
WIPO

WIPO PCT ELECTRONIC DATA INTERCHANGE SERVICE
FUNDAMENTALS
(PCT-EDI)

Version 1.2



**WORLD INTELLECTUAL PROPERTY
ORGANIZATION**

GENEVA

Document Information

Author(s)	J Fullton
-----------	-----------

Revision History

<i>Version</i>	<i>Date</i>	<i>Remarks</i>	<i>Distributed to & Reviewed by</i>
Draft 0.1	May 2004	Document start	Outline plus initial draft for author review
Draft 0.2	May 2004	Initial review	V Gross P Waring
Draft 0.3	May 2004	Internal review	V Gross P Waring J Fullton
1.0	June 2004	Reformatted	V Gross P Waring
1.1	February 2008	Internal review actions	P Waring
1.2	January 2009	Initial Connection Change	P Waring T Song

Document Reviewers

<i>Name</i>	<i>Title</i>
P.Waring	
V Gross	
T Song	
M Leach	

Document Approvals

<i>Name</i>	<i>Title</i>

Document References

<i>Reference</i>	<i>Document Name</i>	<i>Release</i>

TABLE OF CONTENTS

1	INTRODUCTION	5
2	SERVICE OVERVIEW	5
2.1	Why SSH?	6
2.2	The WIPO Server	6
2.3	Client Software	6
3	OFFICE ACCOUNTS	8
3.1	Introduction	8
3.2	Directory Structure	8
3.3	Log files	8
3.4	Maintenance	8
3.5	Examples in this Document	9
4	CLIENT SOFTWARE CONFIGURATION	10
4.1	Introduction	10
4.2	Microsoft Win 32 Environment	10
4.2.1	Setting Up and Using WinSCP	10
4.2.2	Setting Up and Using PuTTY	15
4.2.3	Setting Up and Using the Tectia SSH Client	21
4.3	Unix Environment	32
4.3.1	Installing and Using the OpenSSH Utilities	34
4.3.2	Installing and Using lftp	38
5	ADVANCED TOPICS	41
5.1	Introduction	41
5.2	Adding and Revoking Users	41
5.3	Developing Customized Applications	47
5.4	Example Perl Source Code	48

1 INTRODUCTION

The WIPO PCT Electronic Data Interchange Service (PCT-EDI) provides a flexible, secure mechanism for the exchange of intellectual property information between Offices and the International Bureau (IB). This includes Priority Documents (P-DOCS), and pamphlets, as well as other general-purpose bulk data products. This system is one of a trio of services associated with the WIPO COR environment:

- **Paper-Based Requests and Submissions** - documents are delivered to or requested from the IB via postal mail,
- **Online COR Retrieval** – requests for documents are entered one by one using an online Web interface or by uploading orders in XML. Documents may then be viewed or ordered and received in the chosen format,
- **PCT-EDI Service Order** – documents are requested using a bulk order in XML and results are retrieved in bulk. Priority documents may be submitted in an accepted wrapper. These operations are supported through communications protocols described below.

2 SERVICE OVERVIEW

An important goal in the development of this service is the provision of low-or-no cost software for participating Offices. The IB has elected to support the Secure Shell suite of protocols and services, as it is widely supported in both free and commercial versions. Software packages in each of these categories are discussed below.

From a technical perspective, the Secure Shell Protocol is described in the Secure Shell [secsh] IETF Charter¹ (more commonly known as SSH²). Specifically, the preliminary release of the service uses services defined in the SSH File Transfer Protocol Draft.

The XML data structures used for the request and submission of documents are described in other WIPO documents.

¹ <http://www.ietf.org/html.charters/secsh-charter.html>

² <http://www.ietf.org/internet-drafts/draft-ietf-secsh-filexfer-05.txt>

2.1 WHY SSH?

SSH provides a secure means of reliably transferring data over communications networks via the Secure Shell File Transfer Protocol (SFTP). Unlike the standard Internet File Transfer Protocol (FTP), user authentication information is never transmitted in the clear. All communications are encrypted using a user-selected data encryption algorithm. Unlike Transport Layer Security (TLS) enhanced FTP, only a single port is used for bi-directional communications, eliminating some rather difficult firewall and security configuration issues. Unlike Hypertext Transfer Protocol (HTTP), SFTP provides asynchronous, block-oriented data transfer verification, including the facility to restart interrupted transmissions from the point of interruption. Unlike Simple Object Access Protocol (SOAP), no independent software development is required. It should be noted, however, that Offices desiring to develop customized SSH/SFTP applications for integration with existing systems may do so.

SSH is available in many different implementations, both commercial and freely available. Numerous clients with excellent bulk-transfer functionality are available from a number of sources, thus significantly reducing software development and maintenance costs for national Offices and the IB. However, this does not affect the flexibility offered by the service; in fact, several free³ and commercial development toolkits are available for those Offices wishing to integrate automated priority document access into their existing examination systems and processes.

The modularity of the PCT-EDI service is such that other access mechanisms and protocols can be easily added if needed. These include the protocols mentioned above (TLS-FTP, SOAP, HTTP, etc.), as well as file system oriented services such as samba.

2.2 THE WIPO SERVER

The IB uses the SSH Communications Security Corporation Tectia SSH2 server for Unix. This server supports the full range of SSH2 services, including public key authentication using both digital certificates and the standard SSH key trust model.

2.3 CLIENT SOFTWARE

Several clients are available for use with the PCT EDI. Two freely available Windows Win32 environment clients, PuTTY⁴ and WinSCP⁵, are described below. Both clients use the PuTTYgen key generator package. PuTTY is a simple client that is useful for automated tasks initiated from a Windows machine. WinSCP is a user-friendly full-featured client designed for use by an operator. It is suggested that first-time users become familiar with the WinSCP client.

³ The OpenSSH toolkit is available from <http://www.openssh.org/>. Perl modules for SSH and SFTP are available from <http://search.cpan.org/~drotsky/Net-SSH-Perl-1.25/lib/Net/SSH/Perl.pm> and <http://search.cpan.org/~drotsky/Net-SFTP-0.08/lib/Net/SFTP.pm>, respectively.

⁴ <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

⁵ <http://winscp.sourceforge.net/>

A tested commercial product is the SSH Tectia⁶ client for several platforms, including Microsoft Windows/NT/2000. This client also supports authentication through the use of industry-standard digital certificates.

Offices working in the Unix environment will find detailed instructions for the use of the OpenSSH secure shell package, including the *ssh* and *sftp* clients, and the *lftp* advanced SFTPclient, which includes features such as site mirroring and batch download/upload scheduler.

⁶ <http://www.ssh.com>

3 OFFICE ACCOUNTS

3.1 INTRODUCTION

There are a variety of services available to the more advanced user, including data directory mirroring, the addition of additional authorized users to an Office account, and the development of customized access applications for Unix and Microsoft Win32 platforms. These features are discussed in later sections.

3.2 OFFICE ACCOUNT AND DIRECTORY STRUCTURE

Each participating Office is assigned a unique account. Office accounts have a predefined directory structure as established by the IB:

- **Download** – the directory where the IB systems place documents systematically sent to, and ordered documents, from COR, requested by the Office,
- **Upload** – the directory where the Office places document packages and other agreed files (National Phase Entry Information, Fees) for the IB,
- **Request** – the directory where the Offices places XML requests for PCT documents to be processed by the COR system,
- **Feedback** – the directory where the COR system places information concerning the availability of documents requested contained in an order file.

3.3 LOG FILES

3.4 MAINTENANCE

Offices are required to perform certain basic housekeeping activities with respect to their accounts. Offices should delete old files from their “Download” and “Upload” directories on a daily basis. Files more than 2 weeks old will be deleted automatically by the system. Users are given a warning each day of files due to be soon deleted, and deleted files are maintained in a running list. These lists are automatically placed in the home directory of each Office.

Offices may elect to have multiple users for a single Office account. A local administrator at the Office can control the number and identities of additional users assigned to the Office account. This is discussed in **Advanced Topics**, below. Individuals who will be working as an additional user under an existing Office account should follow steps 1-4 under **Setting up and Using WinSCP** and should mail their public keys (public key only!!) to their designated administrator. The administrator will rename the key as appropriate and install it on the PCT-EDI system.

3.5 EXAMPLES IN THIS DOCUMENT

Throughout this document, we will use the country code “xx” for demonstration purposes. We invite your feedback on the utility of the examples in this document. Comments may be sent to “pct.edi@wipo.int”.

Please also note that Offices are free to select encryption algorithms and key types and strengths for the PCT-EDI cryptographic systems. The terms RSA and DSA are both used in the examples.

4 CLIENT SOFTWARE CONFIGURATION

4.1 INTRODUCTION

Numerous commercial and free application packages for SSH and SFTP exist. This section provides a “hands-on”, point by point reference guide for installing representative packages in the Win32 (Microsoft Windows NT/XP/2000, etc) environment as well as a generic Unix environment.

4.2 MICROSOFT WIN 32 ENVIRONMENT

This section will guide you through the setup of your Office account using two freely available packages, PuTTY and WinSCP, as well as the commercial Tectia SSH2 client.

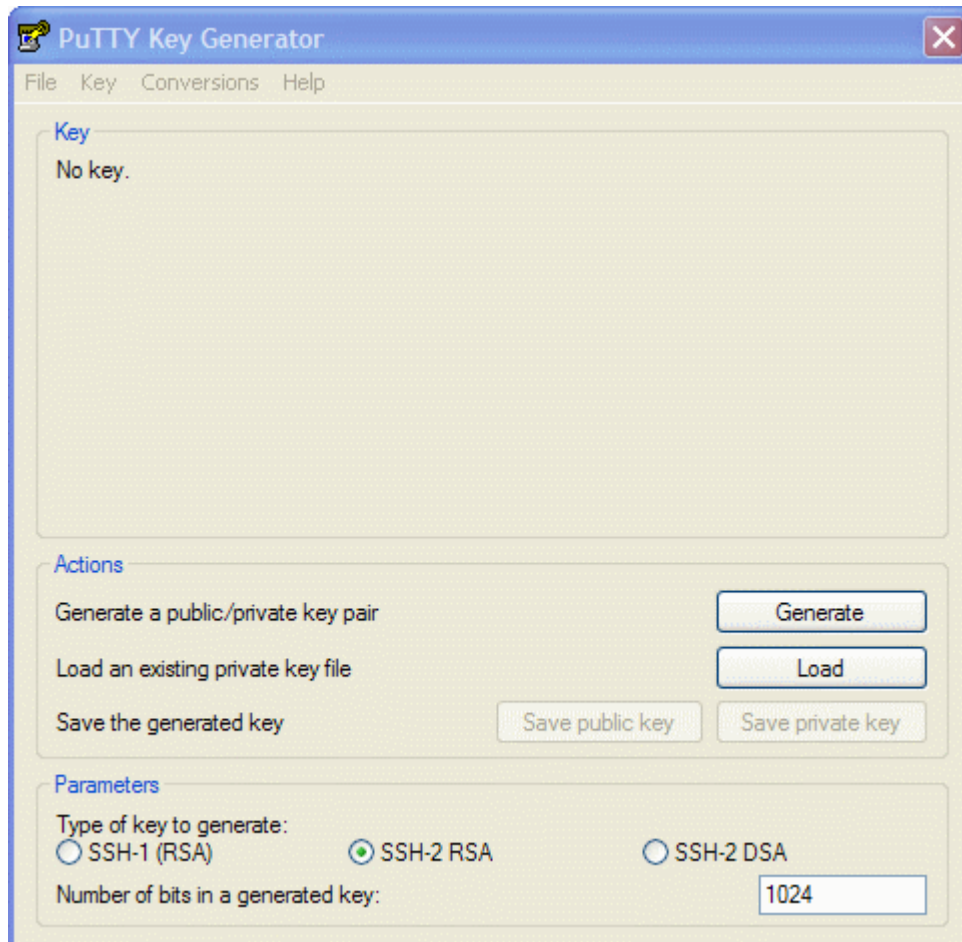
4.2.1 Setting Up and Using WinSCP

1. Install the WinSCP Client. It can be downloaded from:

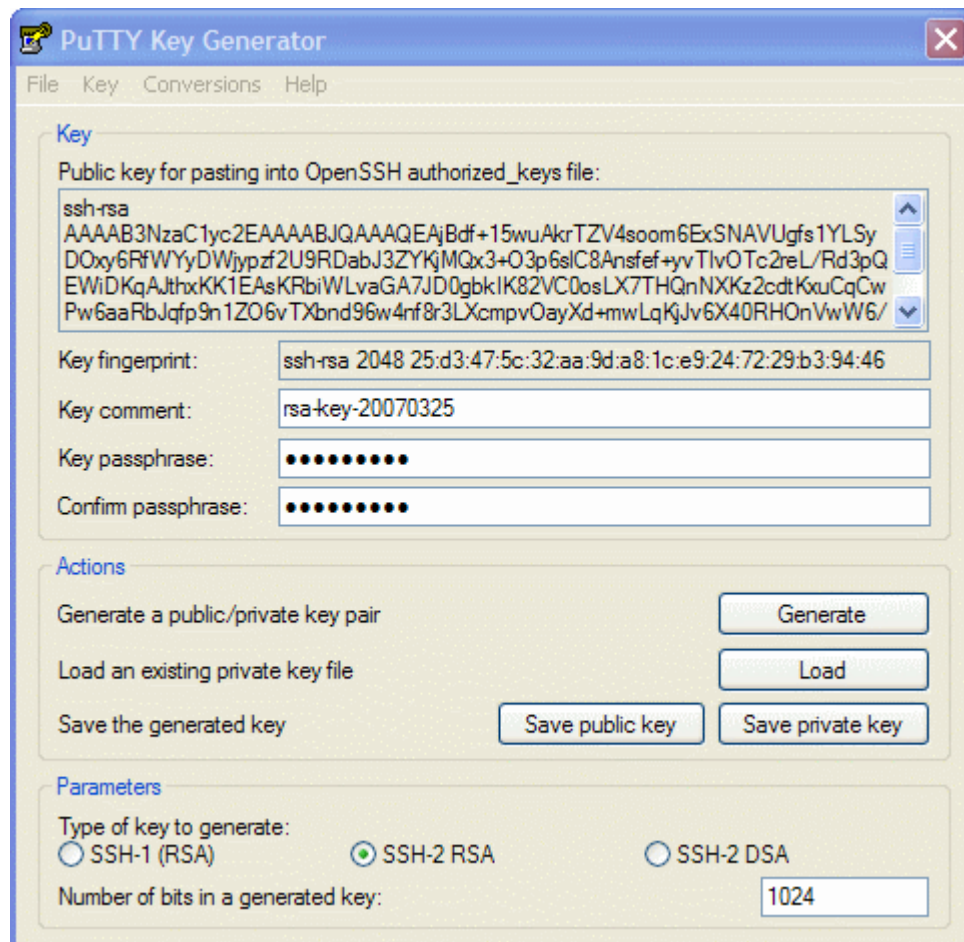
<http://winscp.sourceforge.net/>

After downloading the package, run it. Select “Full Installation” and choose your desired language. Be sure that the boxes concerning the installation of “PuTTYGen” and “Pageant” are checked. Follow the installation instructions. You will need to restart the computer after installation.

2. Start the PuTTYGen application. This application is used to create your authentication keys. Select either “SSH2 RSA” or “SSH2 DSA” for the “Type of key to generate”, with 1024 bit key size or greater. Press the “Generate” button and follow the instructions on the screen.



- When key generation is complete, you will be shown the screen below. Enter a good (at least 8 characters, with letters, numbers and punctuation marks) passphrase in the given blocks. You will be prompted for this passphrase whenever you use this key. The passphrase is never sent to the remote machine.

