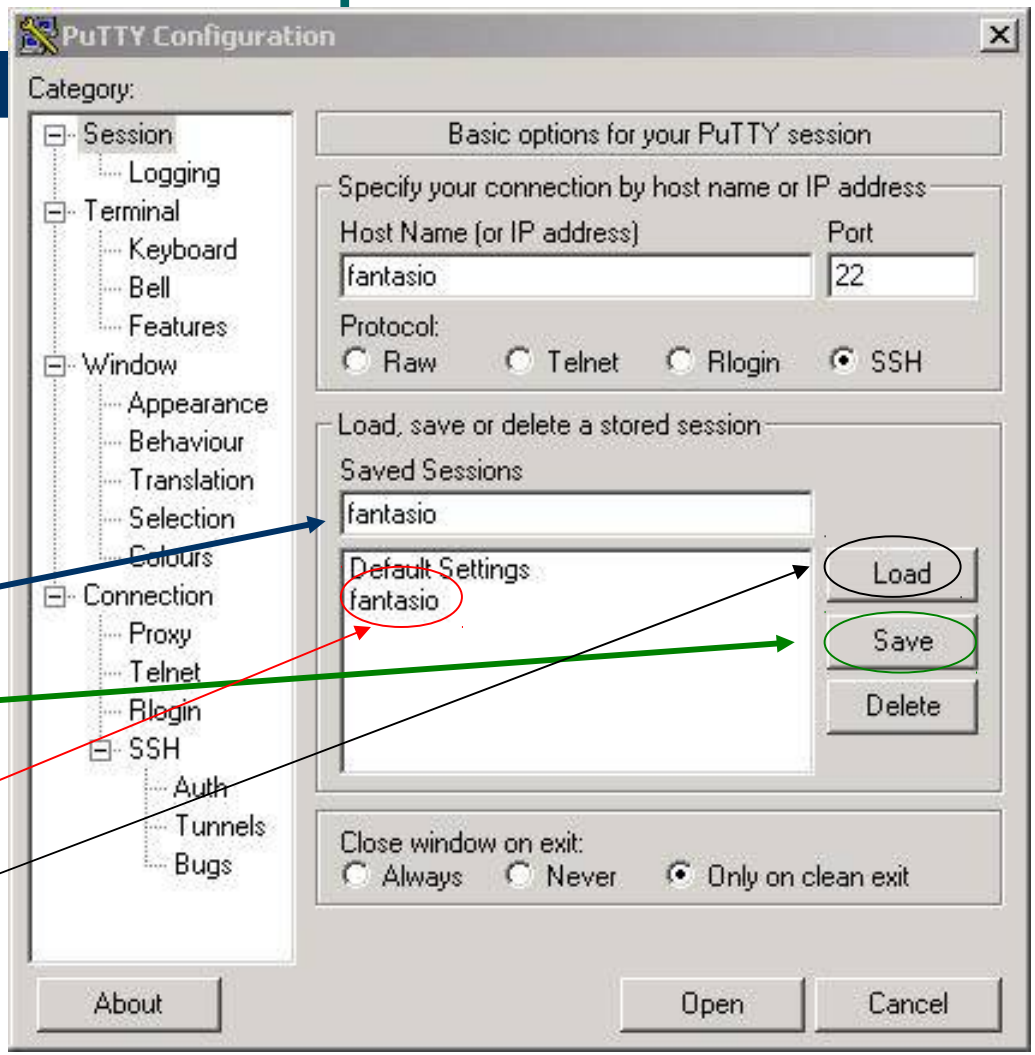
pscp fonctionne comme scp dans un shell

- SSH et Windows : PuTTY
  - SCP
    - Ouvrir une console « ***Invite de commandes*** »
    - Tapper ***pscp*** dans cette fenêtre

      pscp fichier **login**@**machine:**
    - **!** Si la commande n'est pas trouvée, référez vous à la partie configuration avancée pour configurer le PATH

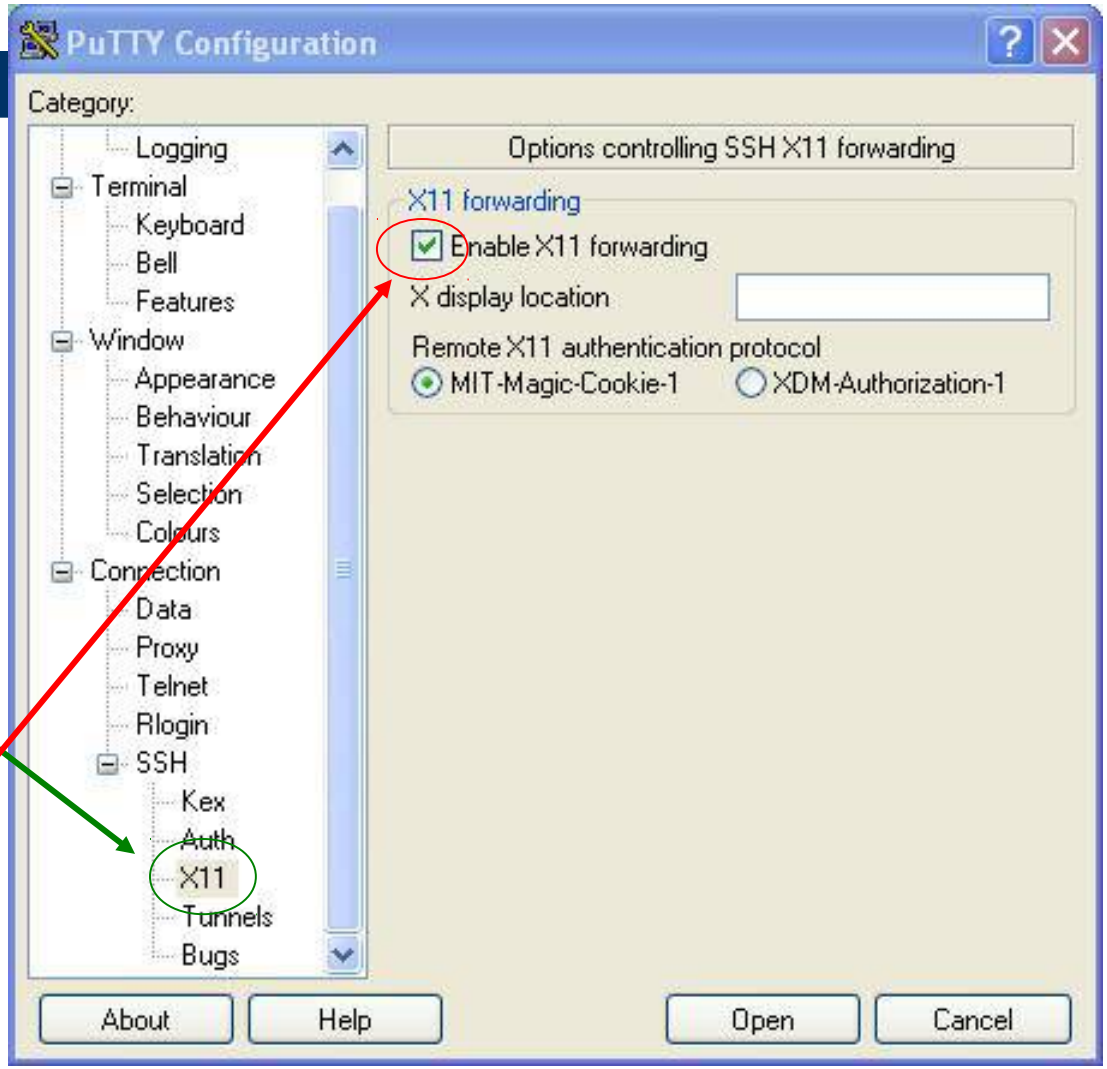
Côté client : PuTTY - profiles

- SSH et Windows : PuTTY
  - Configuration avancée
    - Rattacher les options sélectionnées à un profile d'utilisateur
      - Donner un nom au profile
      - Puis sauver : ***save***
    - Utiliser le profile
      - le sélectioner
      - Puis le charger ***load***



côté client : ssh et X11

- SSH et les applications graphiques Unix :
  - Avec le client Windows, il faut spécifier le demande de redirection du traffic X11 Unix dans la session ssh
  - ***Option : X11 Forwarding*** dans PuTTY :
    - Lancer ***putty***
    - Menu ***SSH → Tunnels***
    - Cocher la case : ***Enable X11 forwarding***
  - N'oubliez pas sous Windows, il faut aussi lancer un émulateur X11 (Xming)



