2-1-1 ssh
Secure SHell

Using Public Key Cryptography

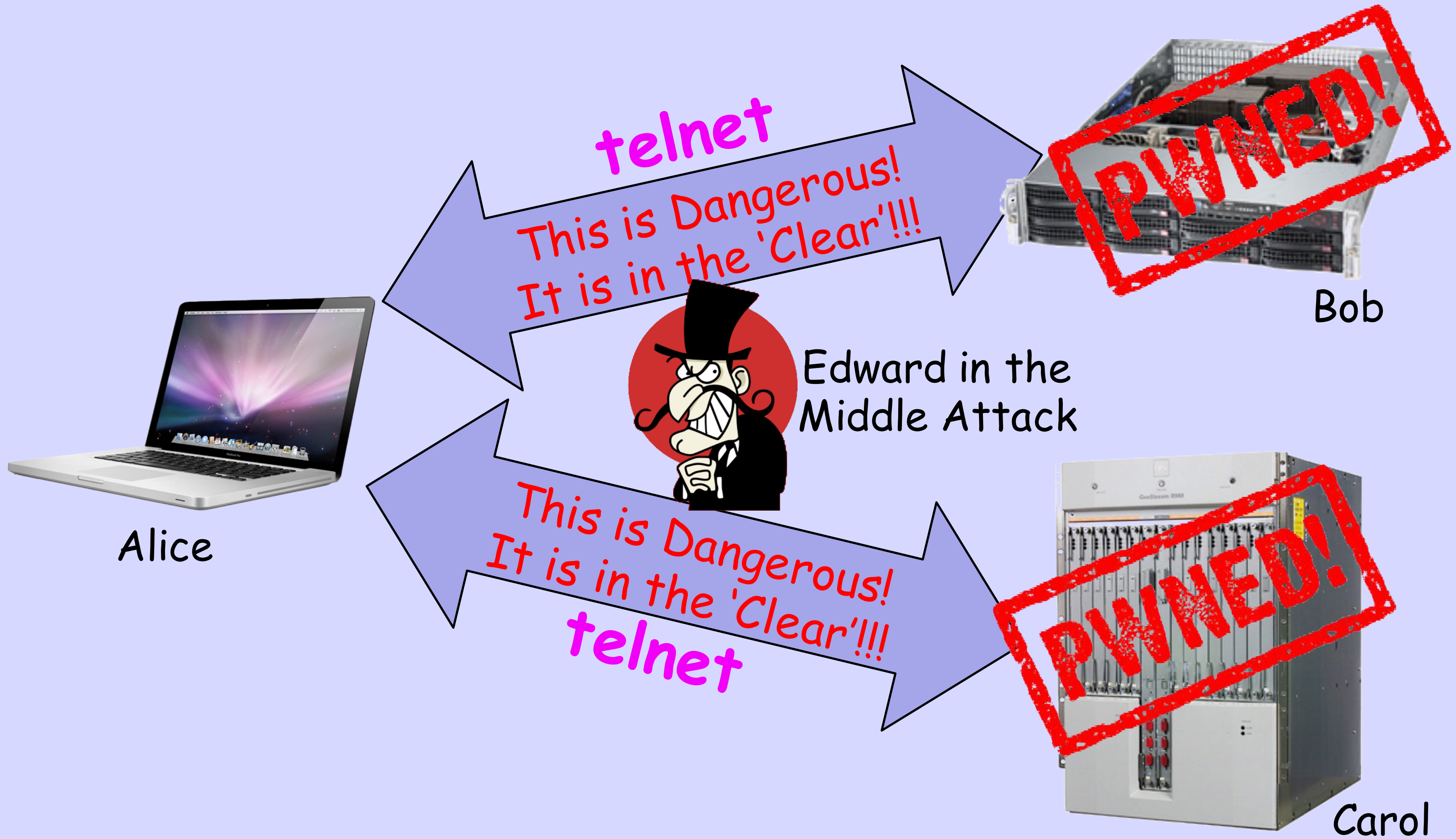
Keying, Key Exchange, and Session Setup

Communicate Safely with Remote Systems

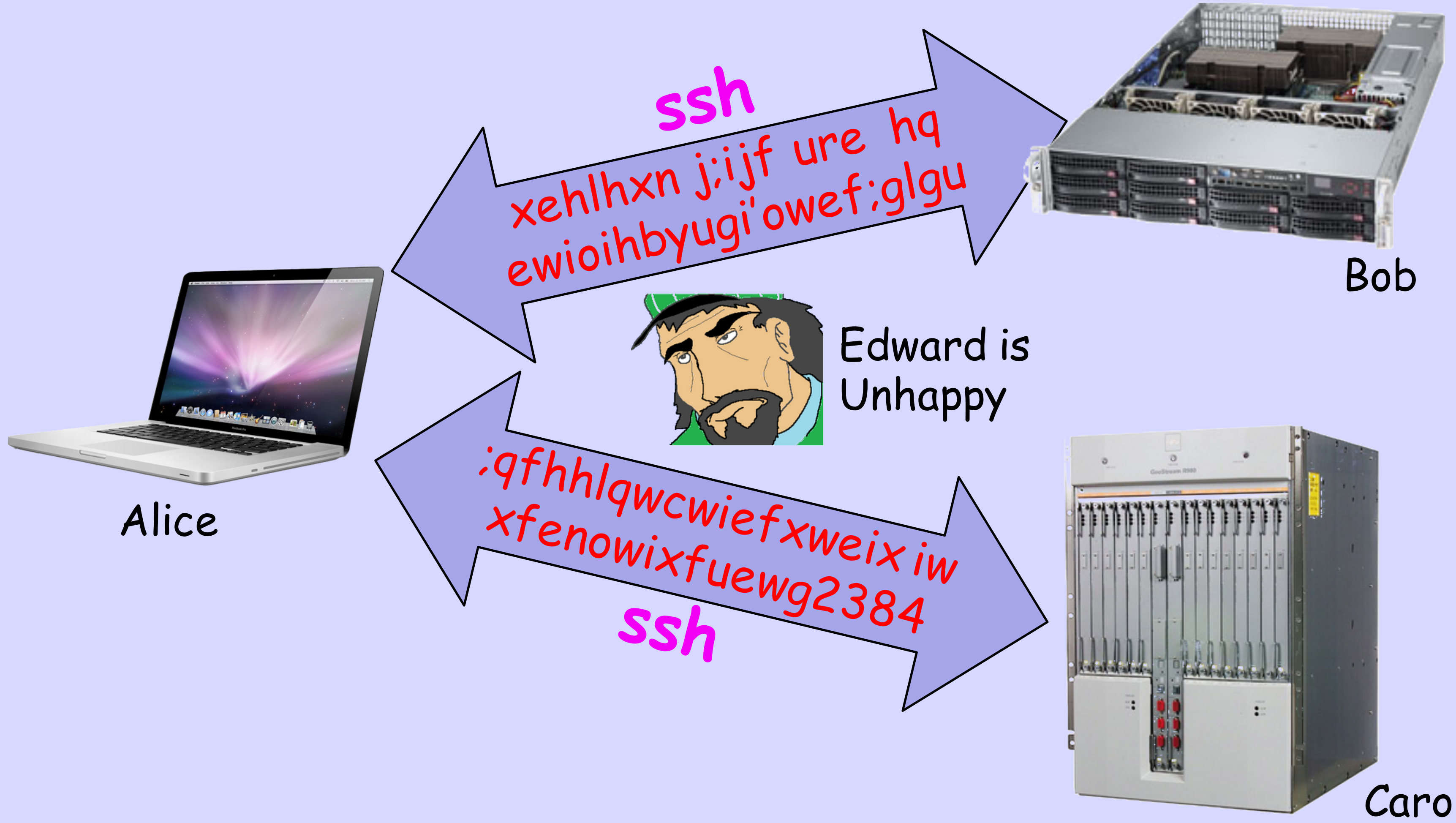
What is “Safely”

- Authentication - I am Assured of Which Host I am Talking With
- Authentication - The Host Knows Who I Am
- Privacy - The Traffic is Encrypted
- Integrity - The Traffic is Unmodified

Traditional



Encrypted


ssh
xehlhxn j;ijf ure hq
ewioihbyugi'owef;glgu
Bob
Edward is Unhappy
Alice
;qfhhlqwcwiefxweix iw
xfenowixfuewg2384
ssh
Carol

Secure SHell

- Provides authenticated and encrypted shell access to a remote host
- But it is much more
- It is used by other protocols, sftp, scp, rsync, …
- You can use it to build custom tunnels