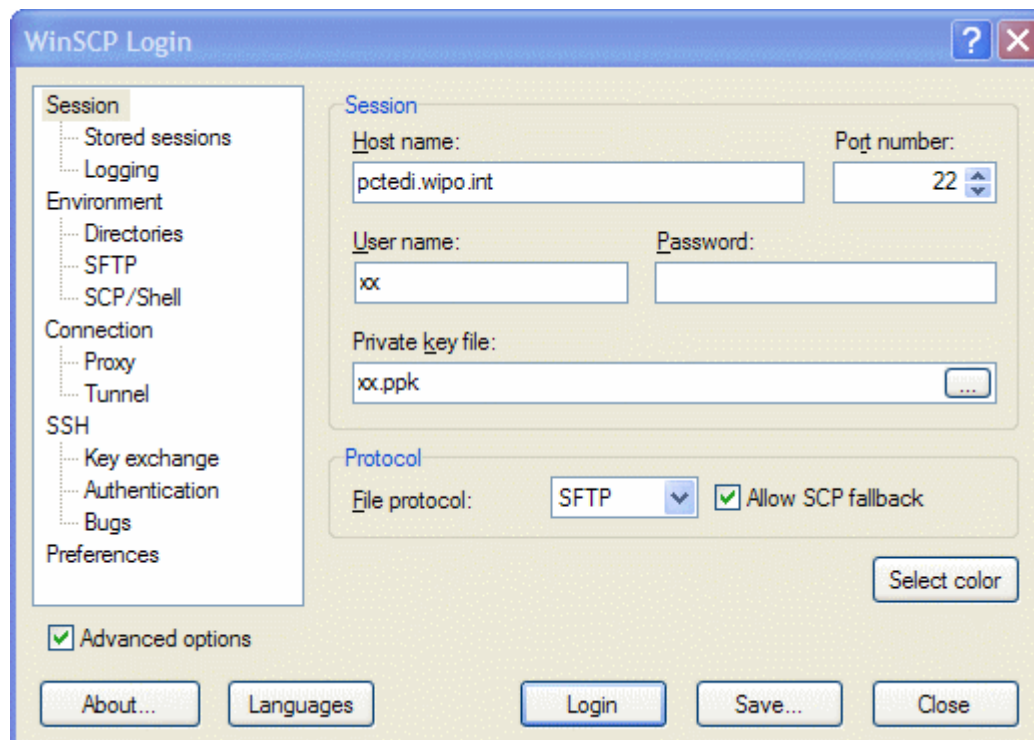


- Press the “Save public key” button to save your public key. Name it using the account name issued by the IB, with the extension “.pub”. Thus, for our example, the public key would be named “xx.pub”. Then, press the “Save private key” button to save your private key. Give it the same name, but without the “.pub” extension, e.g. enter “xx” in the naming box. You have created a 1024 bit SSH2 key using the RSA algorithm. Your public key is named “xx.pub”. Your private key is named “xx.ppk”, where “xx” represents the account name for your country. Write down where you saved these keys! You will need this location later.
- Email your public key to: pct.edi@wipo.int

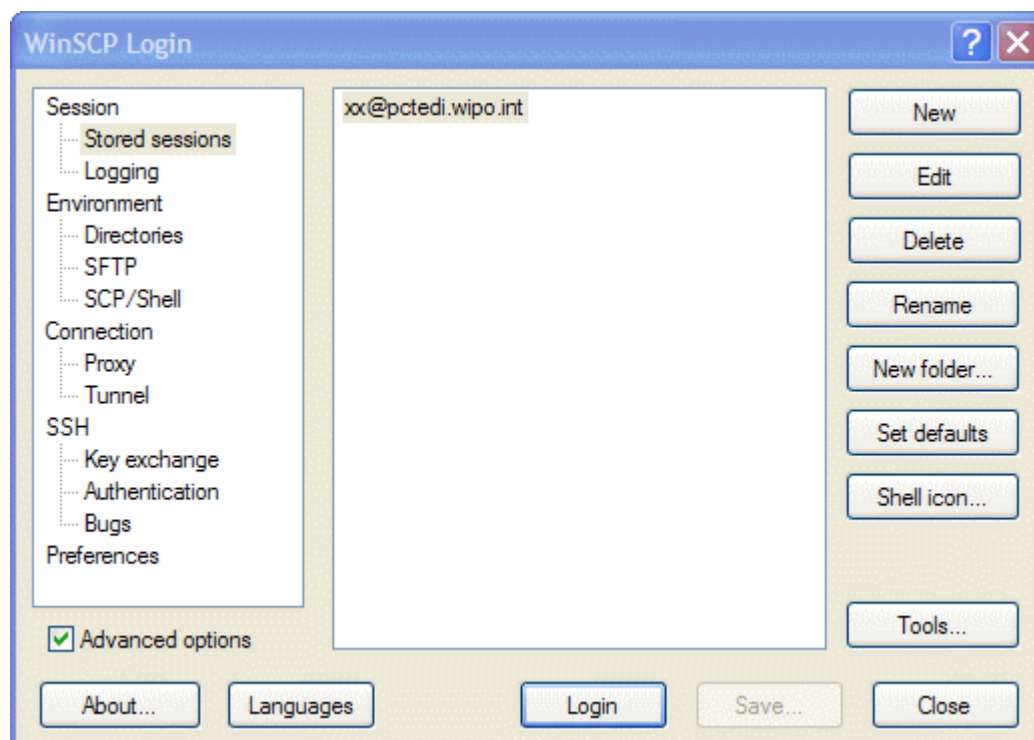
NEVER SEND THE PRIVATE KEY. We will not use any keys where the private key has been transmitted by email.

Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step.

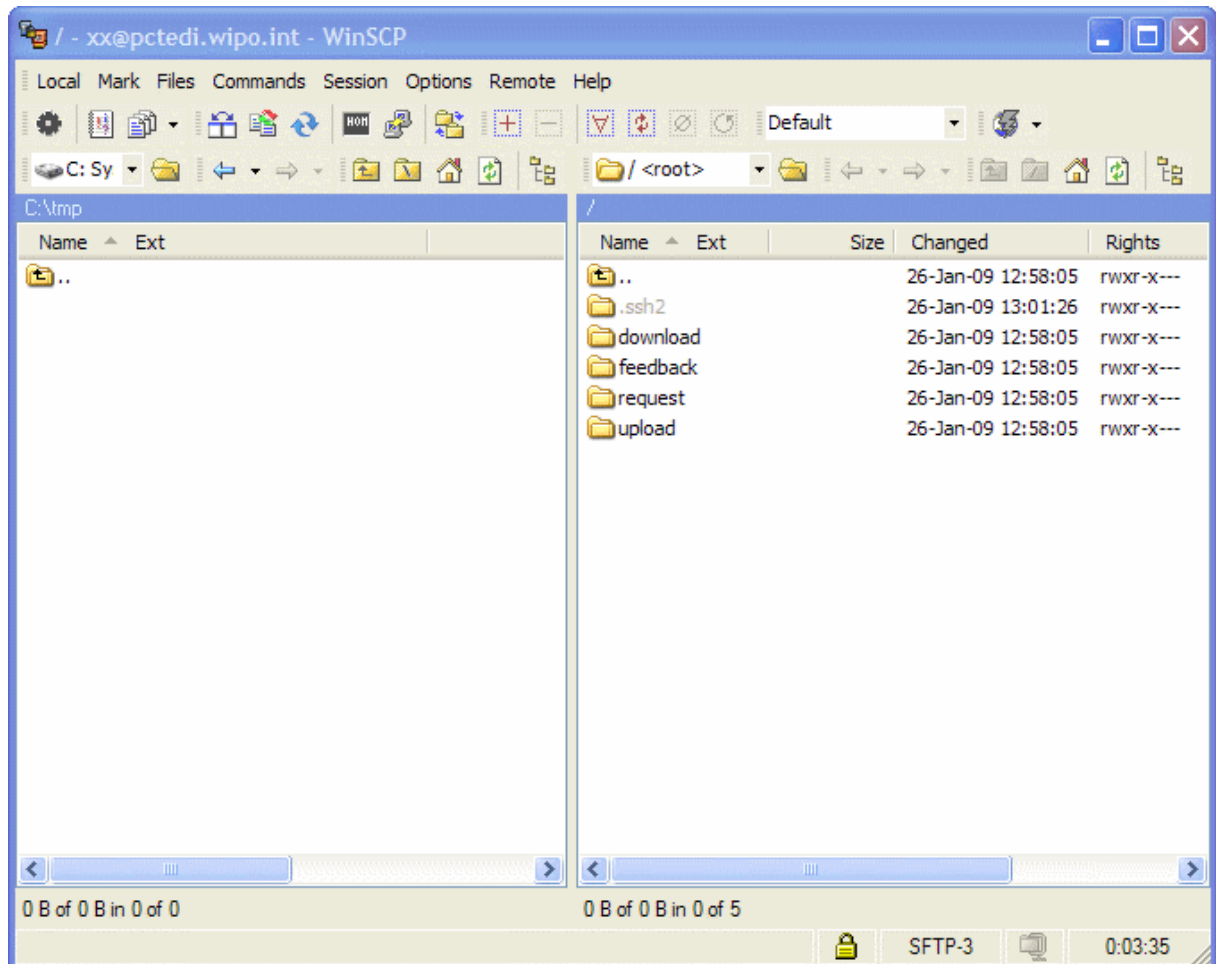
6. Start the WinSCP application and check the “Advanced Options” box



Enter “pctedi.wipo.int” as the host name, and your two-letter country code as the “User name”. Enter the filename for the PRIVATE KEY file you created in step 4. Save the session, naming it as ‘xx@pctedi.wipo.int’.



7. You are now ready to attempt a connection to the system. Press “Login”. You will be asked if you trust this host: Click “Yes”, and then you will be asked for your passphrase (that you input when creating your key-pair).
8. Once you are successfully connected, you will see a split screen with the left-hand side containing a list of files in your local directory, and the right-hand side listing files in the remote directory.

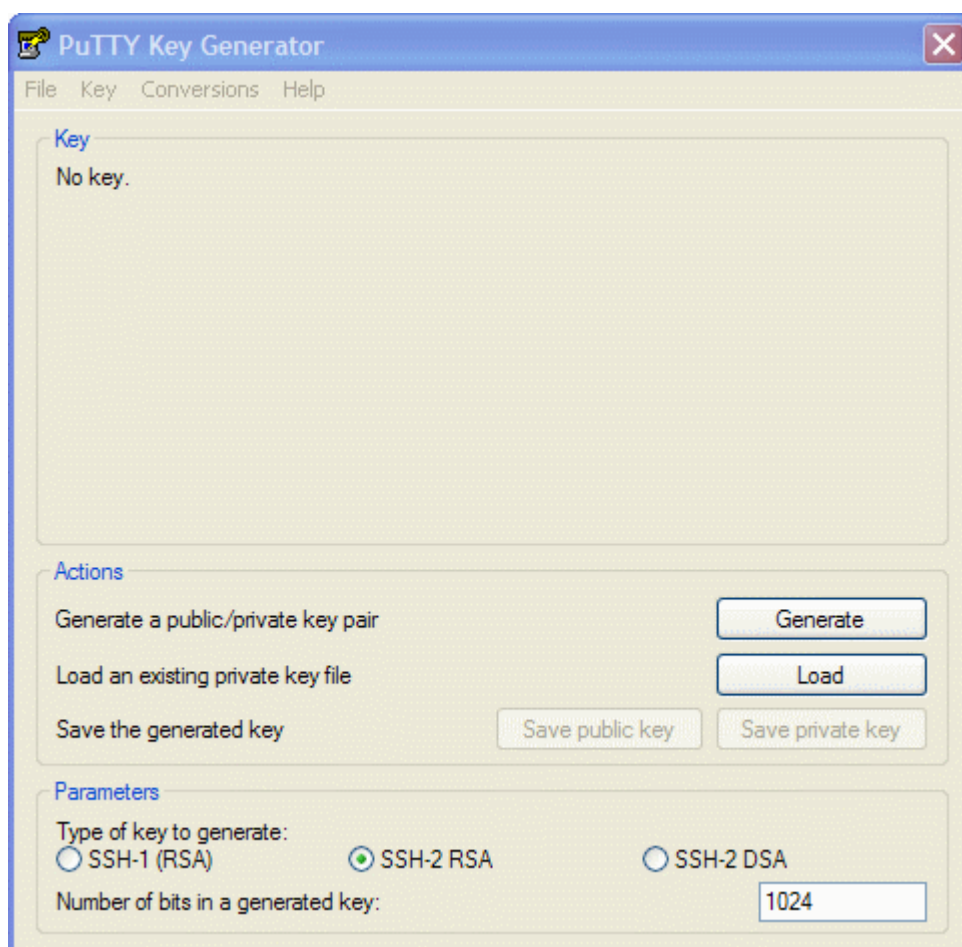


4.2.2 Setting Up and Using PuTTY

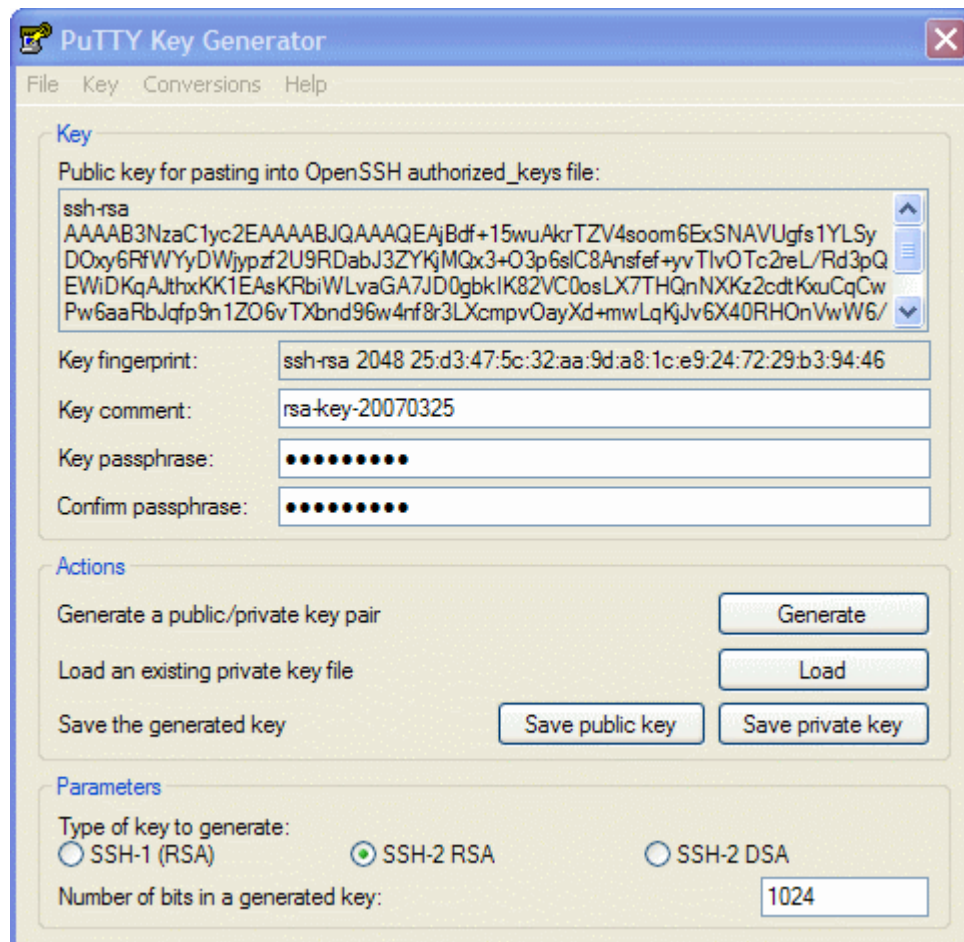
1. Run the installer package and extract the PuTTY components. PuTTY may be downloaded from:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

2. Start the PuTTYGen application. This application is used to create your authentication keys. Select either “SSH2 RSA” or “SSH2 DSA” for the “Type of key to generate”, with 1024 bit key size or greater. Press the “Generate” button and follow the instructions on the screen.



- When key generation is complete, you will be shown the screen below. Enter a good (at least 8 characters, with letters, numbers and punctuation marks) passphrase in the given blocks. You will be prompted for this passphrase whenever you use this key. The passphrase is never sent to the remote machine.

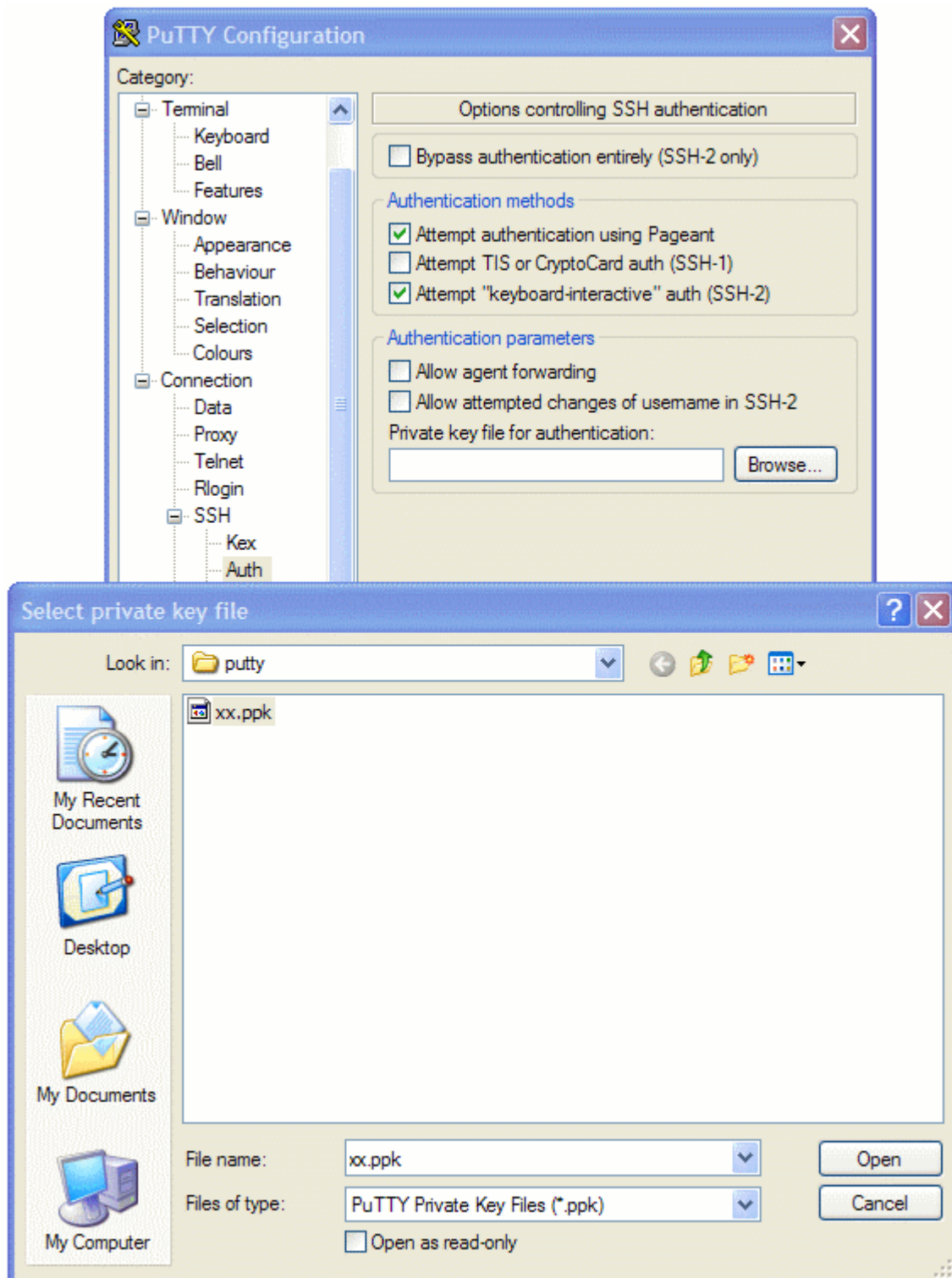


- Press the “Save Public Key” button to save your public key. Name it using the account name issued by the IB your country, with the extension “.pub”. Thus, for our example, the public key would be named “xx.pub”. Then, press the “Save Private Key” button to save your private key. Give it the same name, but without the “.pub” extension, e.g. enter “xx” in the naming box. You have created a 1024 bit SSH2 key using the RSA algorithm. Your public key is named “xx.pub”. Your private key is named “xx.ppk”, where “xx” represents the account name for your country.
- Email your public key to: pct.edi@wipo.int