Authentification forte (AF)

- SSH et l’authentification forte …

Authentification forte

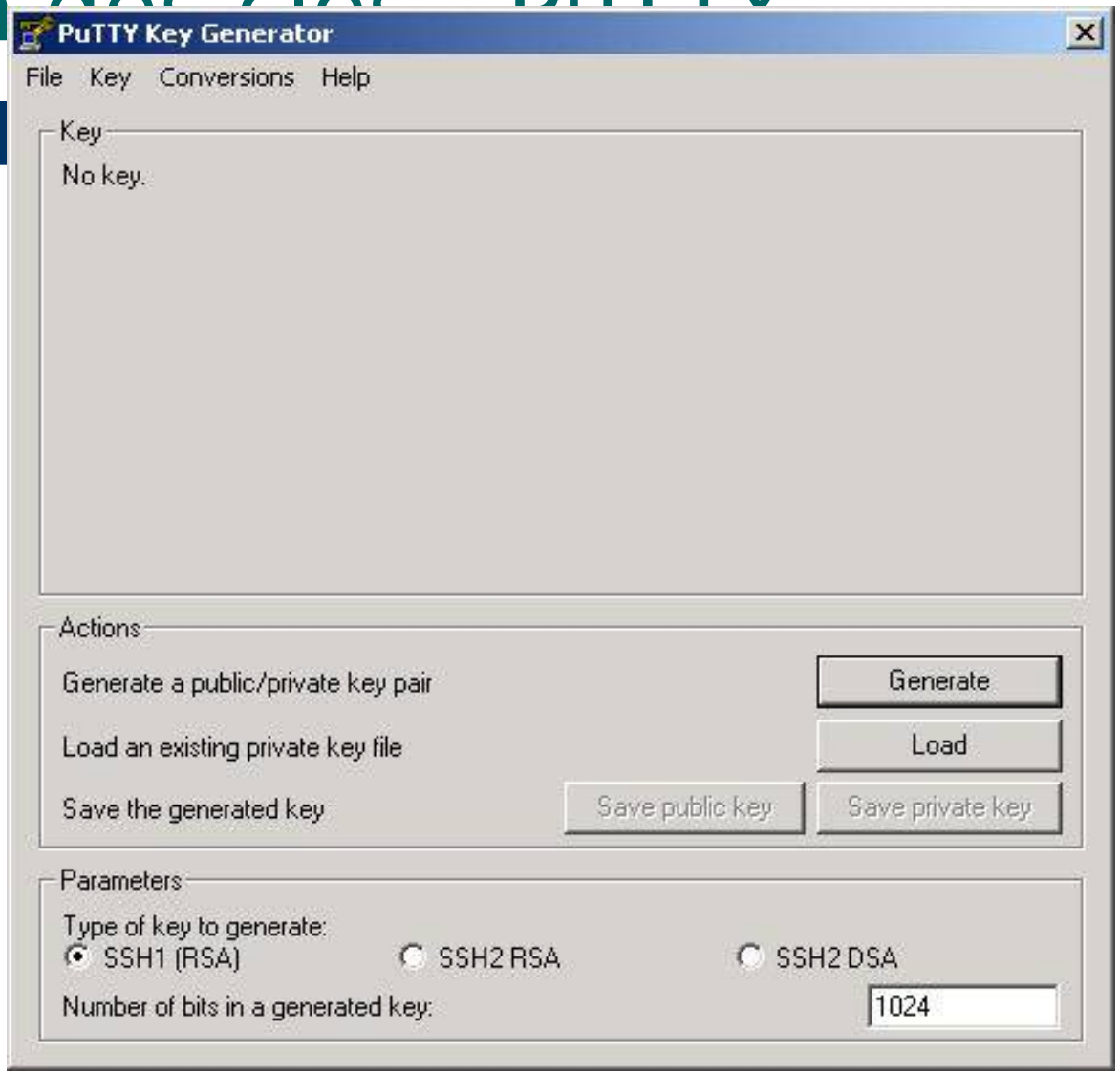
- Connexion par authentification forte
  - Basée sur une procédure d'identification plus complexe et différente de celle des systèmes Unix classiques *(nom et mot de passe)*
  - Cette procédure repose sur un principe d'une paire de clés publique/privée dont la clé privée est protégée par une phrase d'identification
    - **ATTENTION !** la sécurité de *ssh par authentification forte* repose alors sur la protection de la clé privée : il faut **impérativement** mettre une VRAI PHRASE D'AUTHENTIFICATION (au moins 14 caractères)
    - qui est en fait plus qu'un simple mot de passe, mais une véritable phrase (moins de limitation)
      - Permet l'utilisation de caractères blancs (séparateur) et d'autres, mais attention aux caractères utilisés à cause des différents types de claviers afin de pouvoir taper la passe-phrase
    - L'utilisateur s'identifiera alors sans utiliser le mot de passe de la connexion classique *(mot de passe Unix)*, qui lui circule sur le réseau, mais par sa phrase d'identification sur l'hôte local

Authentification forte

- Connexion par authentification forte
  - Utilise un algorithme très puissant pour le chiffrement *(algorithme RSA/DSA)*
  - Ainsi chaque utilisateur possède son propre jeu de clés uniques (une clé privée = secrète, une clé publique = accessible par tous)
  - une nouvelle connexion nécessite l'installation de la clé privée sur le client et de la clé publique sur le serveur
  - le serveur va créer un *challenge* et donner un accès au client si ce dernier parvient à déchiffrer le challenge avec sa clé privée