Think of SSH as a Bit Like PGP where the Other End is a Computer, Not a Human

But PGP is

Object Security

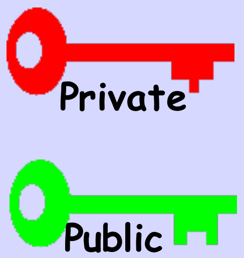
SSH is

Channel/Transport

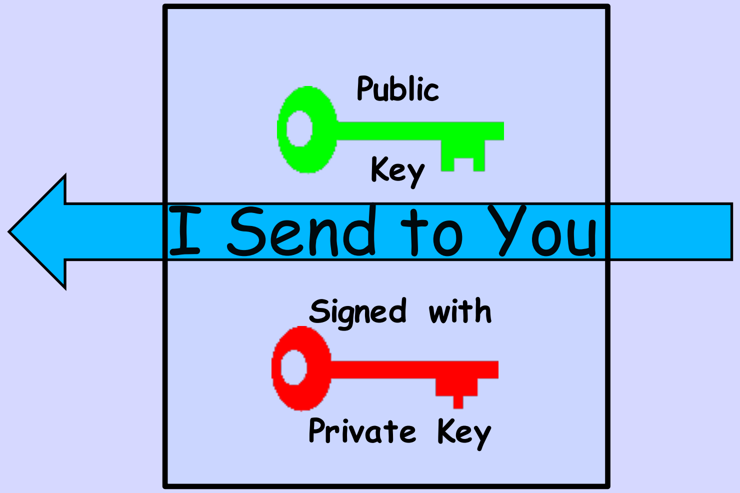
Security

Proof of Possession

If I Have a Key Pair



How Do I Convince You That I Have Both Private and Public Keys Over The Public Net?



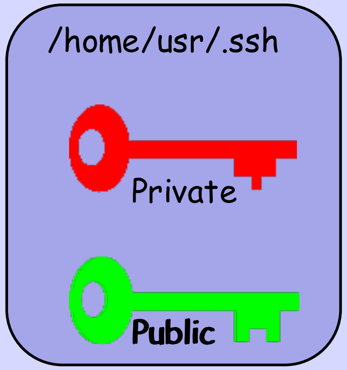
You Verify Signature Using The Public Key

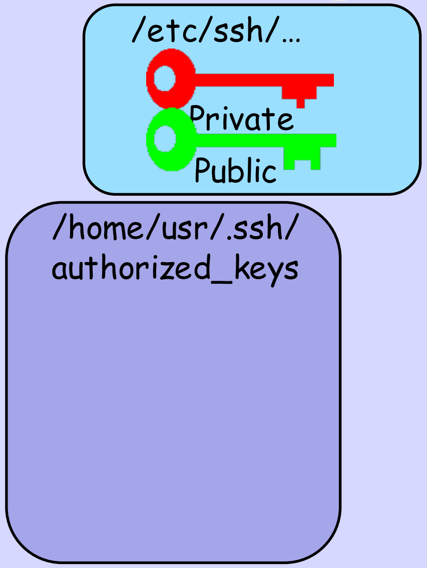
If It Verifies, Then You Know That
I Must Have The Private Key

And You Know You Have the
Corresponding Public Key

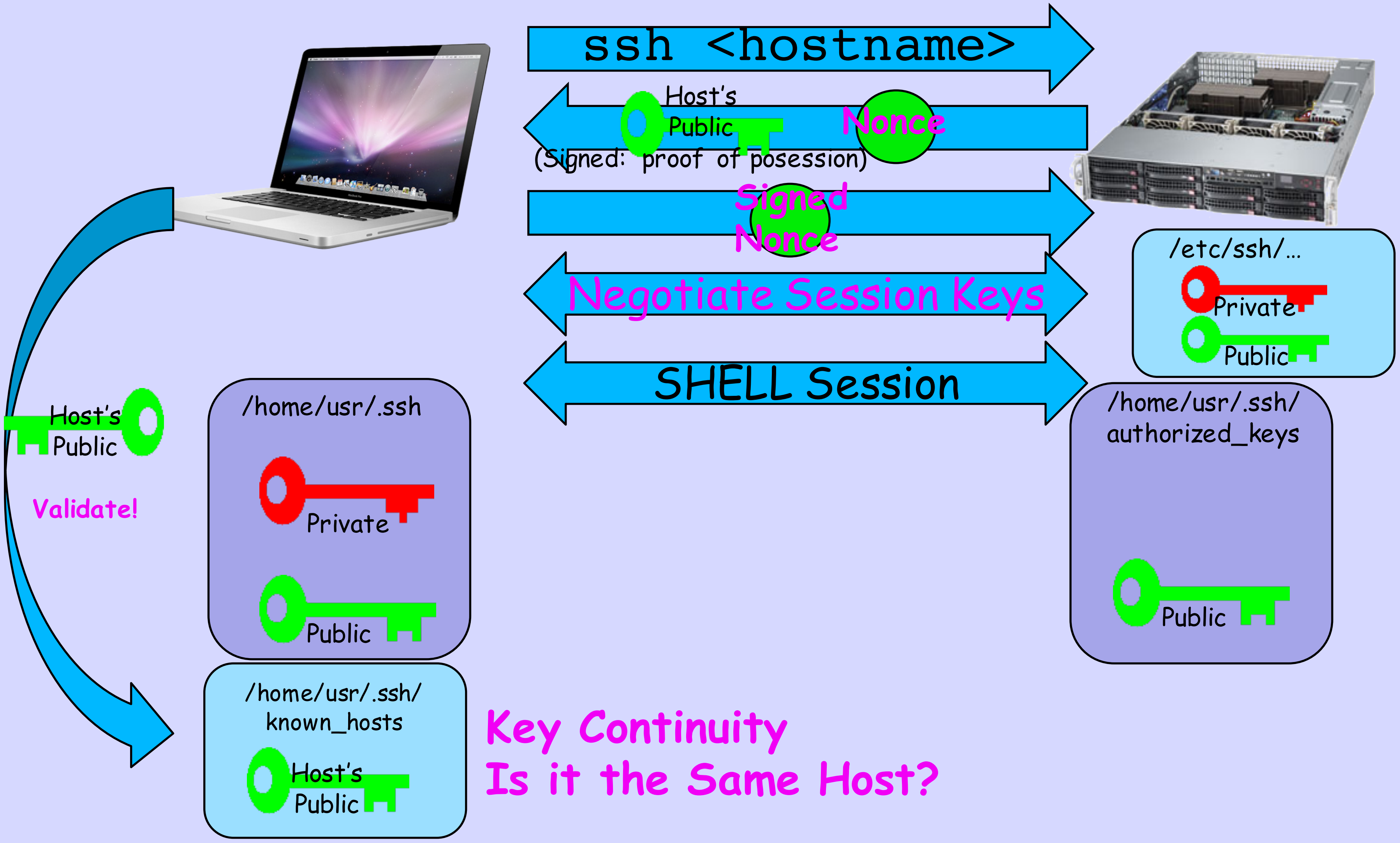
ssh - Keying Setup

```
ssh-keygen -t rsa
```