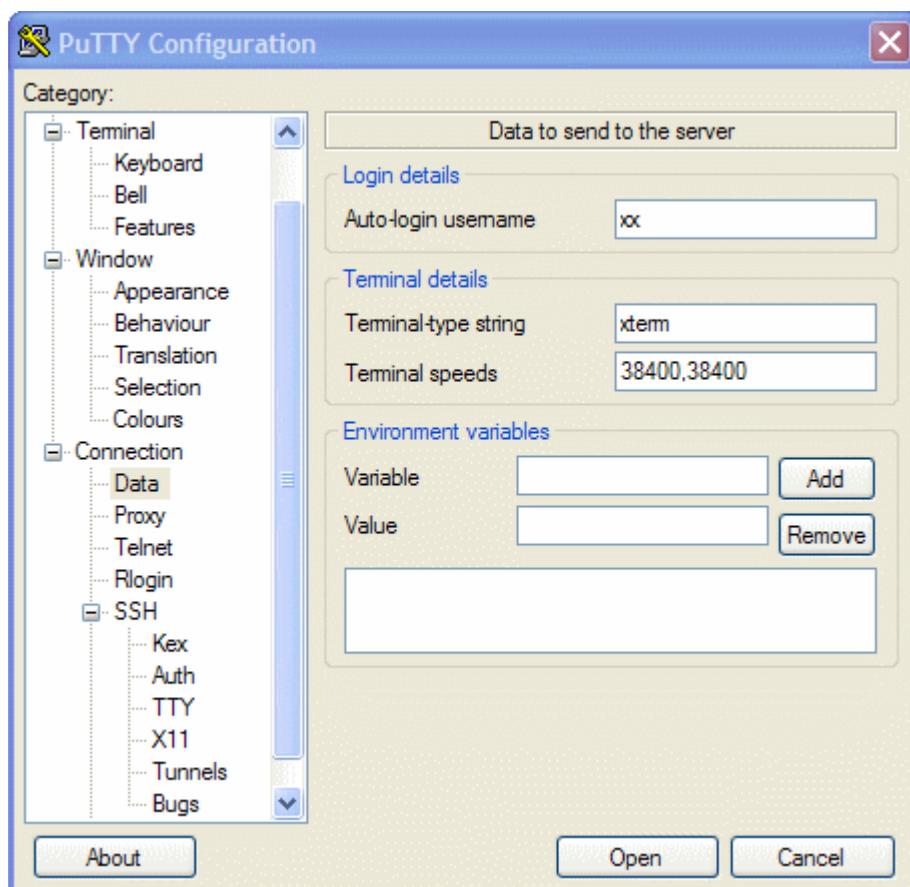
NEVER SEND THE PRIVATE KEY. We will not use any keys where the private key has been transmitted by email.

Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step

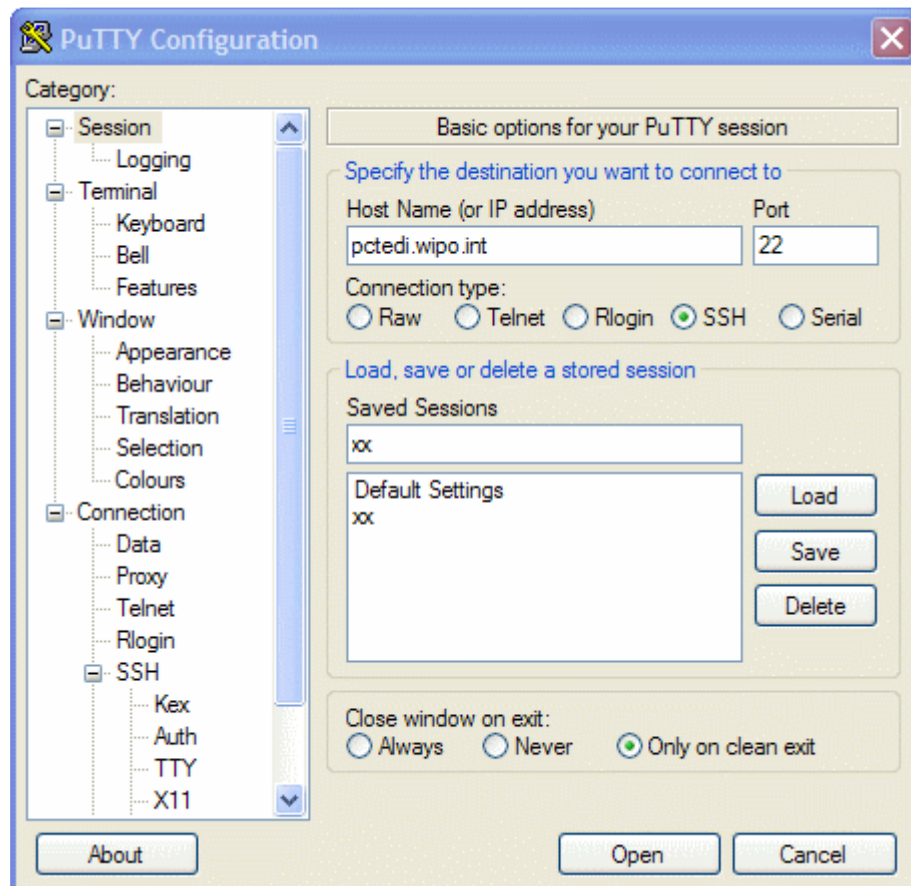
6. Start the PuTTY application. In the left hand panel (Category), find the SSH entry, and click on "Auth". Under "Private key file for authentication", click "Browse", and find the xx.ppk file. Select it.



7. Click on “Connection”, then “Data” in the left hand panel, and enter “xx” in the “Auto-login username” field.

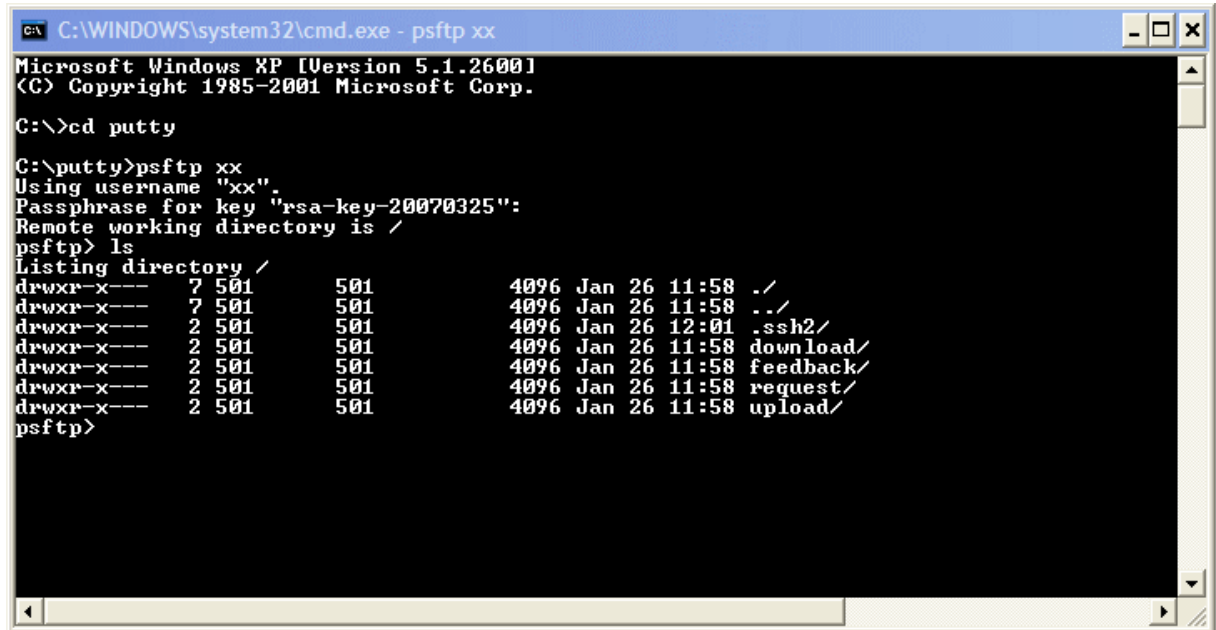


8. In the left hand panel (Category) click on “Session” at the top. You will be back at the entry screen with “pctedi.wipo.int” entered as the host name, and “xx” as the session name. Click Save. You have now saved your key information under the session name xx.



9. For security reasons, the PCT-EDI server does not allow remote SSH2 shell access. Therefore, the PuTTY SSH terminal client itself will not be used to access your account. Instead, the PuTTY psftp client will be used. It is a very simple command-line based client that uses PuTTY sessions such as the one you just made. For a more user friendly client, see the section that covers “WinSCP”.

10. To begin using psftp, open an MS-DOS window and, if necessary, switch to the PuTTY directory (using the MS-DOS "cd " command). Type "psftp xx" where "xx" is the name of the session you just created. You will be prompted for the passphrase you entered when you created the key, and then will be logged on using your public key. Note that the passphrase is not a password; it is never sent to the remote system. It simply secures your private key in the case of an unauthorized user gaining access to your local computer.



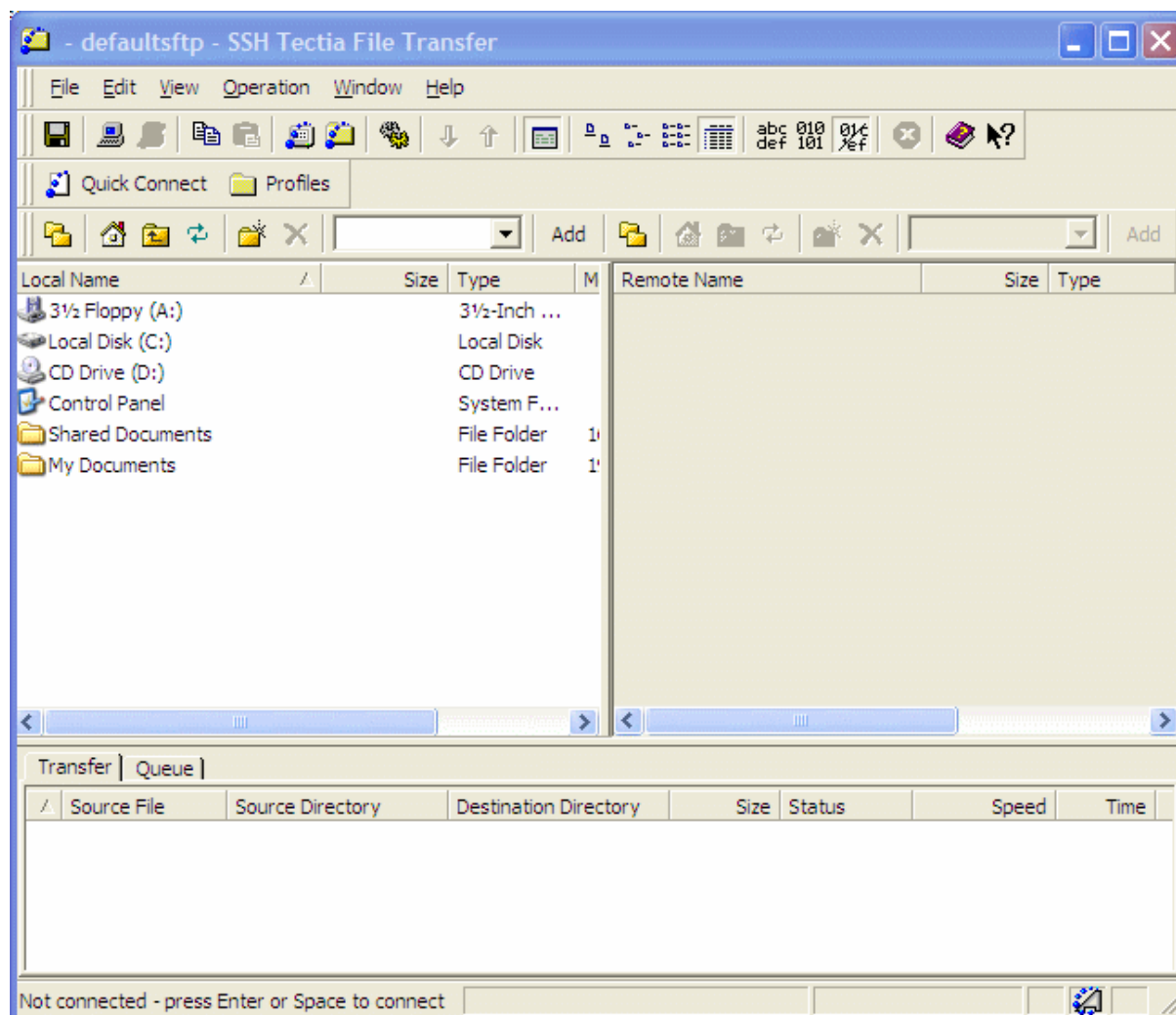
```
C:\WINDOWS\system32\cmd.exe - psftp xx
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>cd putty

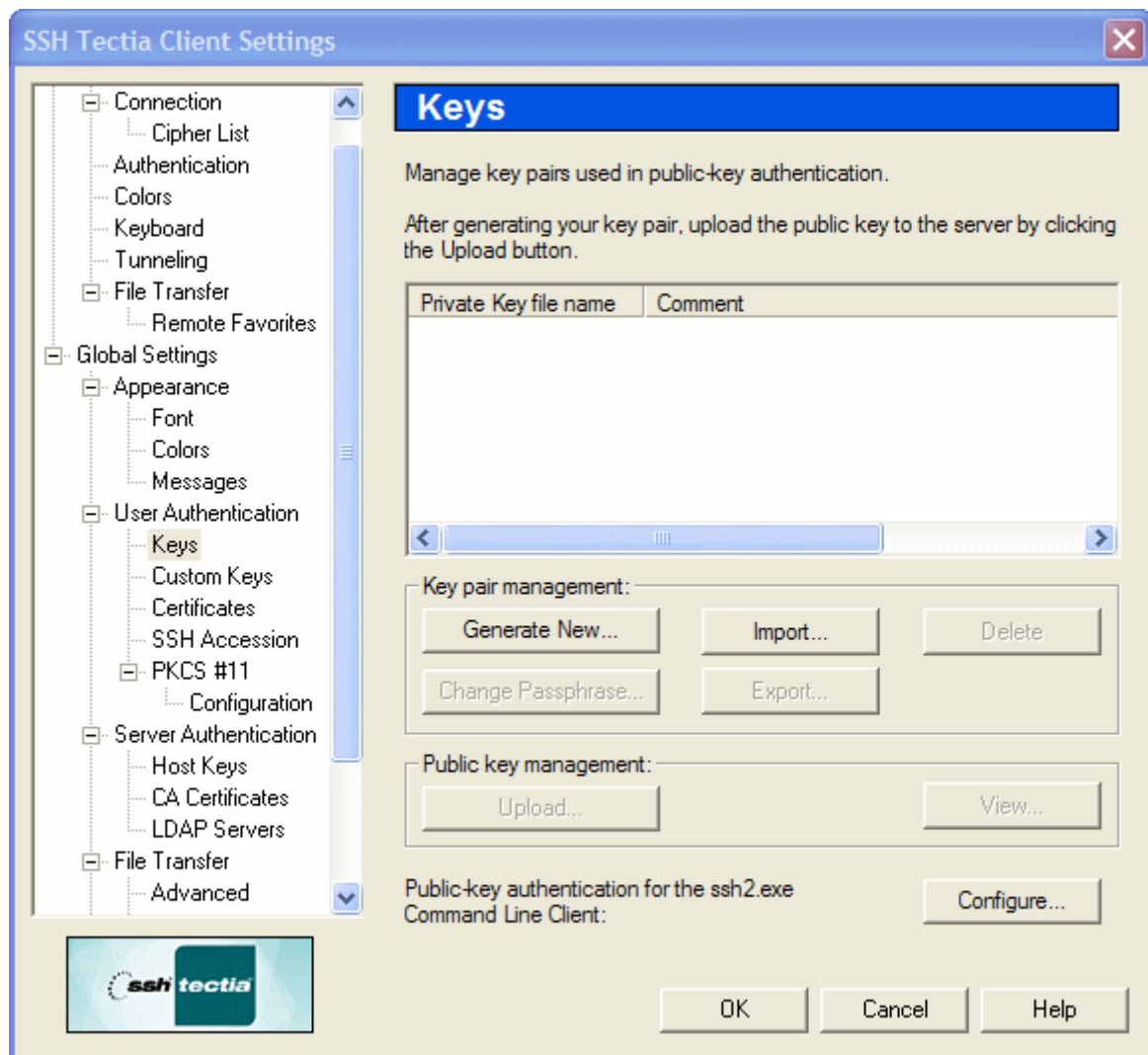
C:\putty>psftp xx
Using username "xx".
Passphrase for key "rsa-key-20070325":
Remote working directory is /
psftp> ls
Listing directory /
drwxr-x---  7 501      501      4096 Jan 26 11:58 ./
drwxr-x---  7 501      501      4096 Jan 26 11:58 ../
drwxr-x---  2 501      501      4096 Jan 26 12:01 .ssh2/
drwxr-x---  2 501      501      4096 Jan 26 11:58 download/
drwxr-x---  2 501      501      4096 Jan 26 11:58 feedback/
drwxr-x---  2 501      501      4096 Jan 26 11:58 request/
drwxr-x---  2 501      501      4096 Jan 26 11:58 upload/
psftp>
```