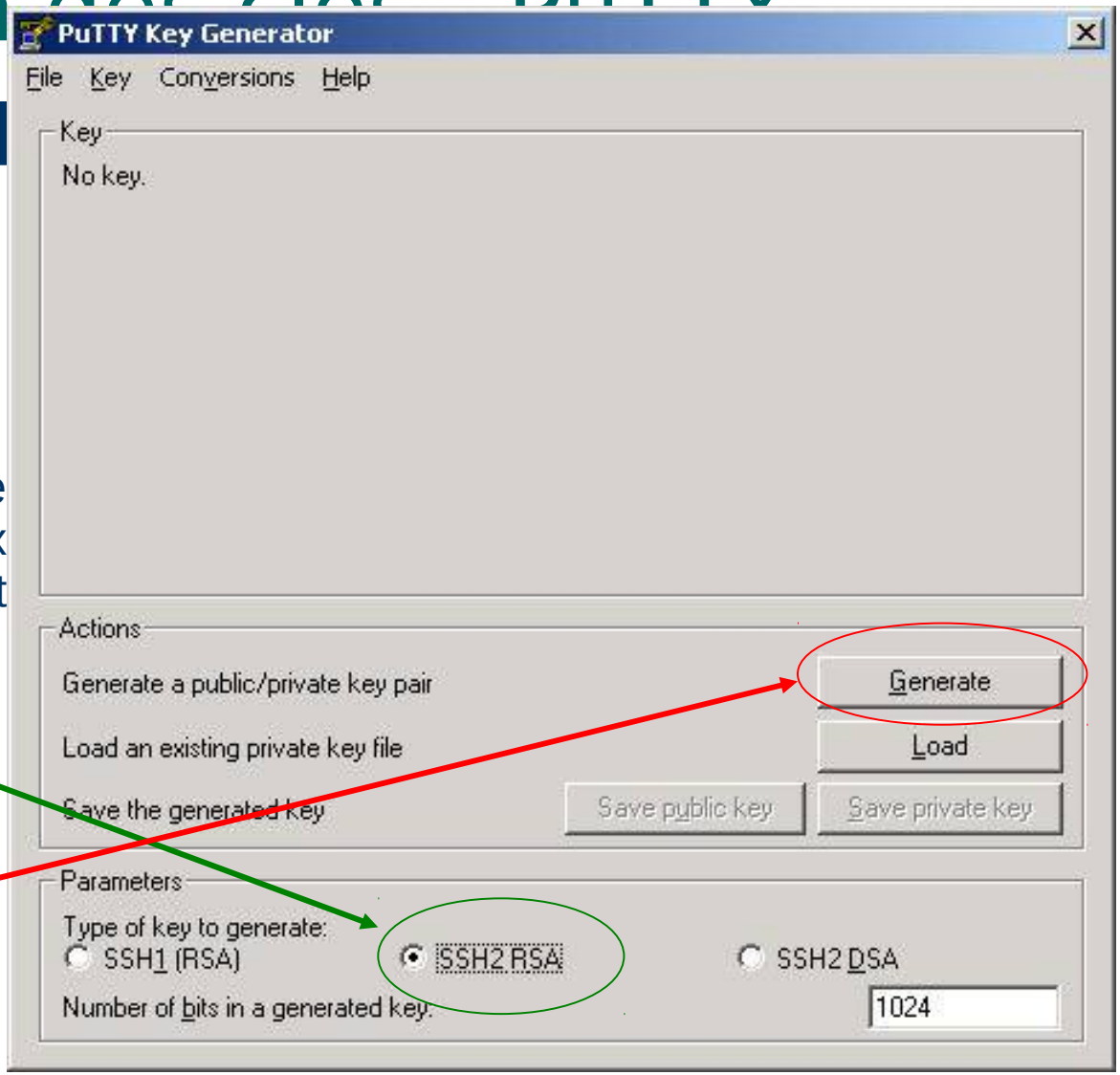
AF : gestion des clés - PuTTY

- Gestion des clés et agents: PuTTY
  - Création des paires de clés avec PuTTY
    - Lancer le programme *puttygen*
      - en double-cliquant sur l'icône :



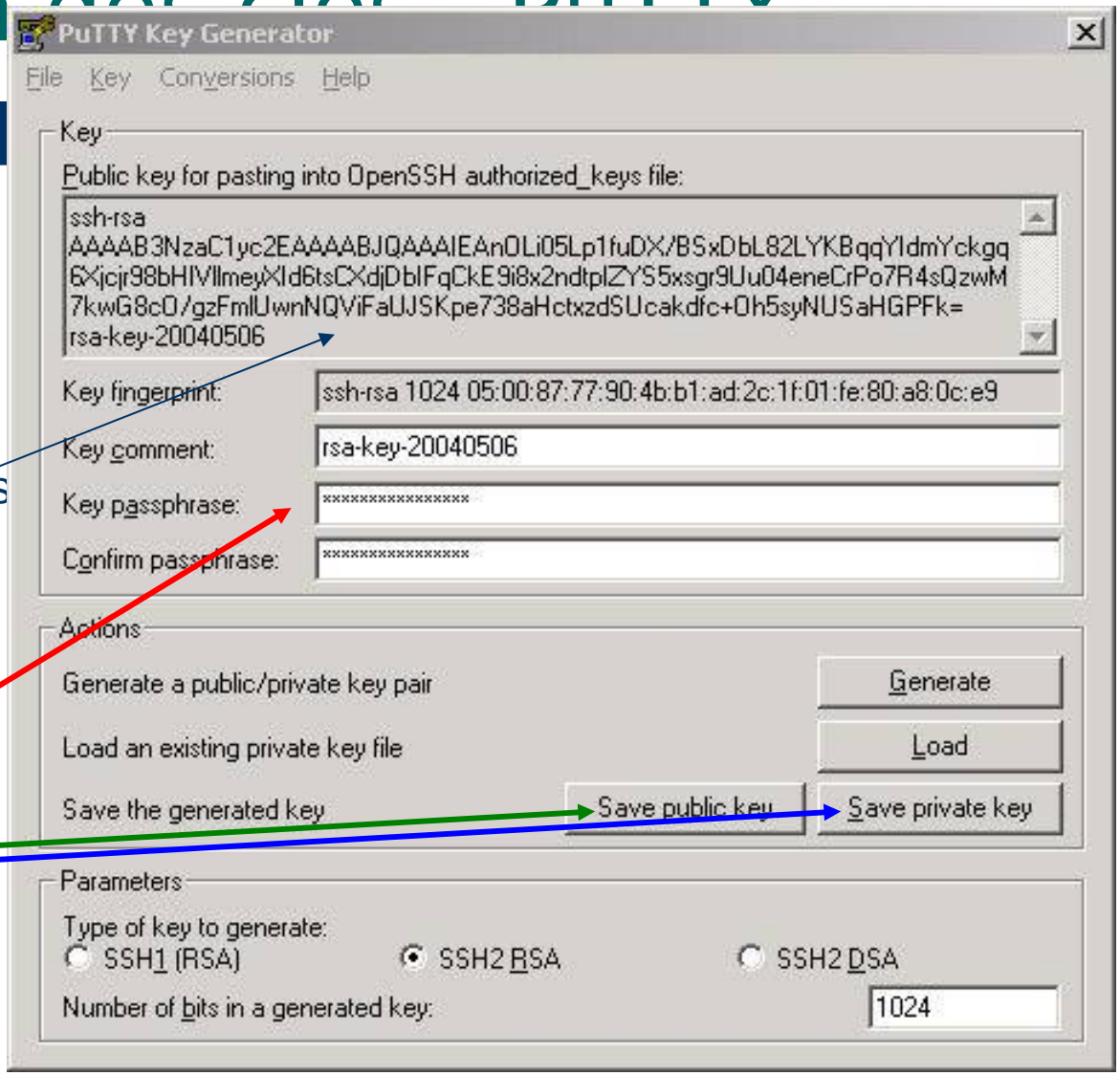
AF : gestion des clés - PuTTY

- Gestion des clés et agents: PuTTY
  - Création des paires de clés avec PuTTY
    - Sélectionner le type de clés à créer (sous Unix le plus souvent ce sont les clés v2 RSA qui sont utilisées)
    - Lancer la génération du couple de clés (publique/privée)