2-Way Authentication



Key Continuity
Is it the Same Host?

Checking Host's Keys

```
$ ssh -o VisualHostKey=yes psg.com
Host key fingerprint is
d2:2b:f1:17:75:0d:c9:86:74:71:e2:00:62:0f:22:02
+--[ RSA 1024]----+
|E.. . . + .ooo=o.|
|   . . o +  .++= |
|          . ..o .|
|       .   . .   |
|      o S .      |
|       + . .     |
|      . o .      |
|       . .       |
|                 |
+-----------------+
```

And you check it against what you got out of band

ssh-keygen RSA Key

```
/usr/home/foo> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/usr/home/foo/.ssh/id_rsa):
Created directory '/usr/home/foo/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /usr/home/foo/.ssh/id_rsa.
Your public key has been saved in /usr/home/foo/.ssh/id_rsa.pub.
The key fingerprint is:
27:99:35:e4:ab:9b:d8:50:6a:8b:27:08:2f:44:d4:20 foo@psg.com
The key's randomart image is:
+--[ RSA 2048]----+
|E.o       .      |
|.. .     o       |
|.         +      |
| .       + o     |
|.       S o      |
|..     o +       |
|.o .  + .        |
|. o .o.= o       |
| .  .oo +        |
+-----------------+
```

Eliptical Curve Key

```
/usr/home/foo> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/randy/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/randy/.ssh/id_ed25519.
Your public key has been saved in /home/randy/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:QdV6B3SOwlndmvJXoatyS0xOFP+aWT1R3JFHAcT1Q2I randy@ran.psg.com
The key's randomart image is:
+--[ED25519 256]--+
|        ...+=E+OB|
|       . . o*=+oB|
|        . +o.oo=+|
|         .o.o.= =|
|        S  + +.+o|
|          =  .* o|
|           +.+ . |
|         ..o     |
|          o..    |
+----[SHA256]-----+
```

ssh-keygen - sshv1 key

```
/usr/home/foo> ssh-keygen -t rsa1
Generating public/private rsa1 key pair.
Enter file in which to save the key (/usr/home/foo/.ssh/identity):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /usr/home/foo/.ssh/identity.
Your public key has been saved in /usr/home/foo/.ssh/identity.pub.
The key fingerprint is:
1e:c2:df:cd:54:60:63:24:58:71:1f:ac:36:67:c8:b6 fo@ran.psg.com
The key's randomart image is:
+--[RSA1 2048]----+
|         o+oB..  |
|        .  = +.. |
|           . oo  |
|     .      B.o  |
|      o S  o.=   |
|       + o +E    |
|        o . o    |
|                 |
|                 |
+-----------------+
```

ssh v1 is ONLY for 2511s and other antiques

Use Keys Not Passwords

- In /etc/ssh/sshd_config

```
PermitRootLogin without-password
PasswordAuthentication no
UsePAM no
```