4.2.3 Setting Up and Using the Tectia SSH Client

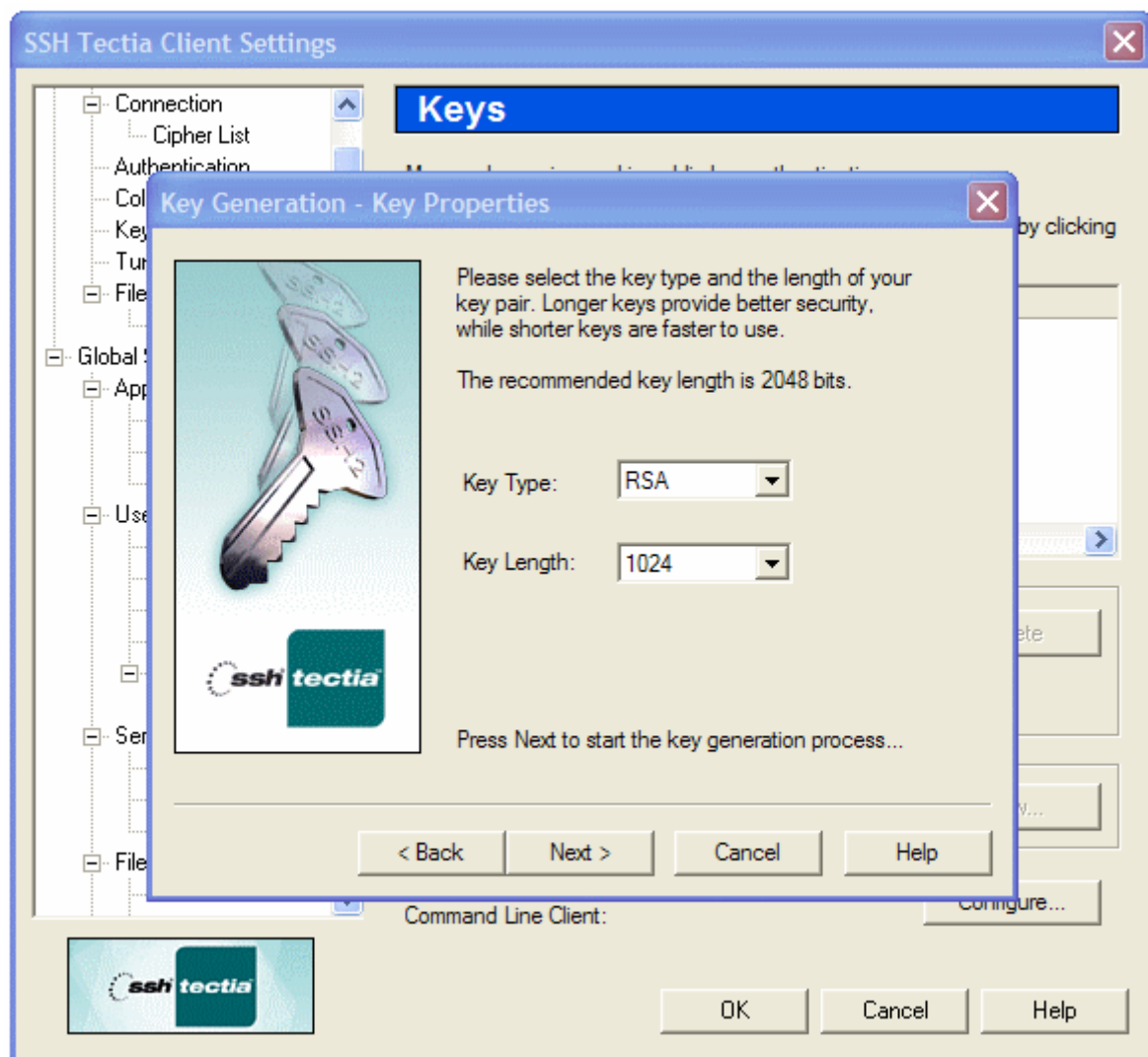
1. Download and install the Tectia SSH client for Windows. Follow the installer instructions and start the Tectia file transfer client. The client has the icon of a file folder, not the icon of a terminal screen. The following examples are based upon the freely available version of the commercial Tectia client, without PKI support. You will see the following screen:



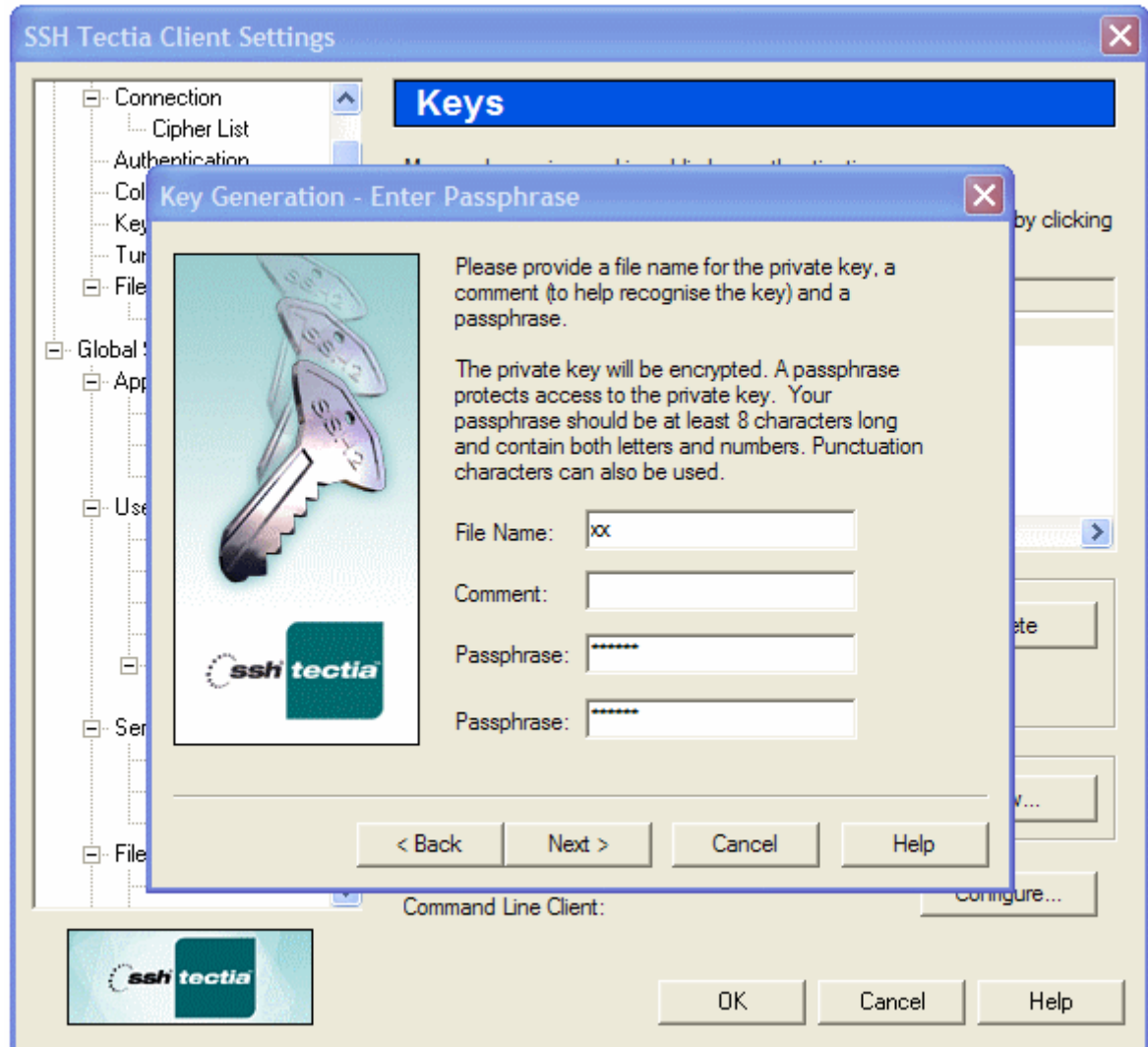
2. Select “Edit->Settings” from the menu. Then on the left-hand side of the Settings control panel, click on “Global Settings->User Authentication->Keys”. Under “Key pair management”, press “Generate New”.



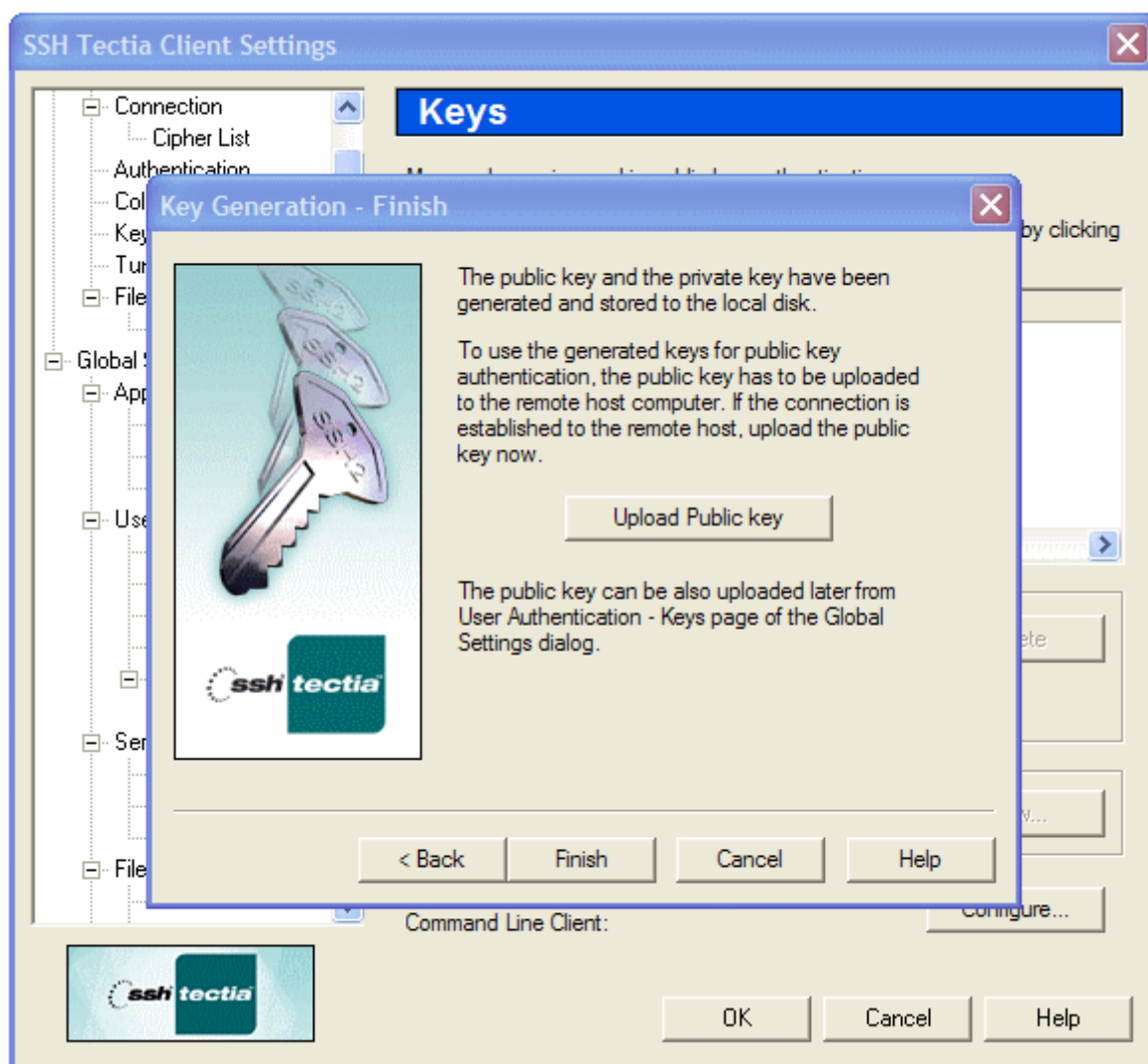
3. Follow the instructions given by the key generation wizard. You will be prompted to select a key type and key length. Select either RSA or DSA, with a minimum length of 1024 bits.



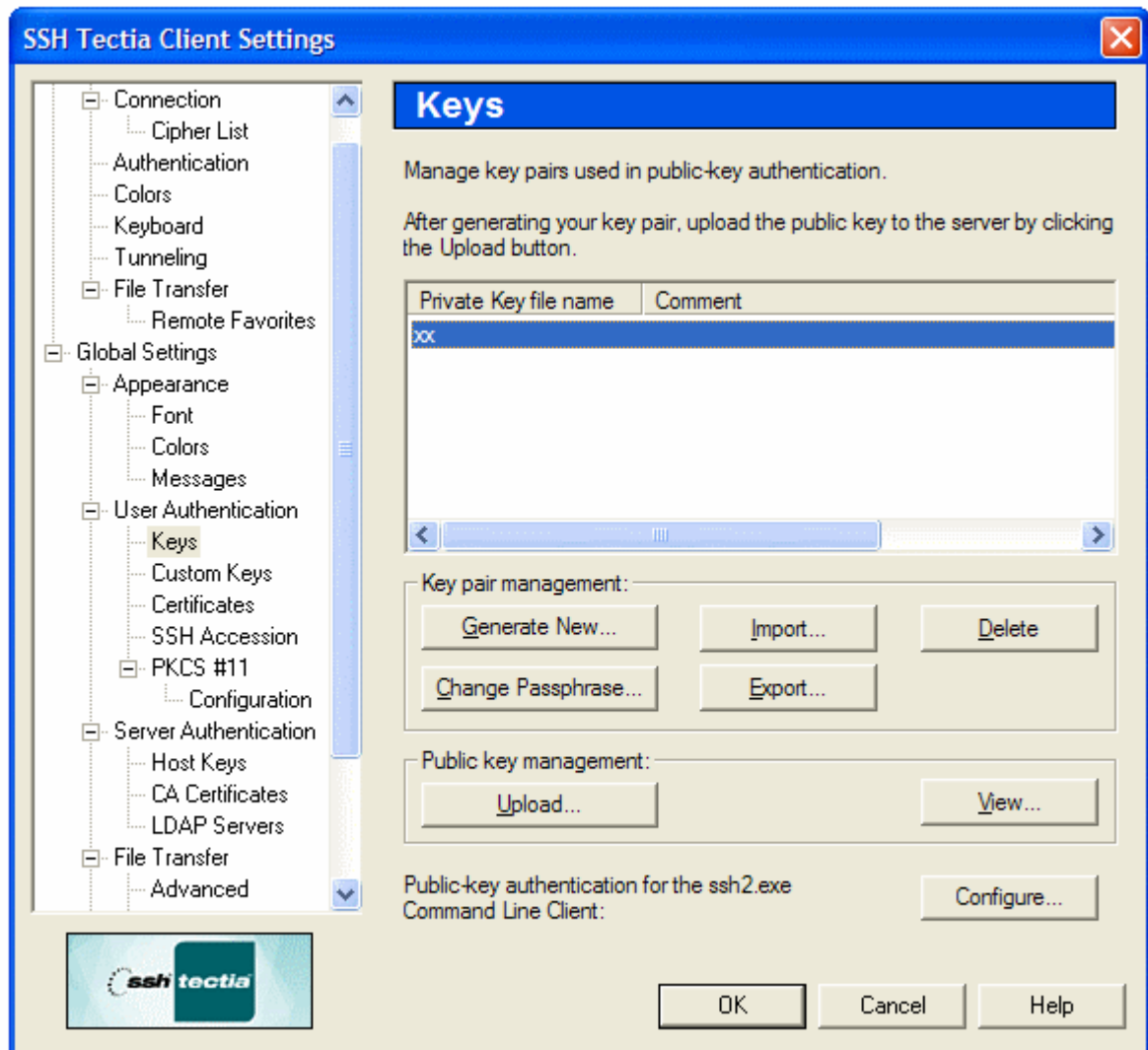
4. Click "Next" and wait for the key generation to finish. Click "Next" again. You will be prompted to provide a file name, comment and passphrase for the private key. Fill in the blanks as you desire, being sure to pick a good passphrase (minimum of 8 characters, consisting of letters, numbers and punctuation marks). Press "Next".