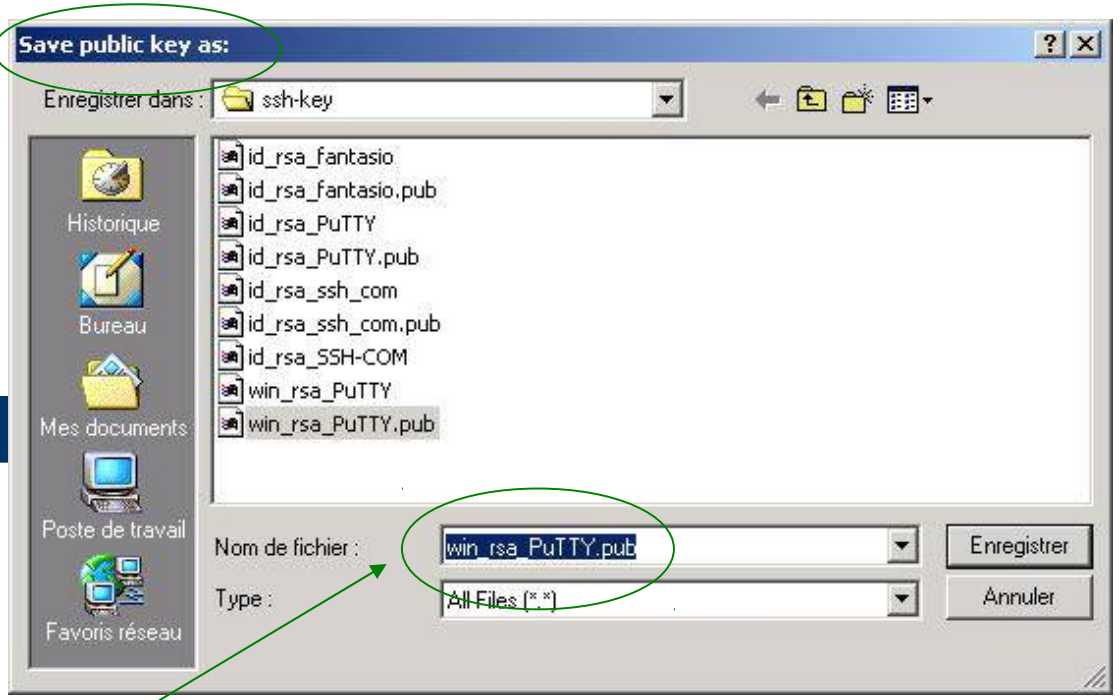
AF : gestion des clés - PuTTY

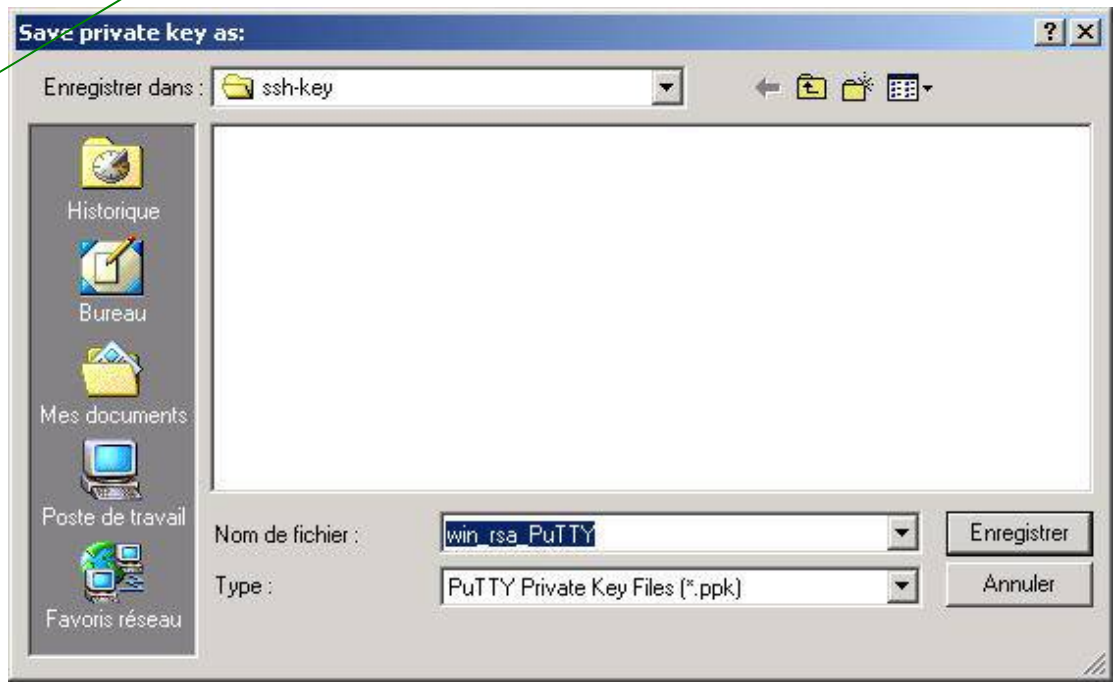
- Gestion des clés et agents: PuTTY
  - Création des paires de clés avec PuTTY
    - Les clés sont générées
    - Donner une phrase d'authentification (au moins 14 caractères minimum)
    - Sauver les clés dans 2 fichiers



AF : gestion

- Gestion des clés et agents: PuTTY
  - Création des paires de clés avec PuTTY
    - Sauvegardes des clés
      - Donner le même nom (comme sous Unix)
      - Clé publique :
        Ajouter ***.pub*** au nom du fichier pour bien la repérer
      - Clé privée :
        Même nom sans extension .pub





AF : gestion des clés - distribution

- Distribution des clés
  - Problématique de l'installation de la clé privée (id_rsa.pub) sur le serveur distant dans le fichier des clés (authorized_keys) :
    - Soit transférer la clé publique
      - Or, Jispose de la clé publique sur la machine locale
      - Il faut la transférer et copier son contenu dans un fichier nommé authorized_keys sur la machine distante
      - Ce fichier (authorized_keys) sur la machine distante va donc pouvoir contenir plusieurs clés publiques d'utilisateurs
    - Soit envoyer le fichier (clé publique) à l'administrateur du serveur distant qui l'installera sur la machine

AF : gestion des clés - distribution

- Mise en place des clés : méthode manuelle
  - La clé privée côté client : *identity* ou *id_rsa (id_dsa)*
  - La clé publique sur le serveur dans le fichier :

    authorized_keys = clés publiques situées dans ***$HOME/.ssh***

client

serveur

$HOME/.ssh/id_rsa.pub

Transférer la clé publique sur la machine distante dans $HOME/.ssh

Puis dans $HOME/.ssh/

Faire :

cat id_rsa.pub >> authorized_keys

chmod 600 authorized_keys

AF : gestion des clés - distribution

- Mise en place des clés : méthode automatique
  - Utilisation d'un script shell : ssh-copy-id
    - installe la clé publique dans la liste des clefs autorisées (authorized_keys) d'une machine distante

client

serveur

 # ssh-copy-id *serveur*

Transférer la clé publique sur la machine distante dans $HOME/.ssh et le fichier authorized_keys directement