- Never Store Private Key on a Multi-User Host
- Store Private Key ONLY on Your Laptop and Protect Your Laptop (Encrypt Disk!)
- It is OK to Use SSH_AGENT to Remember your Key ONLY if your Laptop Locks Very Quickly

The Only Compromise I Have Had to My Infrastructure was a Researcher who Stored Their Private Key on a Shared University Host

Private Key Protection

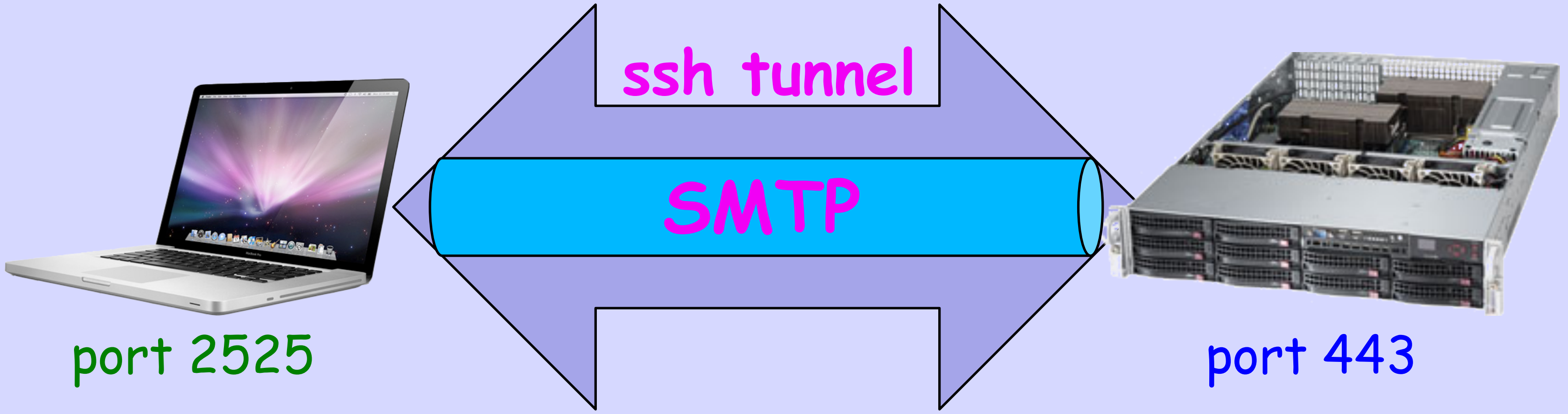
- FreeBSD Repository Compromise Six Years Ago

  “The compromise is believed to have occurred due to the leak of an SSH key from a developer who legitimately had access to the machines in question, and was not due to any vulnerability or code exploit within FreeBSD.”

General Purpose Tunnel

- I am in my hotel room and want to send mail from my laptop
- I do not want unencrypted mail going over the net
- So I want the SMTP traffic to be encrypted to my SMTP server
- I own the SMTP server

General Purpose Tunnel



```
$ ssh -N ssh.psg.com -p 443 -L 2525:127.0.0.1:25
```