5. Click on "Finish".

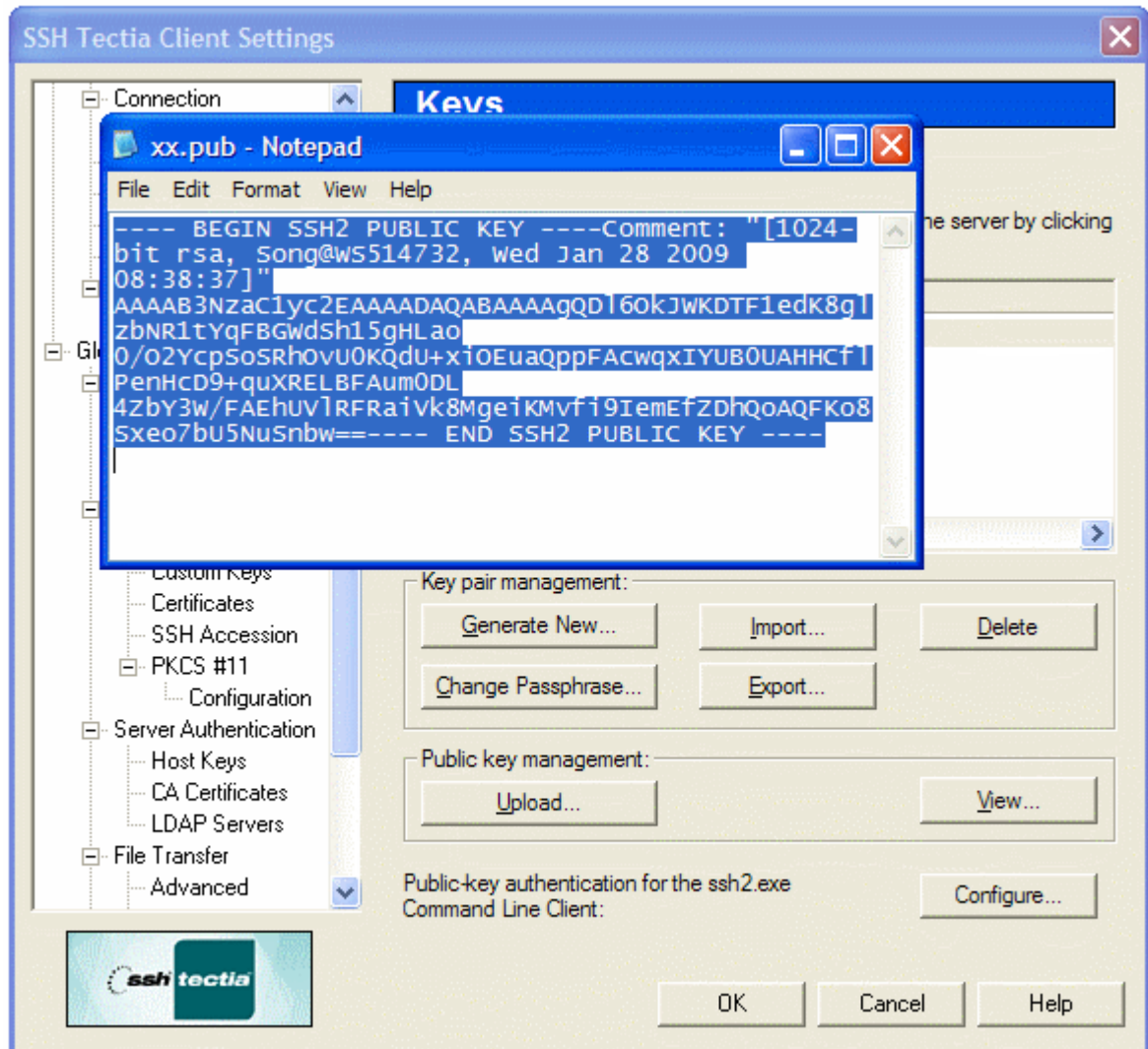


6. Double-click on your key in the “Private Key file name” list. This will open the corresponding public key (e.g. xx.pub) in Notepad.

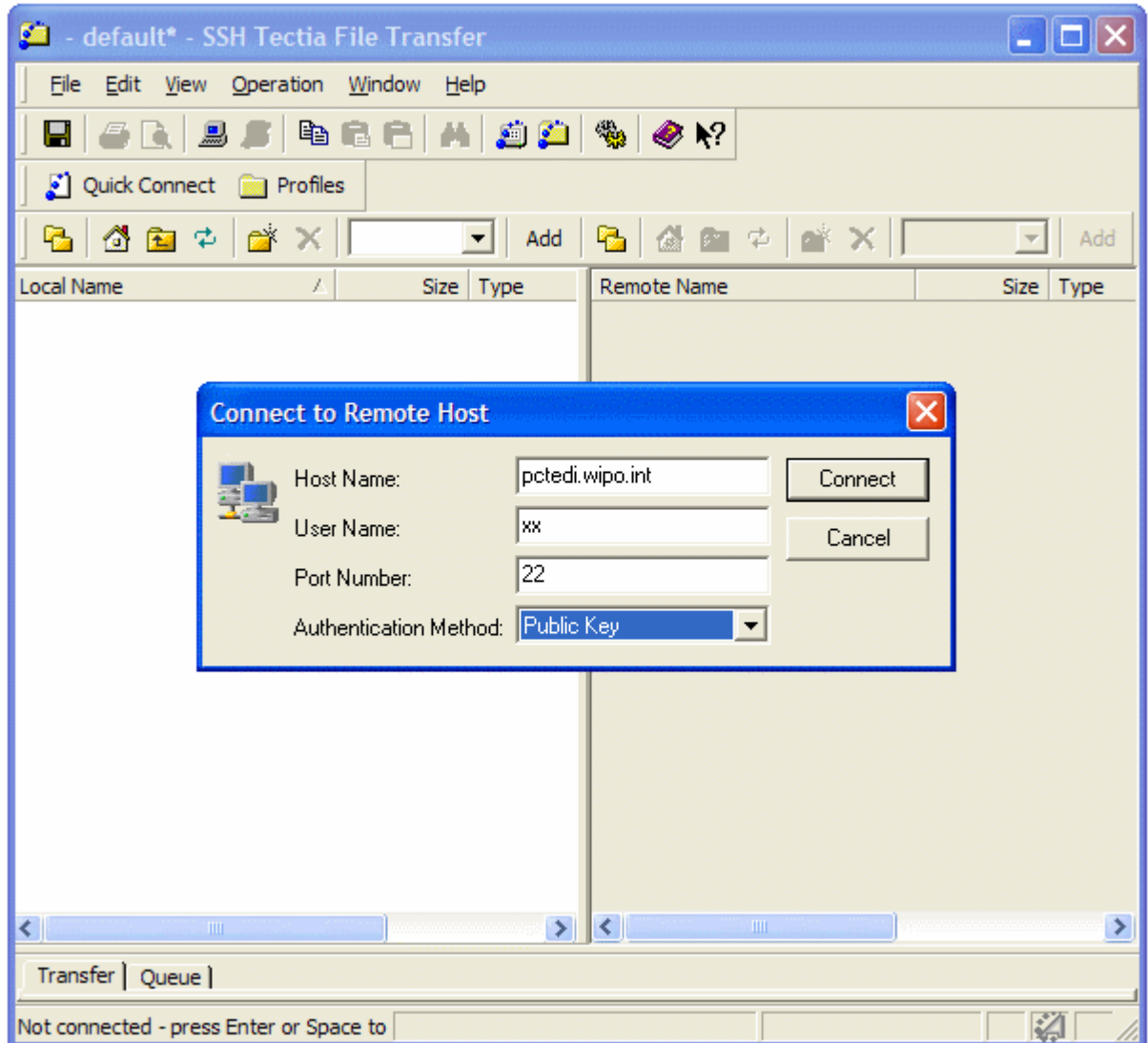


7. In Notepad, do "Select All", then copy and paste the contents of xx.pub into an email and send it to pct.edi@wipo.int

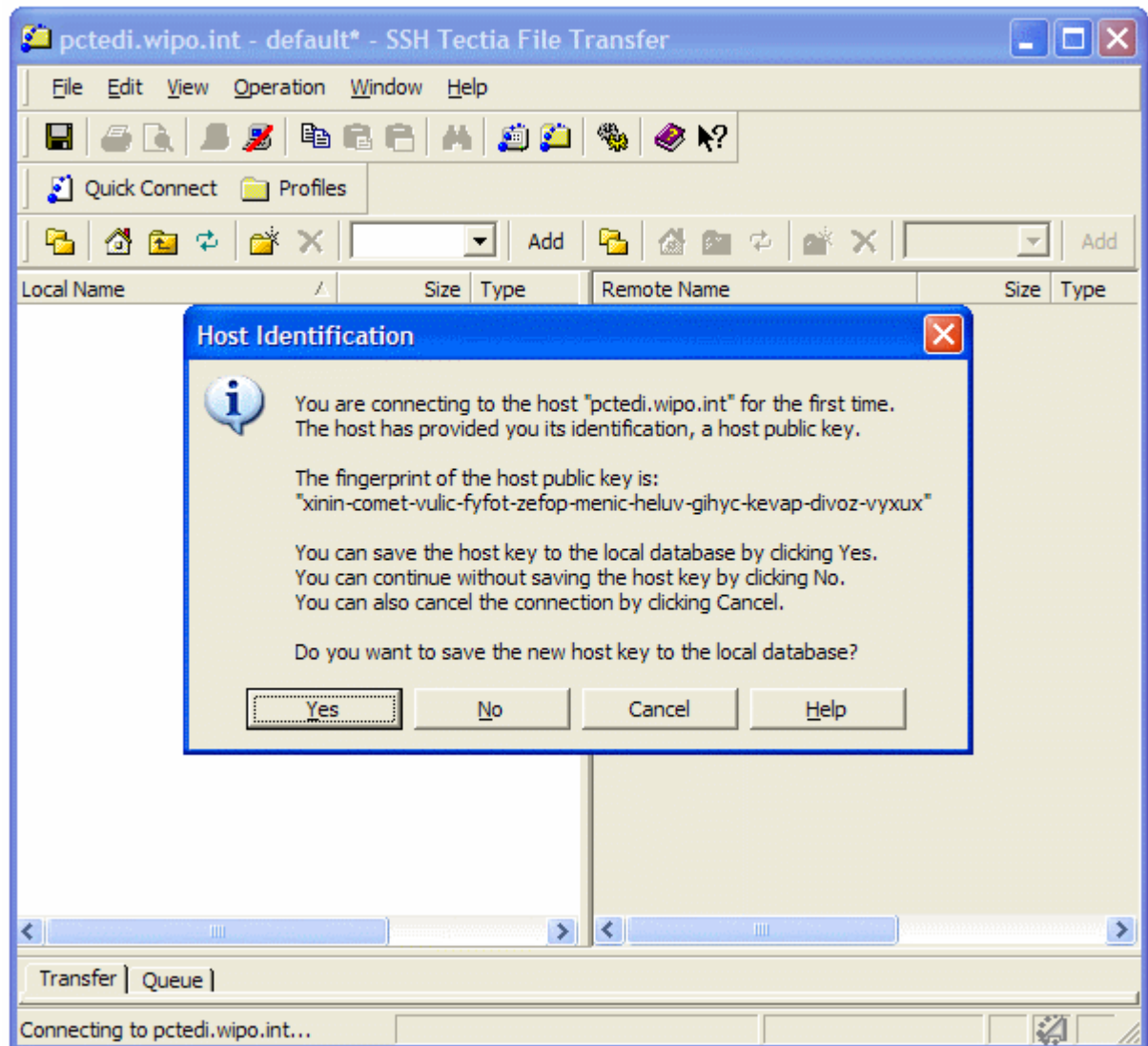
Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step



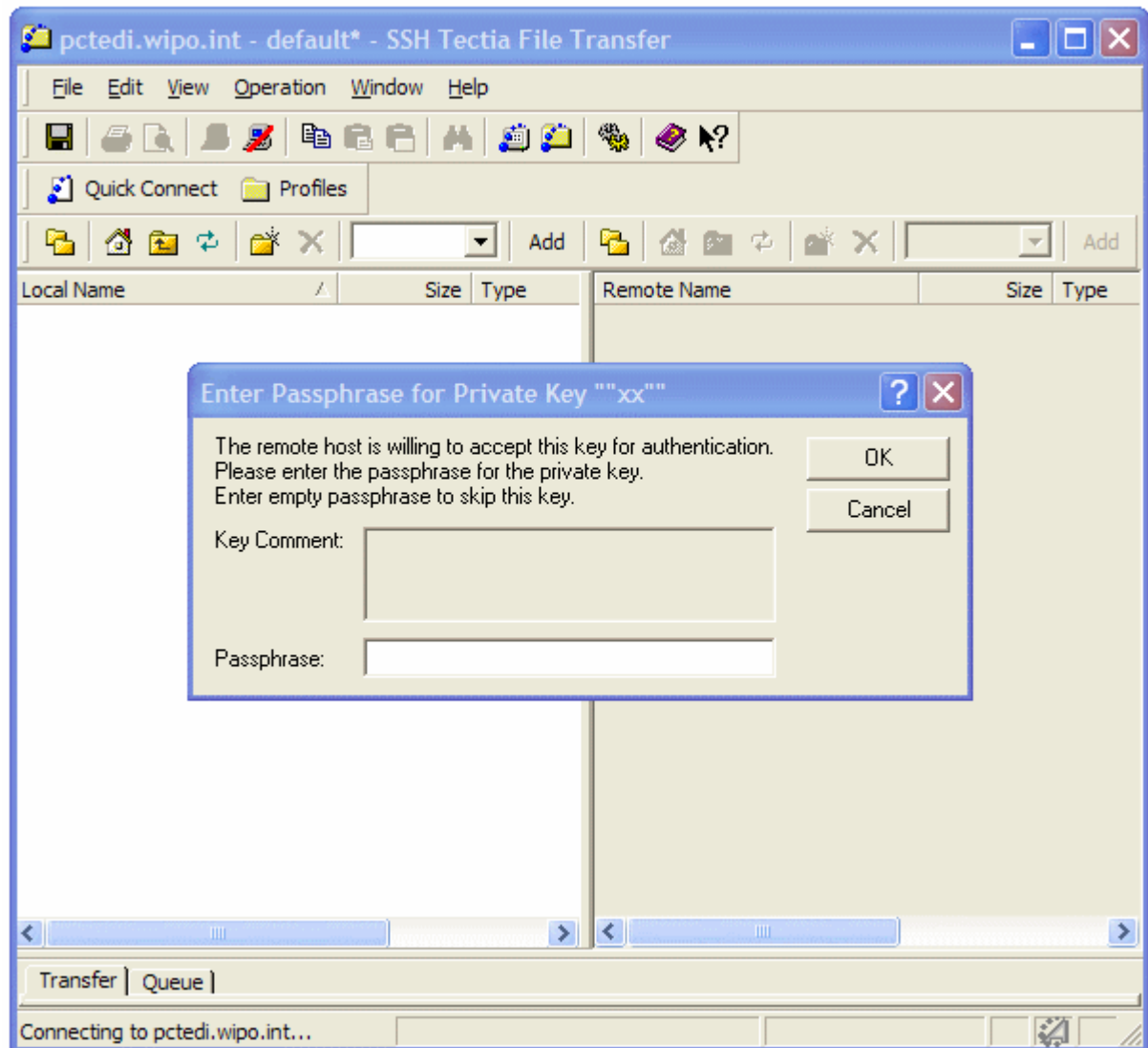
8. Start the Tectia file transfer client, then press the “Quick Connect” button and fill in the dialog box fields as below with “pctedi.wipo.int” for the “Host Name”, your account name for the “User Name”, and select “Public Key” from the drop-down box as the “Authentication Method”. Then, press “Connect”.



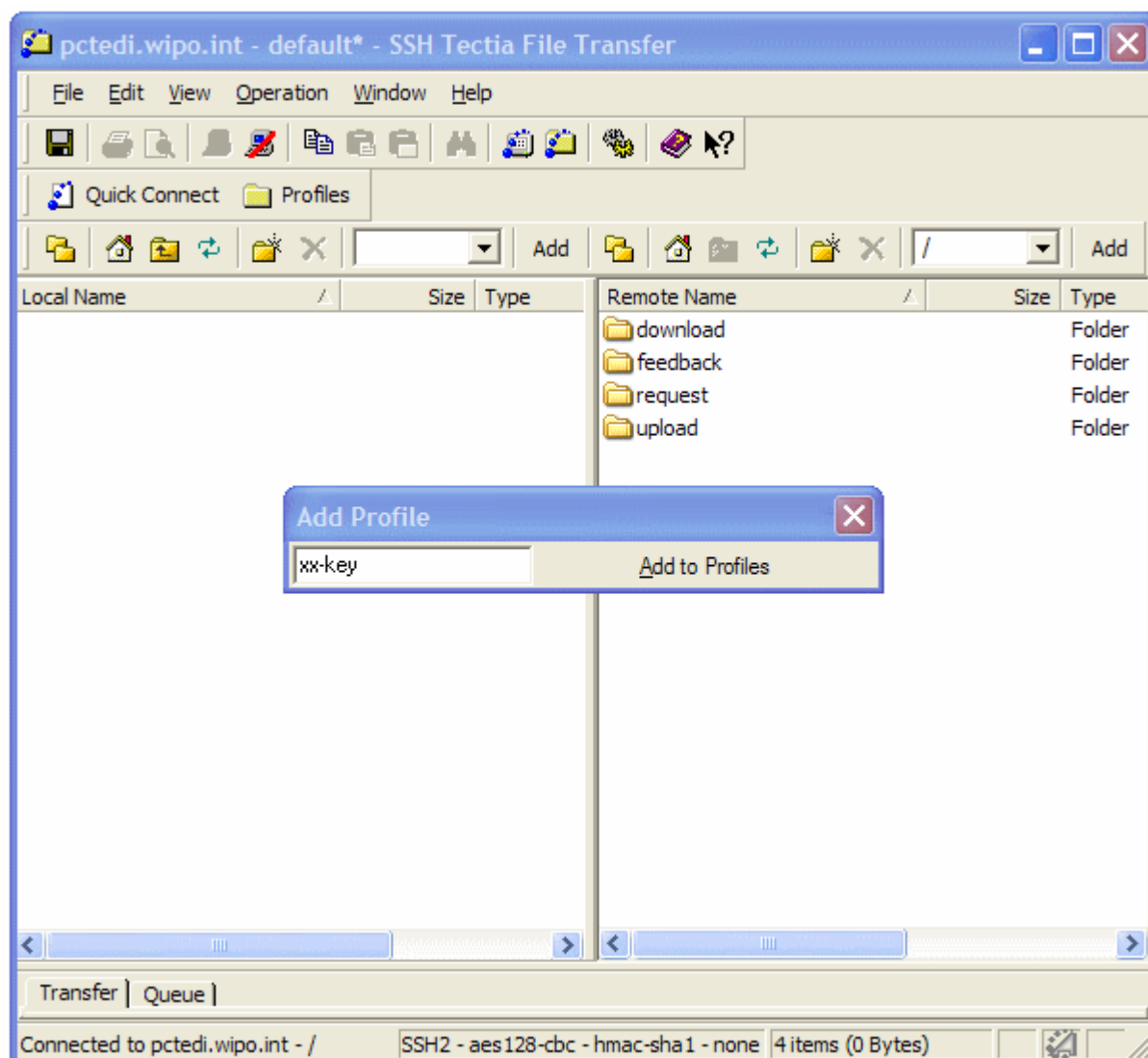
9. You will then be prompted to accept the host key of the PCT-EDI server. Click “Yes”.