Rien à faire sur le serveur

AF : gestion des clés - distribution

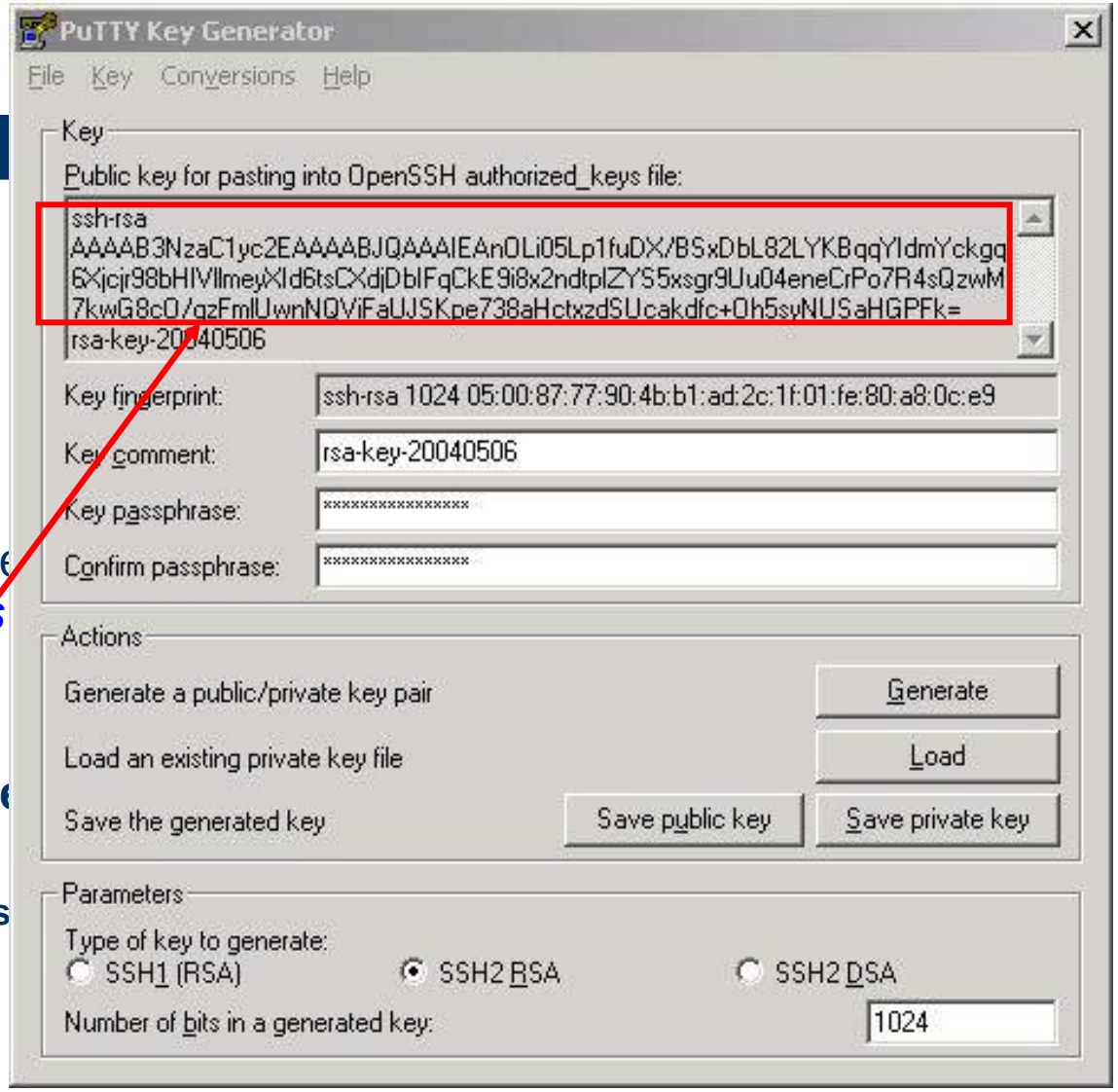
- Mise en place des clés PuTTY
  - La clé publique sur le serveur distant :
    - ***Copier/coller*** du cadre rouge sur **1 ligne** dans le fichier ***authorized_keys*** la machine distante

  (ssh-rsa AAAB3…..k=)

  Vérification de cohérence

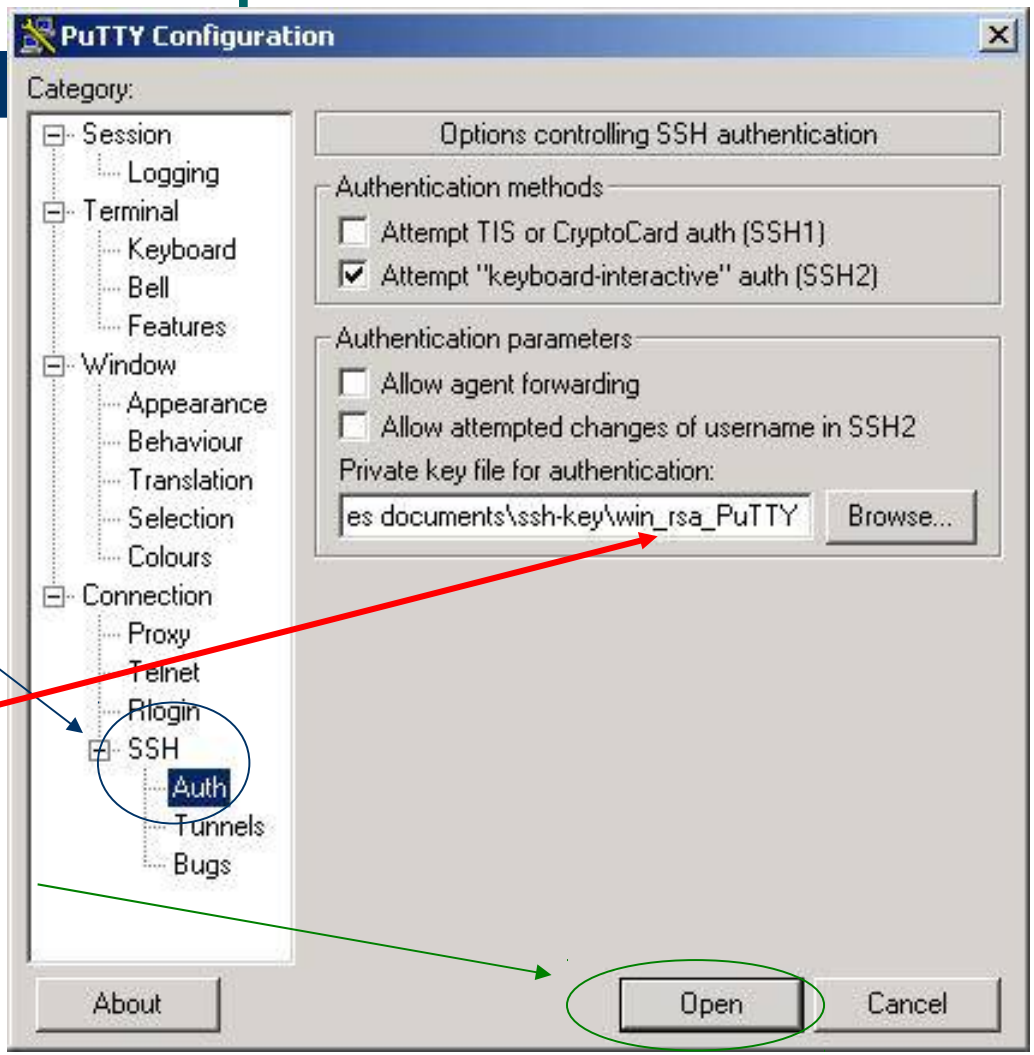
  Sur linux :

  ssh –l –f ~/.ssh/authorized_keys



AF : connexion simple Windows

- Connexions par authentification forte
  - Par ssh PuTTY:
    - Lancer ***putty***
    - Dans la variable ***SSH*** → ***Auth***,
    - charger la clé privée créée par PuTTY
    - Puis cliquer sur ***open***