Target Host

Tunnel Port

Port on MacBook

Tunnel EndPoint

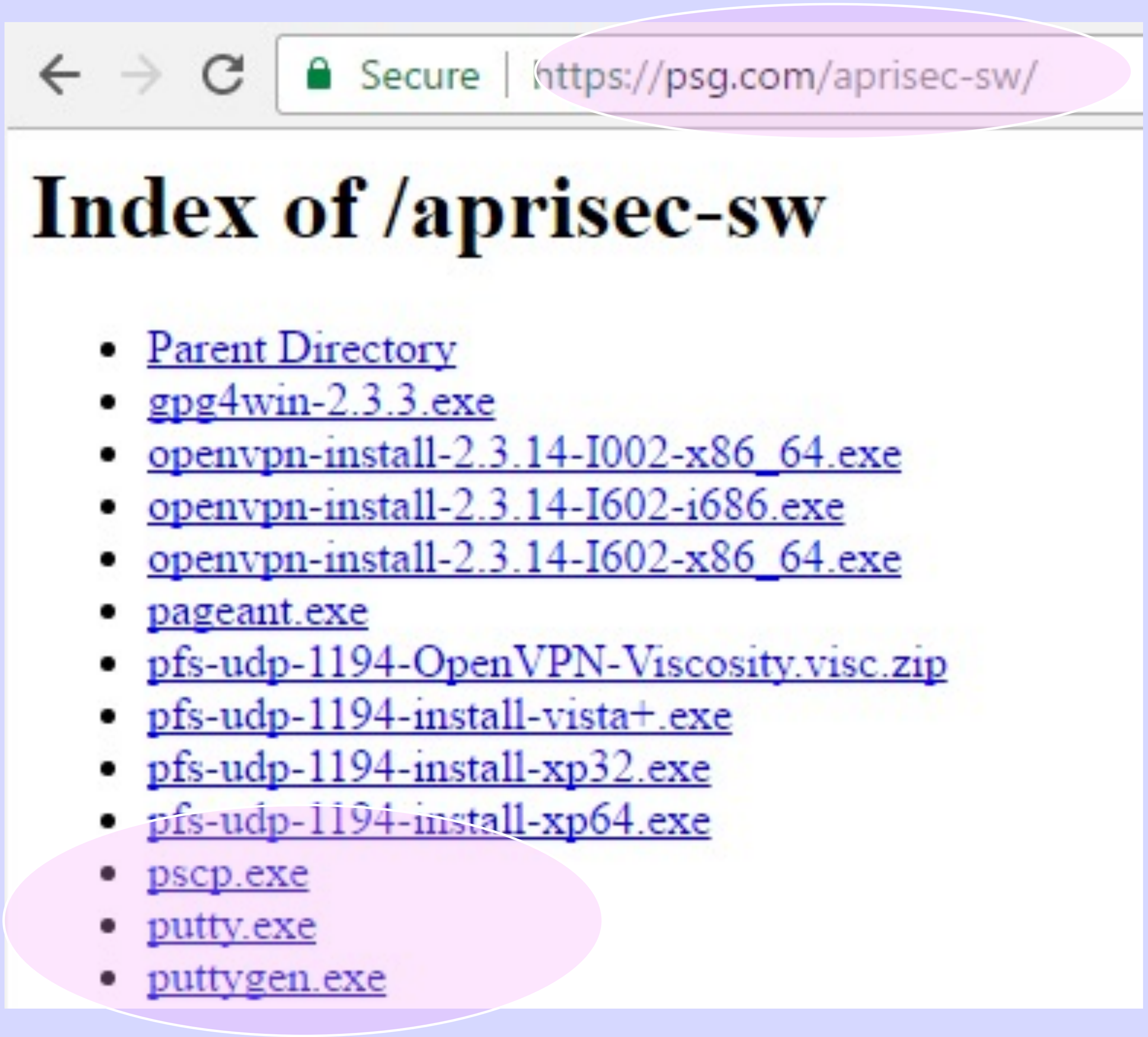
SSH is Built In

UNIX

Linux

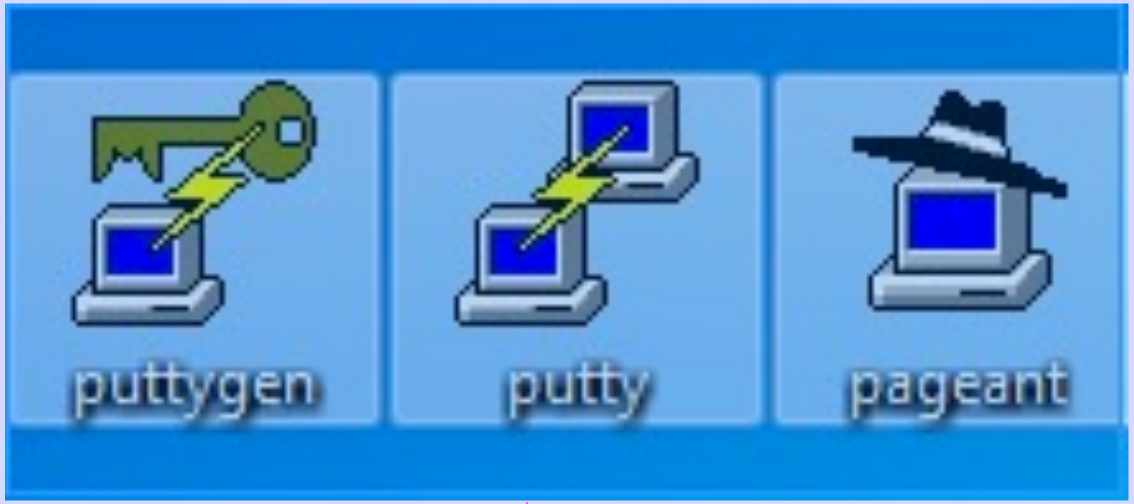
MacOS X

Get Software



Index of /aprisec-sw

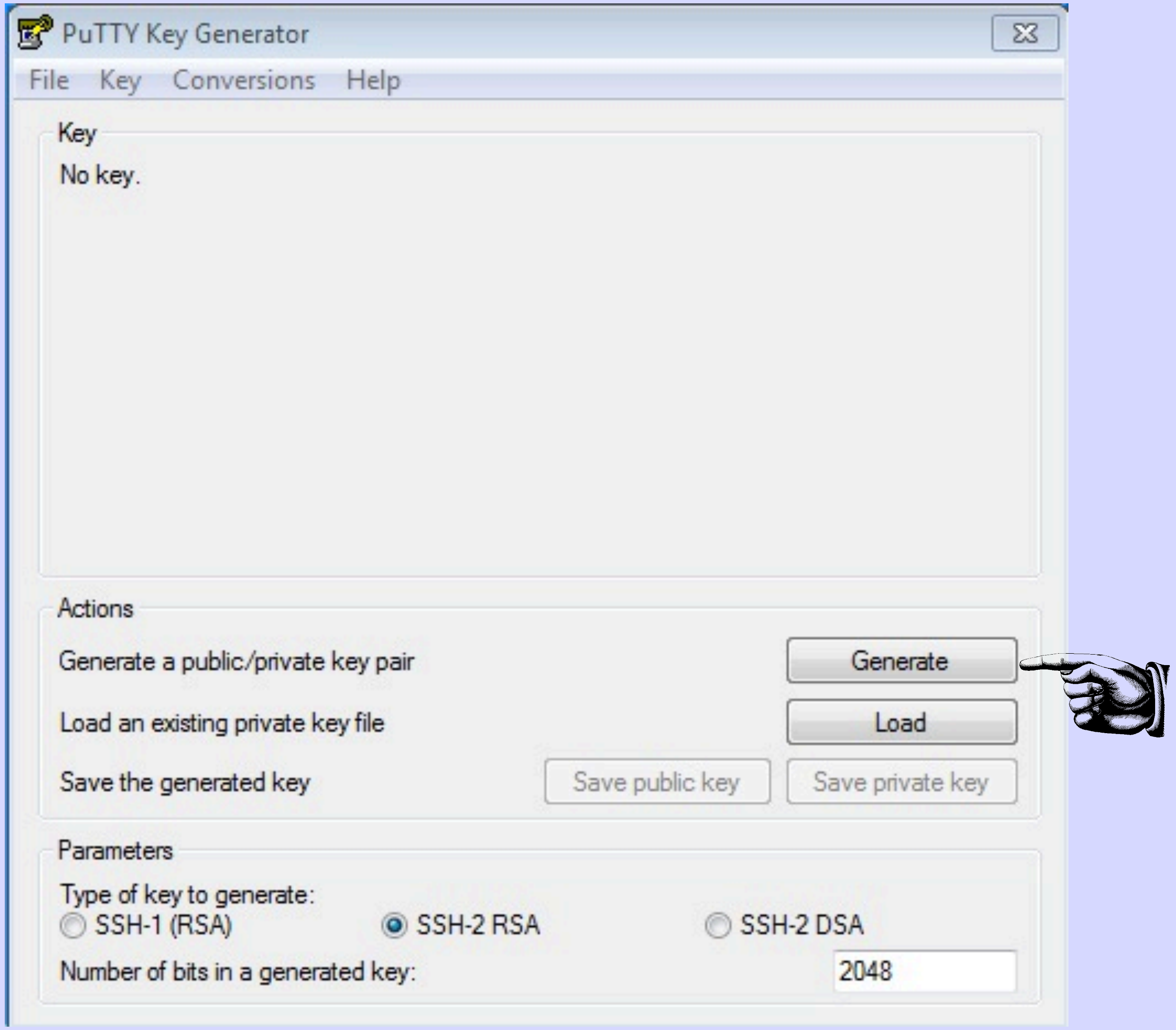
- Parent Directory
- gpg4win-2.3.3.exe
- openvpn-install-2.3.14-I002-x86_64.exe
- openvpn-install-2.3.14-I602-i686.exe
- openvpn-install-2.3.14-I602-x86_64.exe
- pageant.exe
- pfs-udp-1194-OpenVPN-Viscosity.visc.zip
- pfs-udp-1194-install-vista+.exe
- pfs-udp-1194-install-xp32.exe
- pfs-udp-1194-install-xp64.exe
- pscp.exe
- putty.exe
- puttygen.exe



Access Remote System

Generate Your Keys

Generate Key Pair



PuTTY Key Generator

Key

No key.