10. You will be prompted for the key passphrase. Type in the passphrase and click "OK". You will be connected to the PCT-EDI server.



11. Press “Profiles” and “Add Profile”. You will be prompted to name the current profile. Name it “xx-key” and click on “Add to Profiles” to save your key-based connection profile. You may now connect to the PCT-EDI server at any time by selecting the profile “xx-key”.



4.3 UNIX ENVIRONMENT

The following sections are written for the experienced Unix user or system administrator. It discusses the freely available **OpenSSH** and **lftp** packages for the Unix environment.

OpenSSH is a free, non-commercial version of the SSH protocol suite of network connectivity tools.

The OpenSSH suite includes a variety of utilities of interest to intellectual property Offices requiring secure Internet-based communications. Of particular interest to users of the PCT-EDI system is the *sftp* client, which implements (as of OpenSSH 2.5.0) complete SFTP support.

OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0, permitting communication with most UNIX, Windows and other commercial ssh implementations. The SSH 2.0 protocol avoids using the patented RSA algorithm and uses the freely useable DH and DSA algorithms instead.

lftp is a file transfer program that allows sophisticated sftp, ftp, http and other connections to other hosts. **lftp** can handle seven file access methods - ftp, ftps, http, https, hftp, fish, sftp and file (https and ftps are only available when lftp is compiled with the openssl library). You can specify the method to use in the "open URL" command, e.g. "open sftp://pctedi.wipo.int". Sftp is implemented in ssh2 as a call to the external program *sftp* and thus the OpenSSH "ssh" application to be installed and functioning.

Every operation in **lftp** is reliable, that is any non-fatal error is ignored and the operation is repeated. This means that if downloading breaks for any reason, it will be restarted from the point automatically.

lftp has a shell-like command syntax allowing you to launch several commands in parallel in the background (&). It is also possible to group commands within () and execute them in background. All background jobs are executed in the same single process. You can put a foreground job in the background with ^Z (ctl-z) and bring it back with the `wait' command (or `fg' which is an alias to `wait'). To list running jobs, use the command `jobs'. Some commands allow redirecting their output (cat, ls, ...) to file or via a pipe to external command. Commands can be executed conditionally based on the termination status of previous command (&&, ||).

If you exit **lftp** when some jobs are not yet finished yet, **lftp** will automatically place itself in nohup mode in the background.

lftp has a built in mirror function which can download or update a whole directory tree. There is also a reverse mirror function (`mirror -R`) which uploads or updates a directory tree on the server.

The command “`at`” launches a job at specified time in the current context, the command “`queue`” to queue commands for sequential execution with the connected current server, as well as other useful features such as a precious commands history. The best source of usage documentation for **lftp** is the *lftp(1)* man page, installed when the client is installed. It can also be viewed at <http://lftp.yar.ru/lftp-man.html>

On startup, **lftp** executes `/etc/lftp.conf` and then `~/.lftprc` and `~/.lftp/rc`. Individual users can place aliases and ``set'` commands there. These commands are discussed in the *lftp(1)* man page.

lftp has a number of user-controlled variables. You can use ``set -a'` to see all of the variables and their values or ``set -d'` to see list of default values.

4.3.1 Installing and Using the OpenSSH Utilities⁷

4.3.1.1 Installation

Installation of the OpenSSH utilities can only be performed by your Unix systems administrator, and is dependent upon the target Unix system. Installation instructions, source code, and precompiled binaries for a variety of Unix systems can be found at <http://www.openssh.com/portable.html>

The following usage instructions are based upon the existence of a properly installed OpenSSH subsystem.

4.3.1.2 Configuration and Use

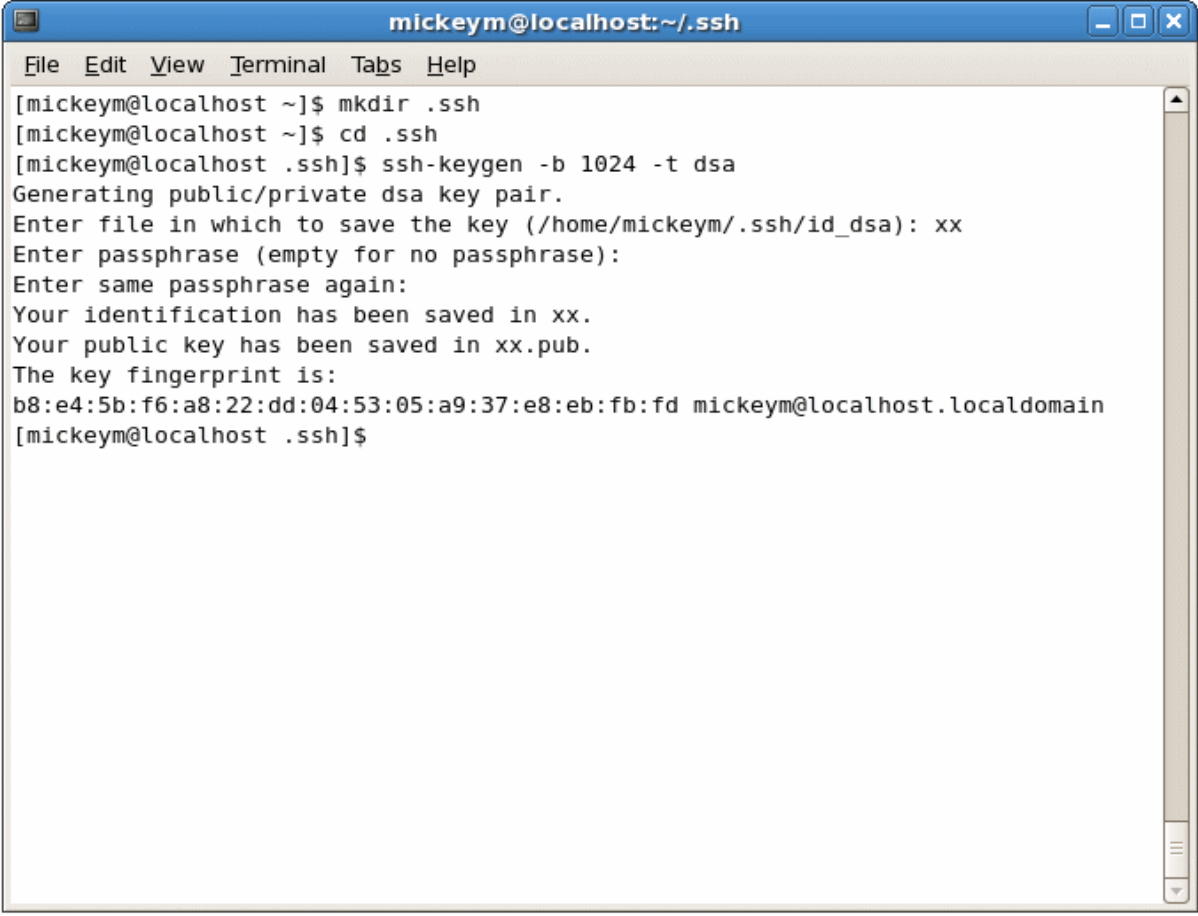
This section describes the configuration and use of the OpenSSH *sftp* client. *Sftp* uses a set of commands similar to those found in normal Internet FTP. It also describes the use of *ssh-keygen* to create your public and private keys.

1. Log in to your local Unix machine and create a directory named “.ssh”. Change to that directory. You must be in the “.ssh” directory when executing the next commands.

⁷ Derived from the OpenSSH web site documentation

2. Enter the command “ssh-keygen -b 1024 -t dsa” to generate a OpenSSH format 1024 bit DSA key. You will be prompted for a file in which to save the key. Enter the two-letter WIPO ST.3 country code for your Office.

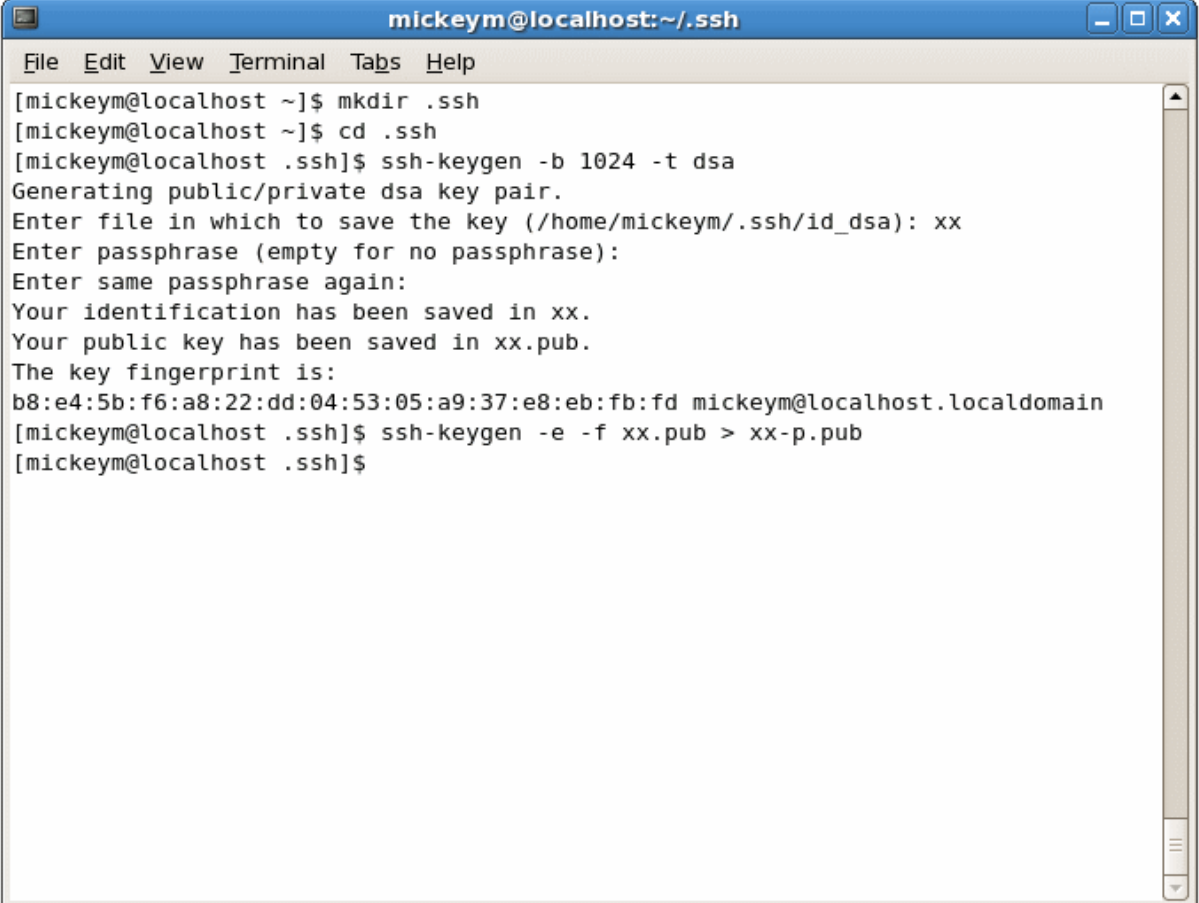
When key generation is complete, you will be prompted to enter a passphrase. If this account will be used for non-interactive applications (such as automated scripts and downloading”, press “Enter”. Otherwise, if the account will be used by humans, enter a good (at least 8 characters, with letters, numbers and punctuation marks) passphrase in the given blocks. You will be prompted for this passphrase whenever you use this key. The passphrase is never sent to the remote machine. The following examples assume the existence of a passphrase.



```
mickeym@localhost: ~/.ssh
File Edit View Terminal Tabs Help
[mickeym@localhost ~]$ mkdir .ssh
[mickeym@localhost ~]$ cd .ssh
[mickeym@localhost .ssh]$ ssh-keygen -b 1024 -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/mickeym/.ssh/id_dsa): xx
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in xx.
Your public key has been saved in xx.pub.
The key fingerprint is:
b8:e4:5b:f6:a8:22:dd:04:53:05:a9:37:e8:eb:fb:fd mickeym@localhost.localdomain
[mickeym@localhost .ssh]$
```

This key pair (xx and xx.pub) are suitable for all OpenSSH applications.

3. The IB uses the commercial Tectia SSH server, which uses public keys in the newer SECSH Public Key format. It is now necessary to convert your xx.pub public key into this format. Type “ssh-keygen -e -f xx.pub > xx-p.pub”, where “xx” is your account name. This will copy your xx.pub public key file to xx-p.pub, converting it to the SECSH format.



```
mickeym@localhost: ~/.ssh
File Edit View Terminal Tabs Help
[mickeym@localhost ~]$ mkdir .ssh
[mickeym@localhost ~]$ cd .ssh
[mickeym@localhost .ssh]$ ssh-keygen -b 1024 -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/mickeym/.ssh/id_dsa): xx
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in xx.
Your public key has been saved in xx.pub.
The key fingerprint is:
b8:e4:5b:f6:a8:22:dd:04:53:05:a9:37:e8:eb:fb:fd mickeym@localhost.localdomain
[mickeym@localhost .ssh]$ ssh-keygen -e -f xx.pub > xx-p.pub
[mickeym@localhost .ssh]$
```

4. Email your converted public key (xx-p.pub) to: pct.edi@wipo.int

NEVER SEND THE PRIVATE KEY. We will not use any keys where the private key has been transmitted by email.

Wait until you receive an email notifying that your key has been activated for login before proceeding to the next step.

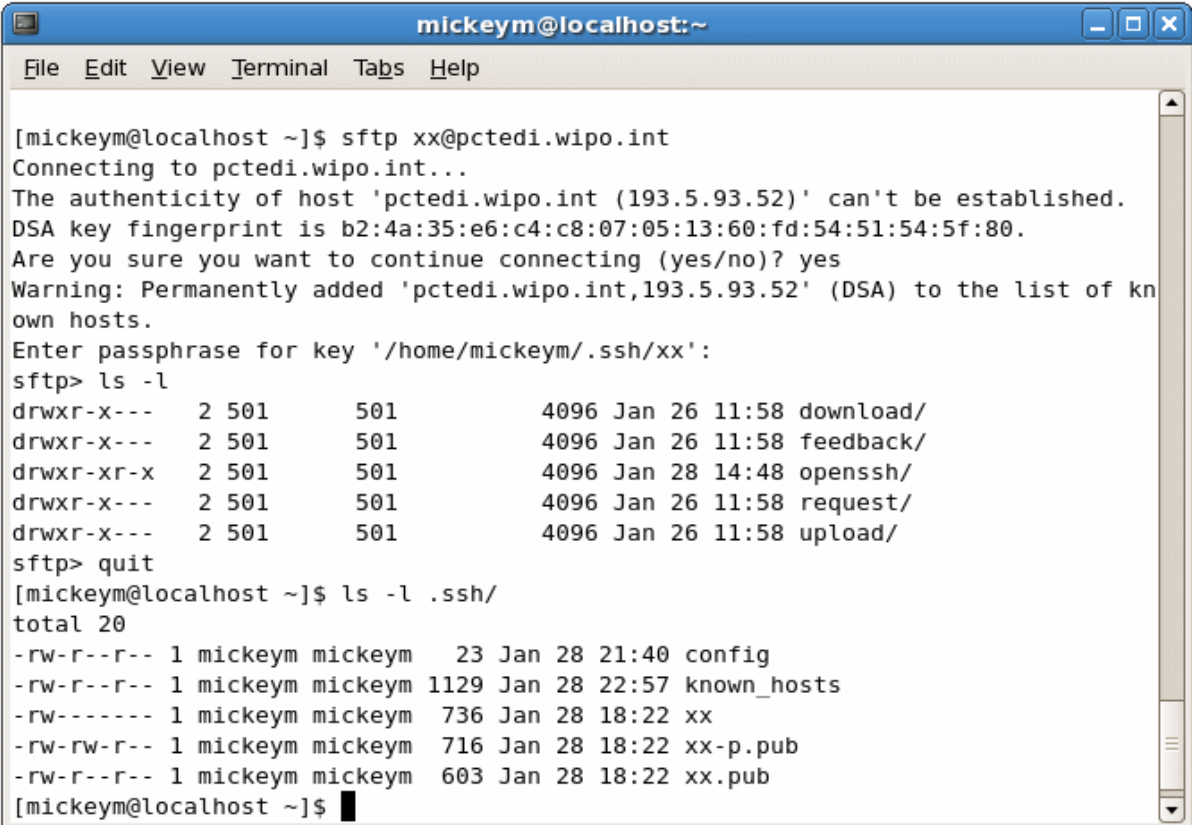
5. Ensure that you are in the ".ssh" directory by using the "pwd" command. You will now create your OpenSSH configuration file. Use your favorite Unix text editor to create the file "config" containing the line (without quotes) "IdentityFile ~/.ssh/xx". This will cause your private key "xx" to be used to identify yourself to the PCT EDI server.