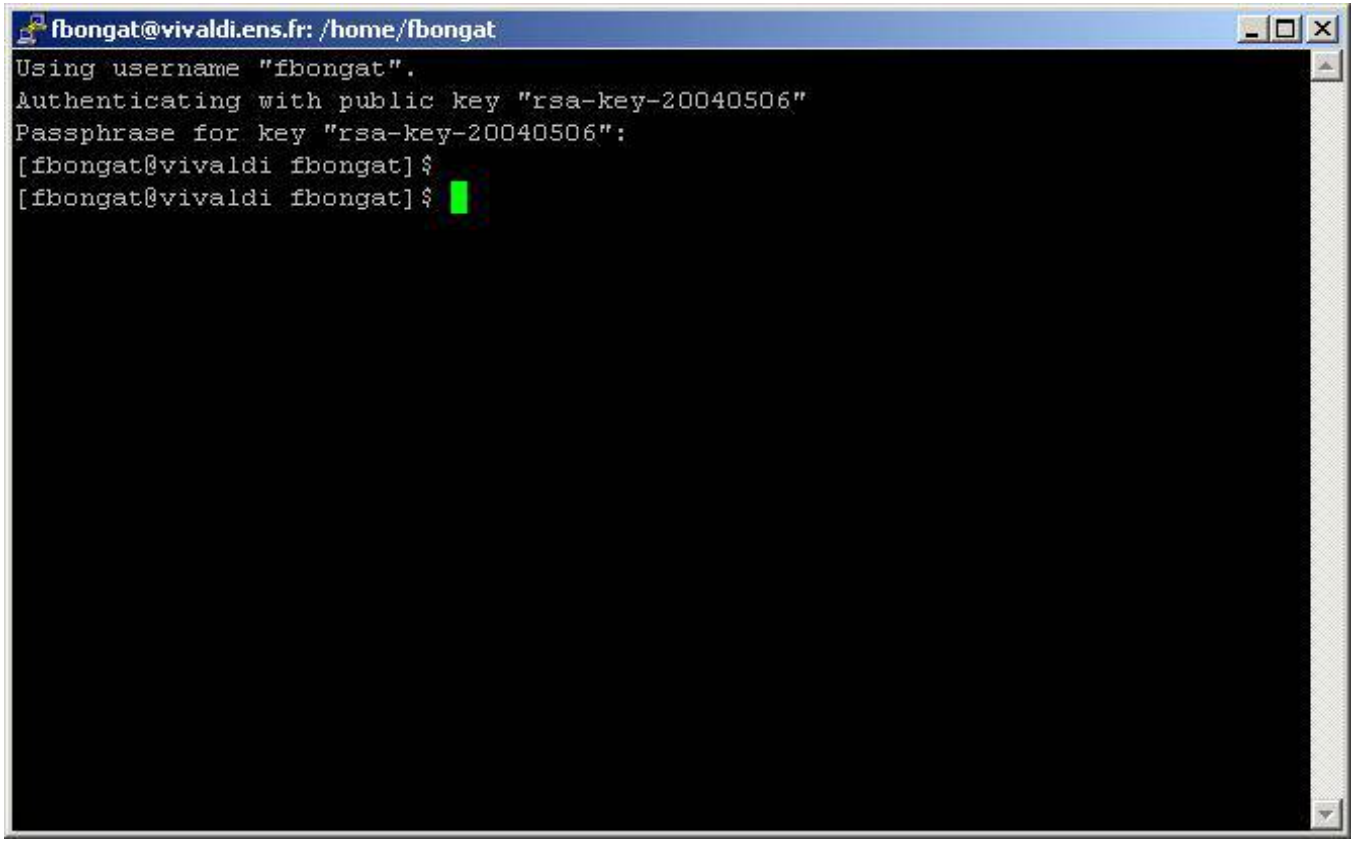


AF : connexion simple Windows

- Connexions par authentification forte

  par ssh PuTTY:

  Idem avec ***psftp*** et ***pscp***



AF : ssh agent – session graphique

- **Keychain** : gestion des clés et des agents
  - Développé par Gentoo Linux
    - Existe pour toutes les distributions (sous forme de package)
  - Ajout de l'AF via les scripts shell de démarrage linux
  - Le programme keychain permet de réutiliser les instances de ssh-agent + ssh-add dans des sessions différentes et, si désiré, d'inviter l'utilisateur à entrer les phrases de passe à chaque ouverture de session.
    - Package à installer sur linux *keychain*
    - Lancé dans les scripts d'initialisation shell
      - */etc/profile.d/keychain.sh*

AF : ssh agent – session Windows

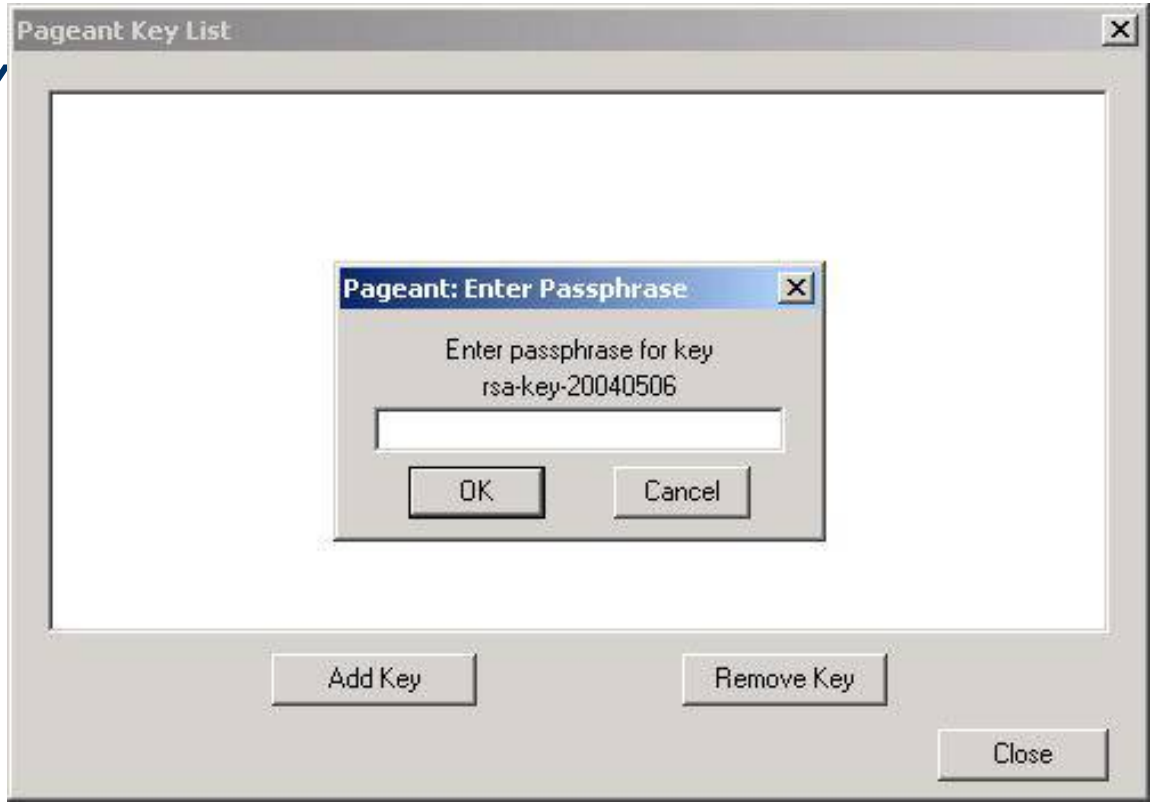
- Gestion des clés et agents avec PuTTY
  - Lancer l'agent en double cliquant sur l'icône ***pageant*** :