Enter the command "*sftp xx@pctedi.wipo.int*" to open a connection to the server using "xx" as the account name.

You will be prompted to accept the authenticity of the pctedi.wipo.int PCT-EDI server. Type "yes".

If you assigned a passphrase to your private key, you will be prompted for it. Then, you will be logged into your account on the PCT-EDI server.

The screenshot below shows the existence of the "config" file, the results of connecting to the server, and a listing of the "xx" account directory space on the PCT-EDI server.



```
mickeym@localhost:~  
File Edit View Terminal Tabs Help  
[mickeym@localhost ~]$ sftp xx@pctedi.wipo.int  
Connecting to pctedi.wipo.int...  
The authenticity of host 'pctedi.wipo.int (193.5.93.52)' can't be established.  
DSA key fingerprint is b2:4a:35:e6:c4:c8:07:05:13:60:fd:54:51:54:5f:80.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'pctedi.wipo.int,193.5.93.52' (DSA) to the list of known hosts.  
Enter passphrase for key '/home/mickeym/.ssh/xx':  
sftp> ls -l  
drwxr-x--- 2 501 501 4096 Jan 26 11:58 download/  
drwxr-x--- 2 501 501 4096 Jan 26 11:58 feedback/  
drwxr-xr-x 2 501 501 4096 Jan 28 14:48 openssh/  
drwxr-x--- 2 501 501 4096 Jan 26 11:58 request/  
drwxr-x--- 2 501 501 4096 Jan 26 11:58 upload/  
sftp> quit  
[mickeym@localhost ~]$ ls -l .ssh/  
total 20  
-rw-r--r-- 1 mickeym mickeym 23 Jan 28 21:40 config  
-rw-r--r-- 1 mickeym mickeym 1129 Jan 28 22:57 known_hosts  
-rw----- 1 mickeym mickeym 736 Jan 28 18:22 xx  
-rw-rw-r-- 1 mickeym mickeym 716 Jan 28 18:22 xx-p.pub  
-rw-r--r-- 1 mickeym mickeym 603 Jan 28 18:22 xx.pub  
[mickeym@localhost ~]$
```

4.3.2 Installing and Using lftp

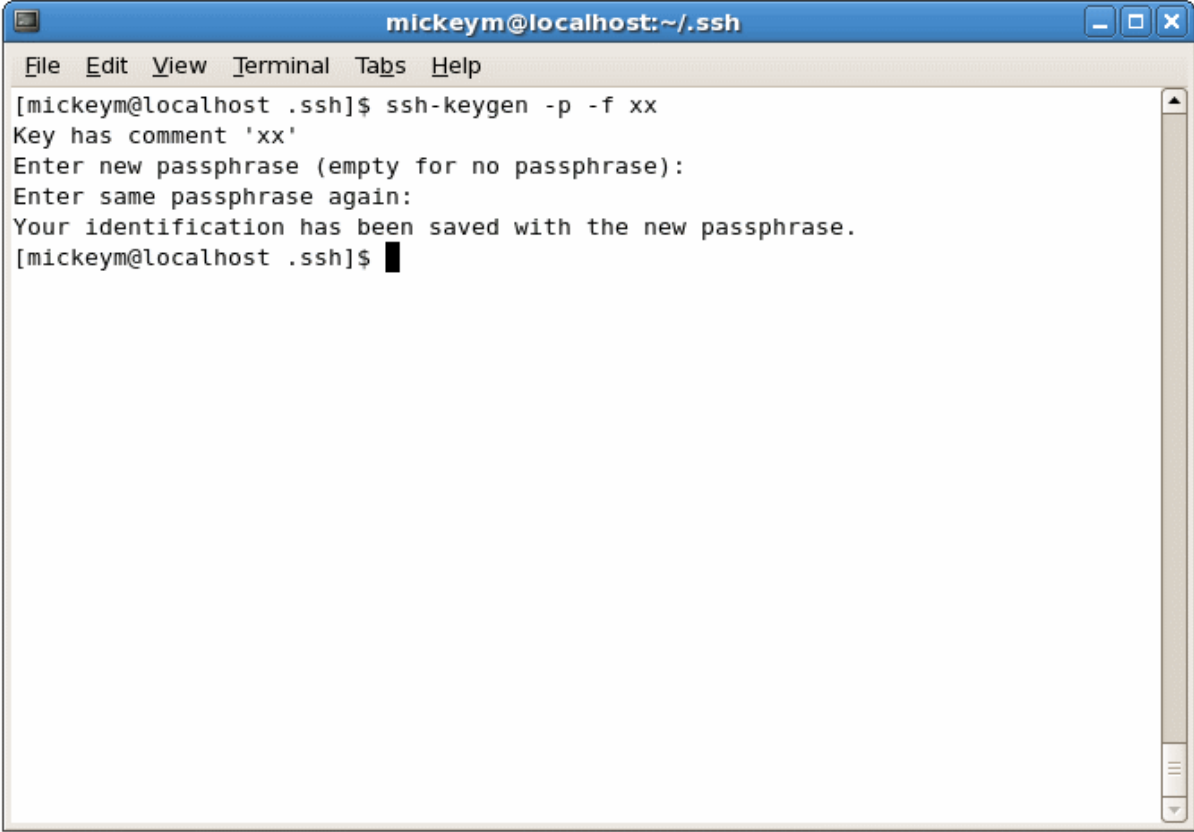
4.3.2.1 Installation

Before installing lftp, the OpenSSH suite of utilities must be installed as above. It is also necessary to have configured your local account to support public key authentication with the PCT-EDI server. Once you can successfully connect to the PCT-EDI server using “*sftp*” and public key authentication as above, you may install lftp. As with OpenSSH, lftp should be installed by an experienced Unix systems administrator. Source code and precompiled binaries, along with installation instructions, may be found at <http://lftp.yar.ru/>

The local configuration and use instructions below assume the presence of a working, correct lftp installation.

4.3.2.2 Configuration and Use

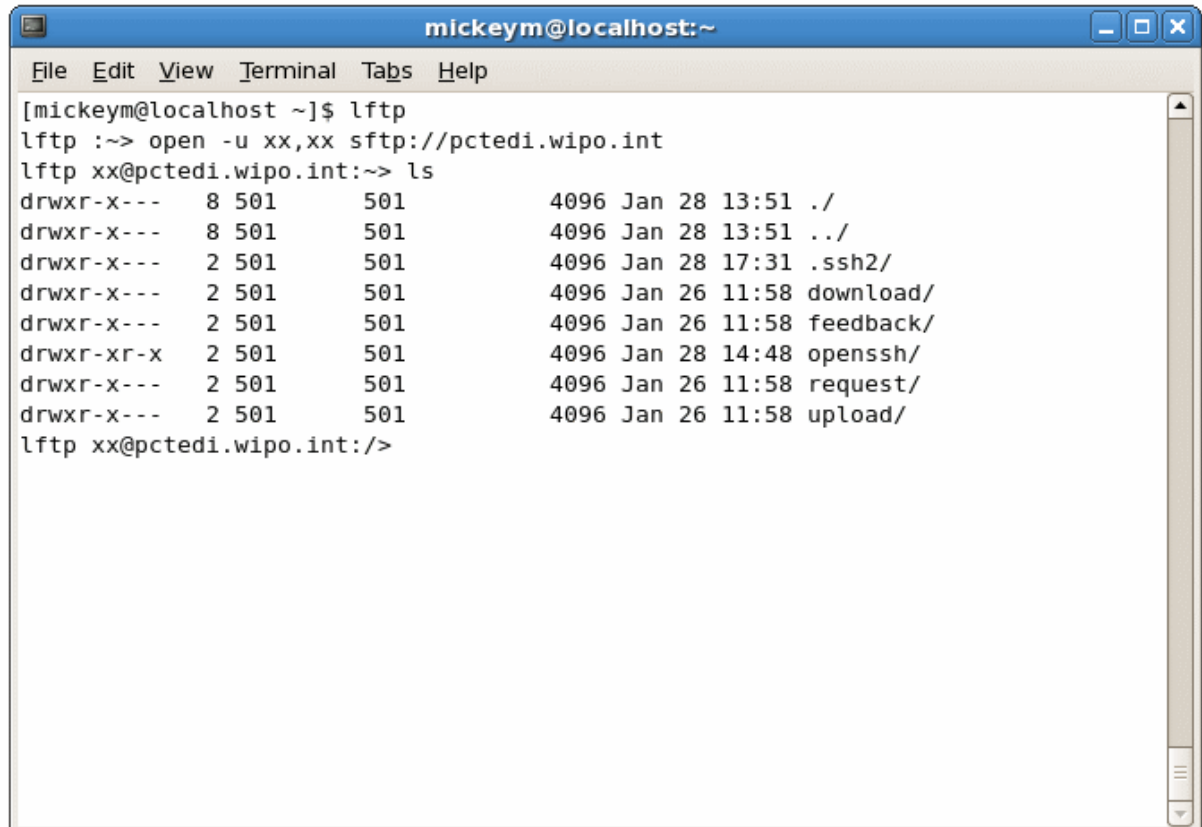
1. Lftp requires the use of an unprotected private key file in the user “.ssh” directory. If you current OpenSSH private key is protected, you may unprotect it with the “ssh-keygen” application. First, cd to the “.ssh” directory. Then, type “ssh-keygen -p” and follow the instructions as shown below. Enter your account name as the file name. You will be prompted for your existing passphrase. You may then change the passphrase to empty.



```
mickeym@localhost: ~/.ssh
File Edit View Terminal Tabs Help
[mickeym@localhost .ssh]$ ssh-keygen -p -f xx
Key has comment 'xx'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
[mickeym@localhost .ssh]$
```

2. You may now connect to the PCT-EDI system. Type *lftp*. You will see the lftp prompt. Then, type *open -u xx,xx sftp://pctedi.wipo.int*. This specifies an SFTP connection to *pctedi.wipo.int*, with the username *xx*. Note that the second *xx* after the *-u* flag is the account password. No password is needed for this account, as we are using public key authentication. The second *xx* is simply a filler to prevent lftp from asking for a password.

After entering the “command” you will once again see the lftp prompt. Simply type */s* to make the connection and show a directory listing.



```
mickeym@localhost:~  
File Edit View Terminal Tabs Help  
[mickeym@localhost ~]$ lftp  
lftp :~> open -u xx,xx sftp://pctedi.wipo.int  
lftp xx@pctedi.wipo.int:~> ls  
drwxr-x---  8 501      501      4096 Jan 28 13:51 ./  
drwxr-x---  8 501      501      4096 Jan 28 13:51 ../  
drwxr-x---  2 501      501      4096 Jan 28 17:31 .ssh2/  
drwxr-x---  2 501      501      4096 Jan 26 11:58 download/  
drwxr-x---  2 501      501      4096 Jan 26 11:58 feedback/  
drwxr-xr-x  2 501      501      4096 Jan 28 14:48 openssh/  
drwxr-x---  2 501      501      4096 Jan 26 11:58 request/  
drwxr-x---  2 501      501      4096 Jan 26 11:58 upload/  
lftp xx@pctedi.wipo.int:/>
```

3. You may now use lftp however you like. The lftp man page contains extensive information on the various lftp features.

5 ADVANCED TOPICS

5.1 INTRODUCTION

This section covers the advanced topics of managing multiple users within a single Office account, and the development of customized applications for the PCT-EDI service.

5.2 ADDING AND REVOKING USERS

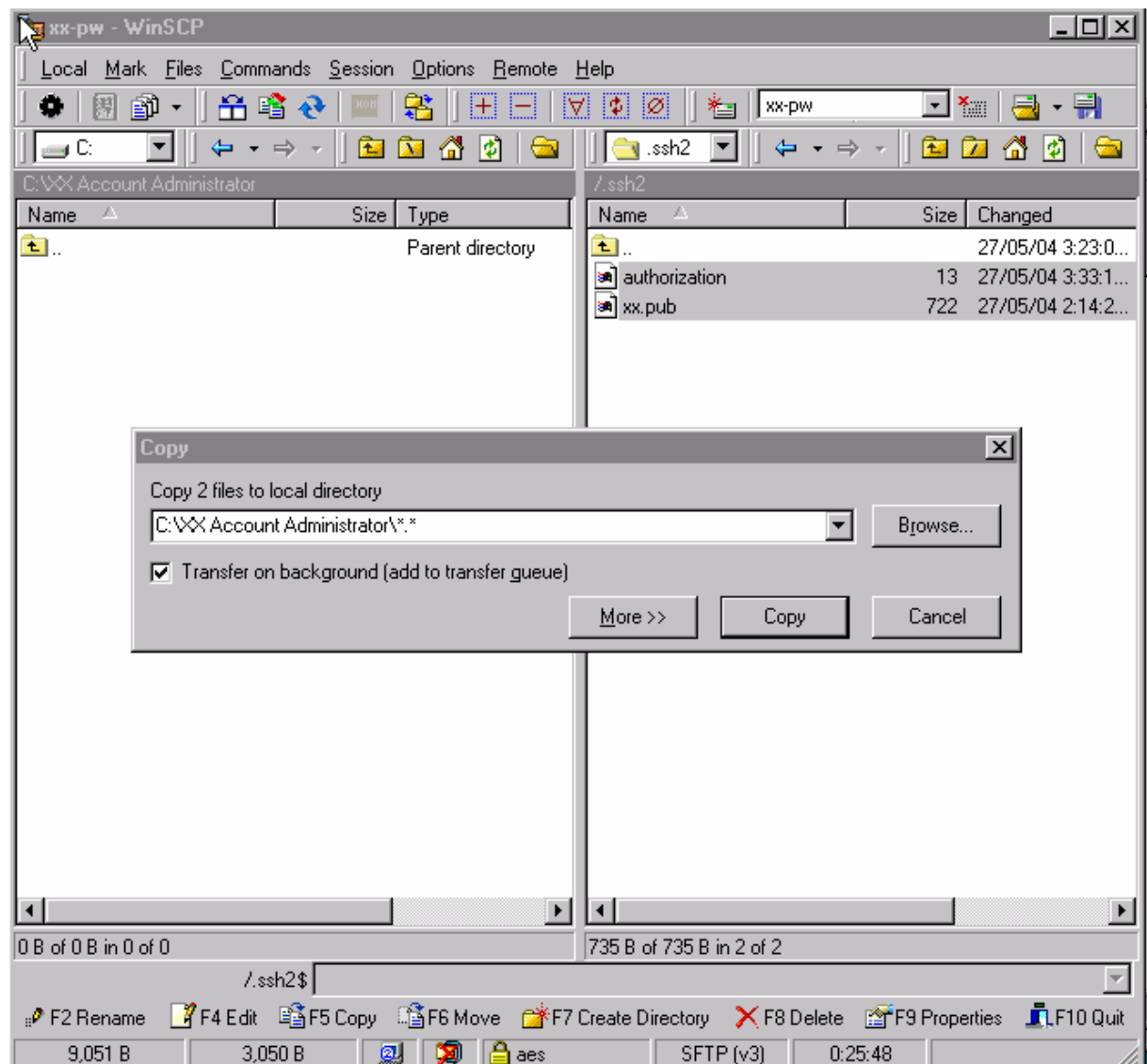
While each Office is assigned only one account, it is quite simple to add additional users to the Office account, and revoke them as necessary. Using public key authentication, no passwords need be circulated when new users are added, or changed when existing users are deleted.

It is suggested that one individual in each Office be responsible for the addition and/or deletion of users. This individual will be referred to as the “account administrator” in this example. The account administrator will be shown using the WinSCP client.