  - Il apparaît alors dans la barre des tâches actives :

  - Cliquer avec le ***bouton droit de la souris*** pour faire apparaître le menu contextuel du pageant et cliquez ensuite sur « ***Add Key*** » pour ajouter les clés privées gérées par l'agent ssh

AF : ssh agent – session Windows

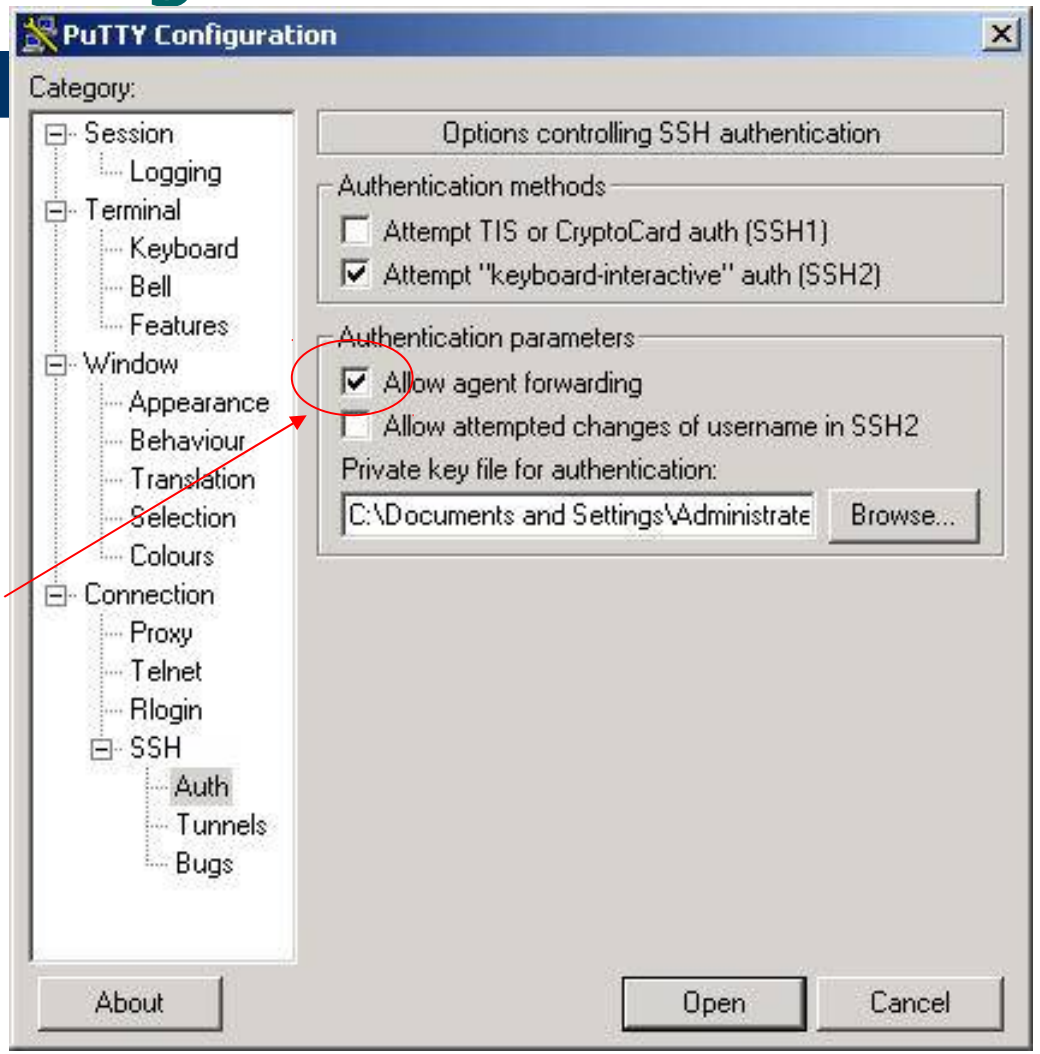
- Gestion des clés et agents avec PuTTY
  - Cliquer ***AddKey***
  - Charger la clé privée Putty
  - Entrer la phrase d'authentification



AF : transfert d'agent - Windows

- Transfert d'agent
  - relais des demandes d'authentification entre hôte à partir d'un client PuTTY
    - Cocher la case :

      Allow agent forwarding



AF : batch ou AF sans mot de passe

- Mode « Hostbased »
  - C'est une authentification par hôte de confiance
    - On utilise une paire de clés publique et privée pour établir une connexion sécurisée et de confiance
      - Il faut cependant faire attention à ne pas se faire voler sa clé privée ou sa machine
    - Donne l'accès sans passphrase à tous les utilisateurs d'une machine de confiance vers une autre.
    - Utilisation pour le batch notamment pour les fermes de calcul
      - *Attention aux accès à la frontale --> problème de sécurité ensuite*

Tunneling : e

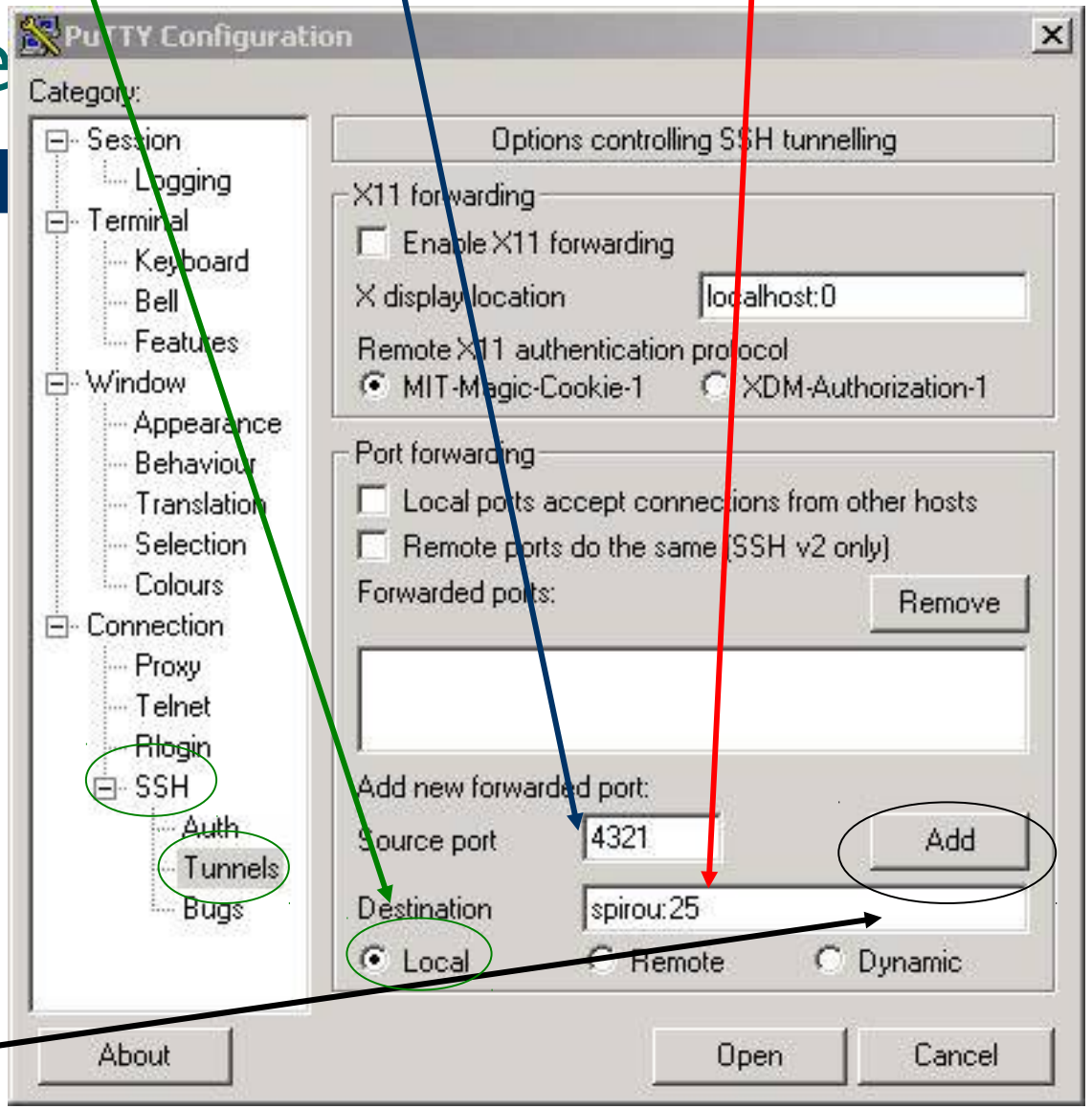
- PuTTY et smtp
  - Envoi d'un message vers un relai smtp filtré via une passerelle ssh
    - Lancer putty
    - Aller dans la valeur ***SSH*** → ***Tunnels***
    - Remplir les champs liés au tunneling dans la fenêtre
    - Puis valider cliquant sur ***Add***

Le port local (> 1024), ici : 4321

Connexion sur localhost

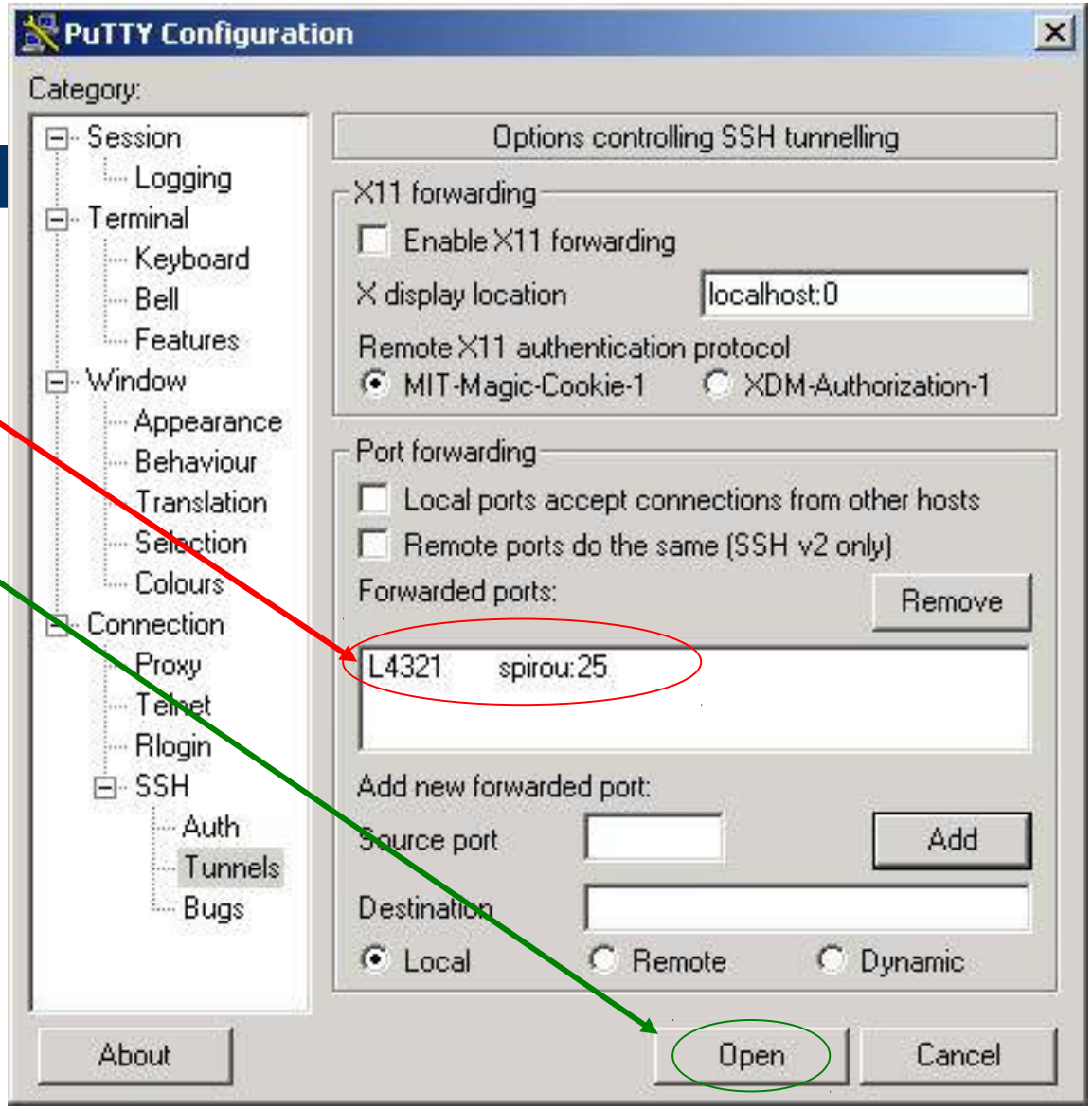
Nom de la machine distante et le port distant

Syntaxe : spirou:25



Tunneling : e

- PuTTY et smtp
  - Lancement du tunnelssh :
    - Le tunnel est configuré
    - Il faut maintenant lancer la connexion « ***Open*** »
    - Donner son mot de passe afin d'obtenir une connexion ssh classique
    - Le tunnel sera ainsi prêt
    - Configurer l'application pour qu'elle se connecte sur :
      - ***Nom distant*** : ***localhost***
      - ***Port*** : ***4321***



Tunneling : envoi de messages smtp

- PuTTY et smtp
  - Exemple smtp : simulation d'un client de messagerie
    - Configurer l'application pour qu'elle se connecte sur :
      - ***serveur smtp sortant*** **:** ***localhost***
      - ***port*** **:** ***4321***

Exemple en simulant une connexion d'un client de mail par un *telnet* en localhost sur le port 4321

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrateur>telnet localhost 4321_
```

Le serveur répond enfin ! On peux envoyer des messages depuis ce serveur bien qu'il soit filtré

```
220 mailhost.bdnet ESMTP Postfix
_
```

Conclusion

- SSH est une boîte à outils complète
- Les connexions sont sécurisées
  - Améliore la sécurité mais il permet aussi de la contourner
- Existe différentes possibilités de connexions et de transferts de fichiers
- Les relais possibles des applications TCP permettent d'accéder à de nouvelles ressources
- Installé en standard sur tous les systèmes (client et serveur) Unix et MacOS X
- De très bons clients Windows notamment PuTTY, Filezilla notamment en les associant à Xming