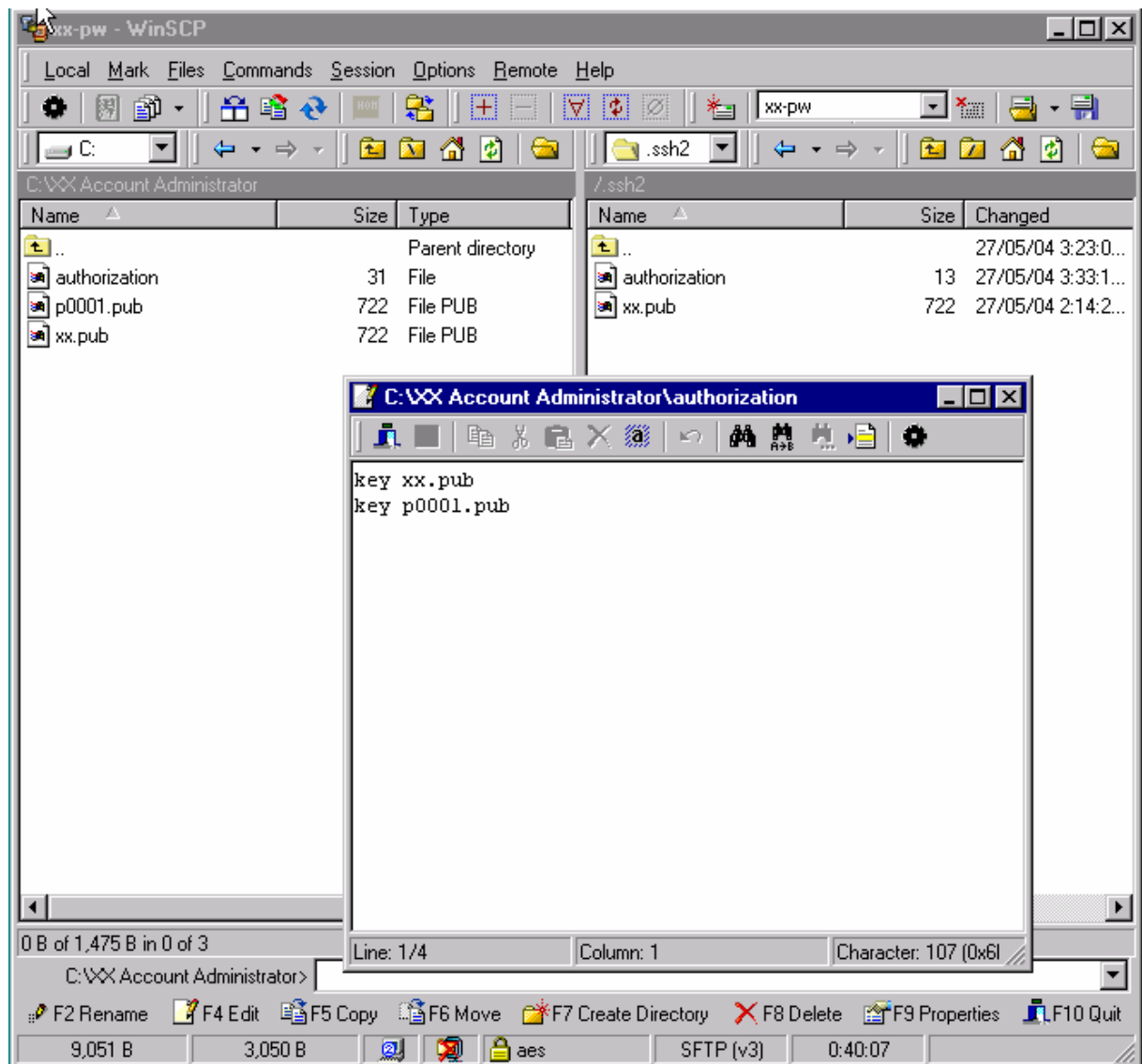
The addition and deletion of users is simply a function of uploading a SECSH-format public key for each new user, and editing the “authorization” file in the .ssh2 directory. Earlier sections of this document made only passing reference to the “authorization” file as it is pre-configured to refer the SFTP server to the “xx.pub” public key.

To manage multiple users, the account administrator should create a local “mirror” of the remote “.ssh2” directory. In this example, there is a folder on the local machine called “XX Account Administrator” which shall be used to store copies of the user public keys and the authorization file. The account administrator must also develop a naming system for public keys that allow them to be associated with an individual or machine. The account administrator should recall that any authorized user can view and manipulate the contents of the “.ssh2” directory for the Office account. Thus, the account administrator may wish to implement a naming scheme that distinguishes automated access accounts from individual accounts. In our example, keys will be differentiated by type (P=Personal, A=Automated) and 4 digit serial number. Example, P0001, P0002, A0003.

1. Begin by logging into the Office account using the WinSCP client. Switch to the .ssh2 remote directory, and the local "XX Account Administrator" folder. Select the two files in the .ssh2 directory and copy them to the local folder.

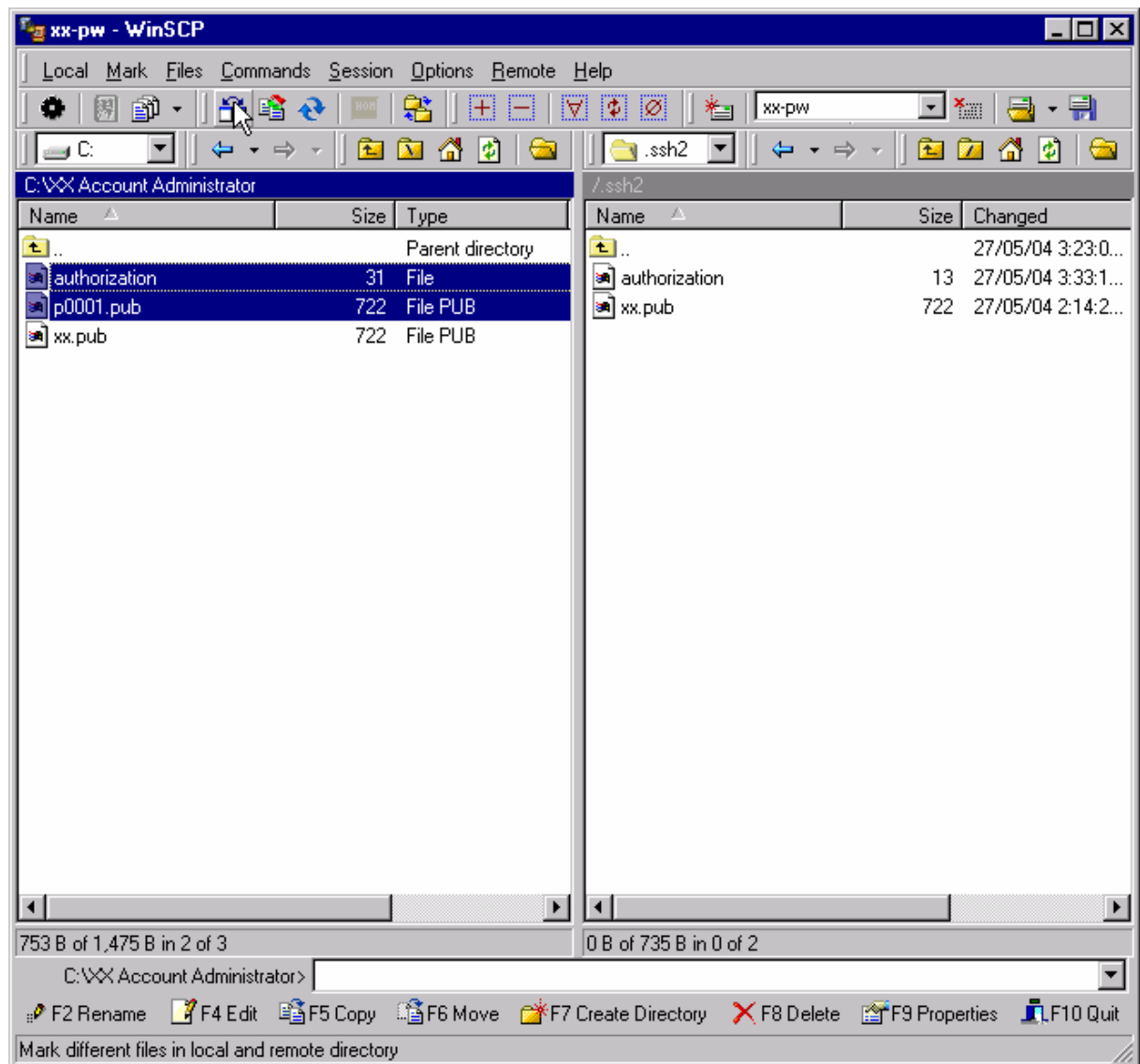


- Then, drag the properly named public key into the local folder. This public key is called p0001.pub. In the local folder, open the “authorization” file by highlighting it and pressing the F4 key. Add the line “key p0001.pub” as shown below.

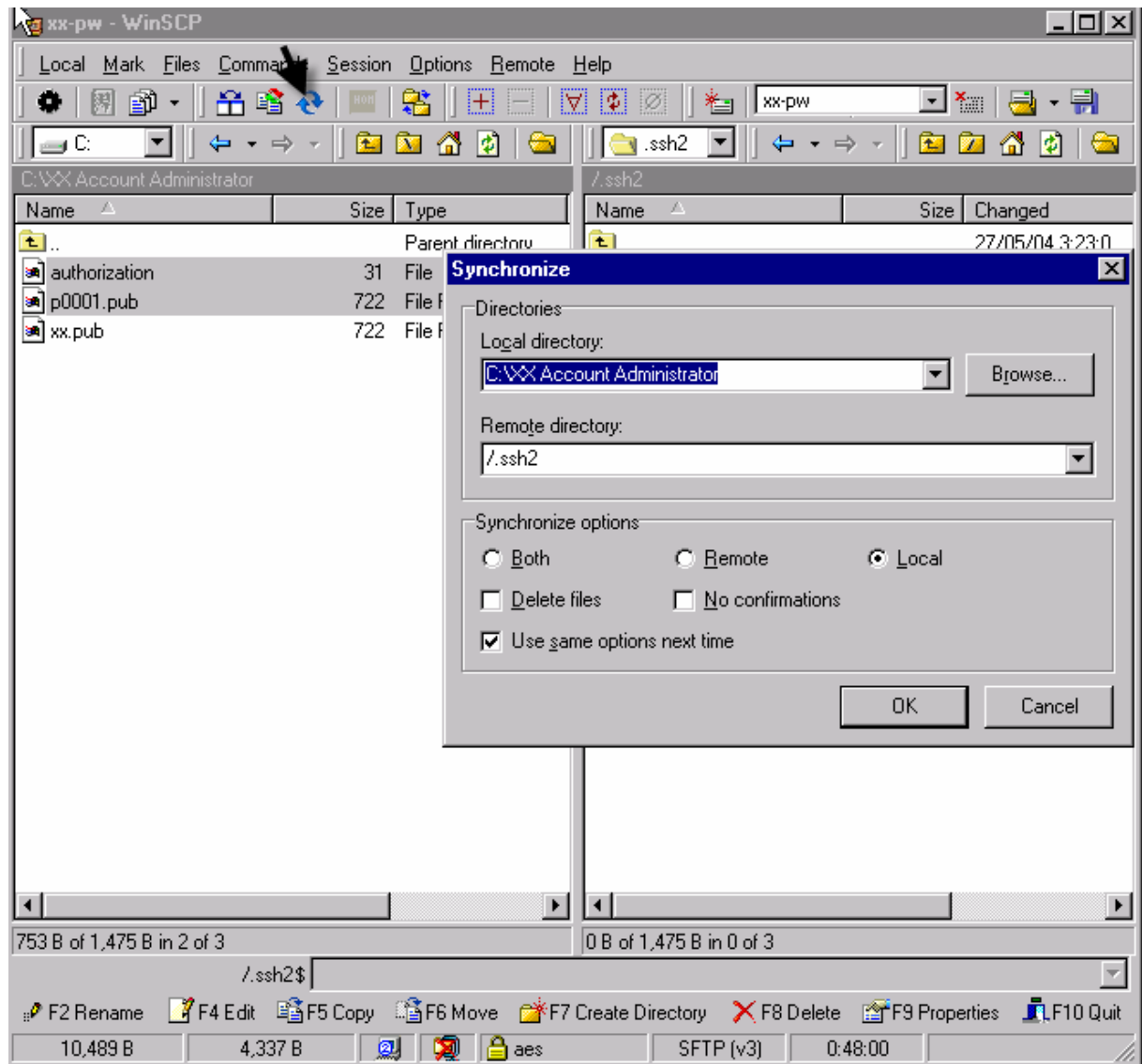


- Save the “authorization” file in the local folder by clicking on the small diskette icon in the editing window. Your local folder now contains two public keys “xx.pub” and “p0001.pub”. Keep track of which person or machine is assigned to which public key!

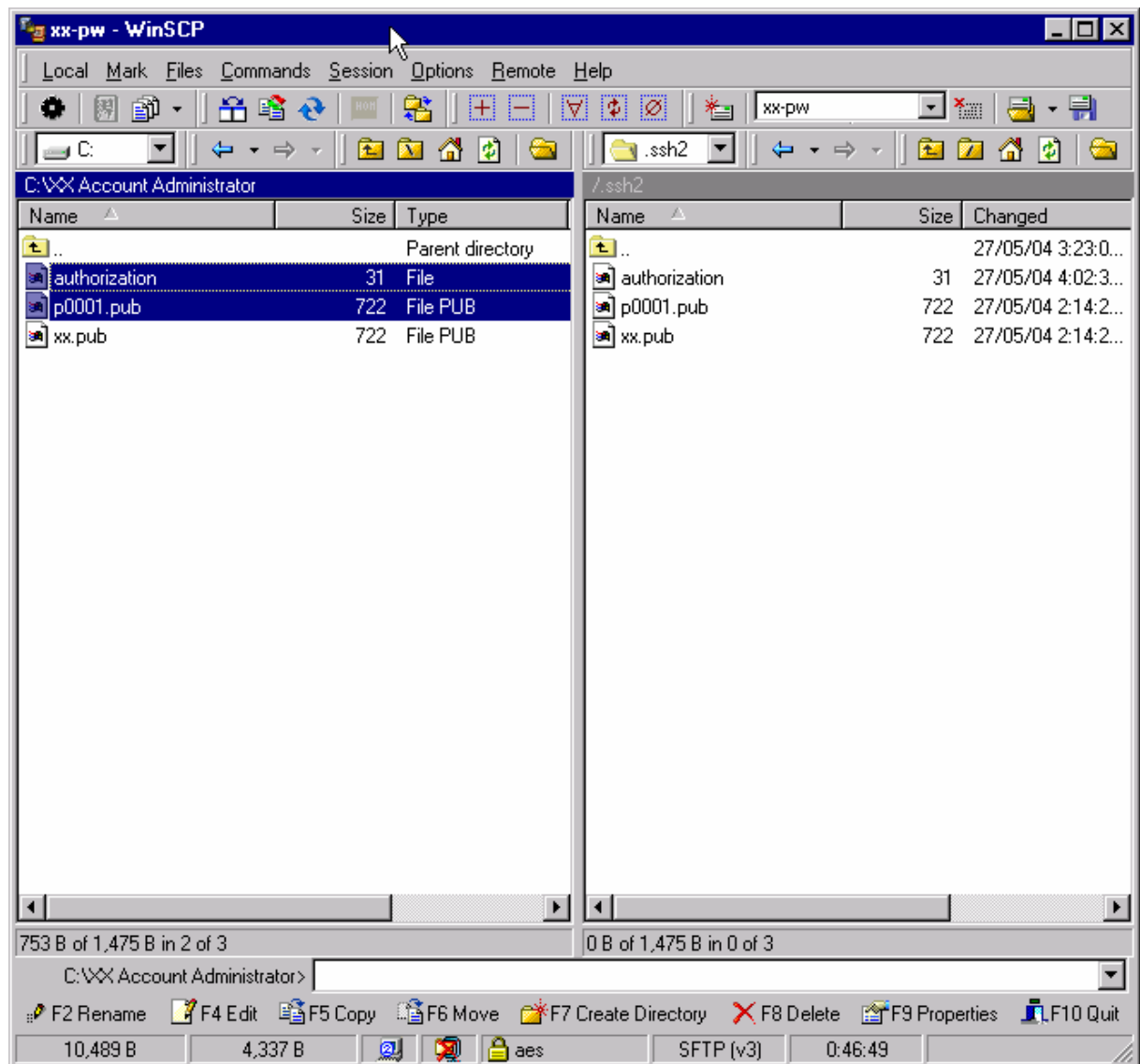
4. When all new keys have been added, click on the “Compare Directories” button as indicated by the arrow below. The changed files will be highlighted.



- Then, click on the “Update Directories” button (see the black arrow). Deselect the “No confirmations” box. Then click on “OK”.



The .ssh2 directory will be updated with the new p0001.pub key and the edited “authorization” file.



6. The account administrator has now installed the new user public key and authorized the system to use the key. A duplicate of the entire .ssh2 directory is stored on the local machine.
7. To remove a user, simply remove the public key and remove the appropriate “key” line from the “authorization” file in the local directory, and repeat steps 4 and 5.

5.3 DEVELOPING CUSTOMIZED APPLICATIONS

If your Office prefers to develop customized software for accessing the PCT-EDI service, this is easily accomplished. Presented is an example of a simple demonstration application written in Perl⁸. Perl is a multiplatform development environment ideally suited for the development of portable code.

Perl itself and all of the modules referenced in this section are freely available. To use this example, you must have an up-to-date version of Perl with the Net::SFTP and Net::SSH-Perl modules referred to in an earlier section installed.

With these modules installed, try the following example. You must first have set up a correct OpenSSH user installation as described above, with working public key authentication. As with the other examples, replace “xx” with your account name. This demonstration is based upon sample code provided with the Net::SFTP module, and will simply print a directory listing of the user account, and then exit. As outlined in the Net::SFTP documentation (see the footnote above for the URL), the functions to Get a file from the server and Put a file on the server are almost identical to the Ls request below.

⁸ <http://www.perl.org>

5.4 EXAMPLE PERL SOURCE CODE

```
#!/usr/bin/perl -w

#####
#
# demo1.pl - demonstrate the basic concepts of using the
# Net::SFTP Perl module with the WIPO PCT-EDI server.
#
# Makes a connection and returns a directory listing
# Requires a properly configured OpenSSH public key authentication
# environment for user "xx", where xx is the account name.
#
# May 27, 2004  Jim Fullton
#
# Arguments:  demo1.pl -v -u=username
# Both are optional.  If -u username is not provided
# the current username for the account is used
#
# You must be using public key encryption for this example
# to work.  This is the first of many examples.
#
# ./demo1.pl ptedi.wipo.int -u xx
#
#####

use strict;

use Net::SFTP;
use Getopt::Long;

my %opts;
my $user;
Getopt::Long::Configure('no_ignore_case');
GetOptions(\%opts, "v", 'u=s'=>\$user);

my($host) = @ARGV;
die "usage: demo1 [options] hostname" unless $host;

# set up the arguments based on the command line options
my %args = (ssh_args => []);
$args{debug} = 1 if $opts{v};
```



```
push @{ $args{ssh_args} }, user => $user ;

# make our connection

print "Connecting to $host...\n";

my $sftp = Net::SFTP->new($host, %args);

# get a listing of the base directory

$sftp->ls(".", sub { print $_[0]->{longname}, "\n" });

print "Finished\n";
```