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

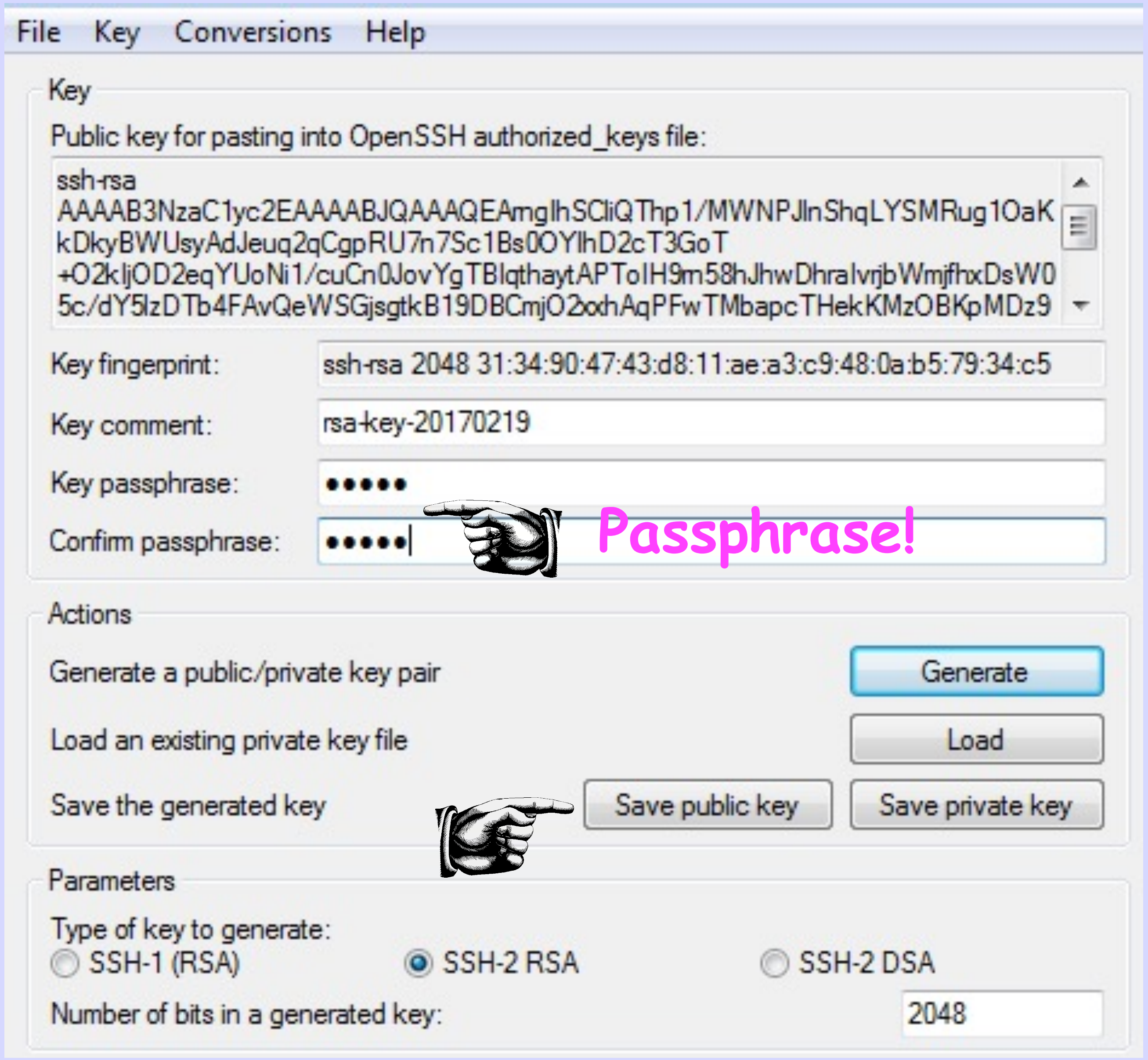
Parameters

Type of key to generate:

SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

Number of bits in a generated key: 2048

Save Public Key



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAmglhSCliQThp1/MWNPJlnShqLYSMRug1OaK
kDkyBWUsyAdJeuq2qCgpRU7n7Sc1Bs0OYlhD2cT3GoT
+O2kljOD2eqYUoNi1/cuCn0JovYgTBlqthaytAPToIH9m58hJhwDhralvrjbWmjfhxDsW0
5c/dY5lzDTb4FAvQeWSGjsgtkB19DBCmjO2oxhAqPFwTMbapcTHekKMzOBKpMDz9
```

Key fingerprint: ssh-rsa 2048 31:34:90:47:43:d8:11:ae:a3:c9:48:0a:b5:79:34:c5

Key comment: rsa-key-20170219

Key passphrase: •••••

Confirm passphrase: •••••

Actions

Generate a public/private key pair

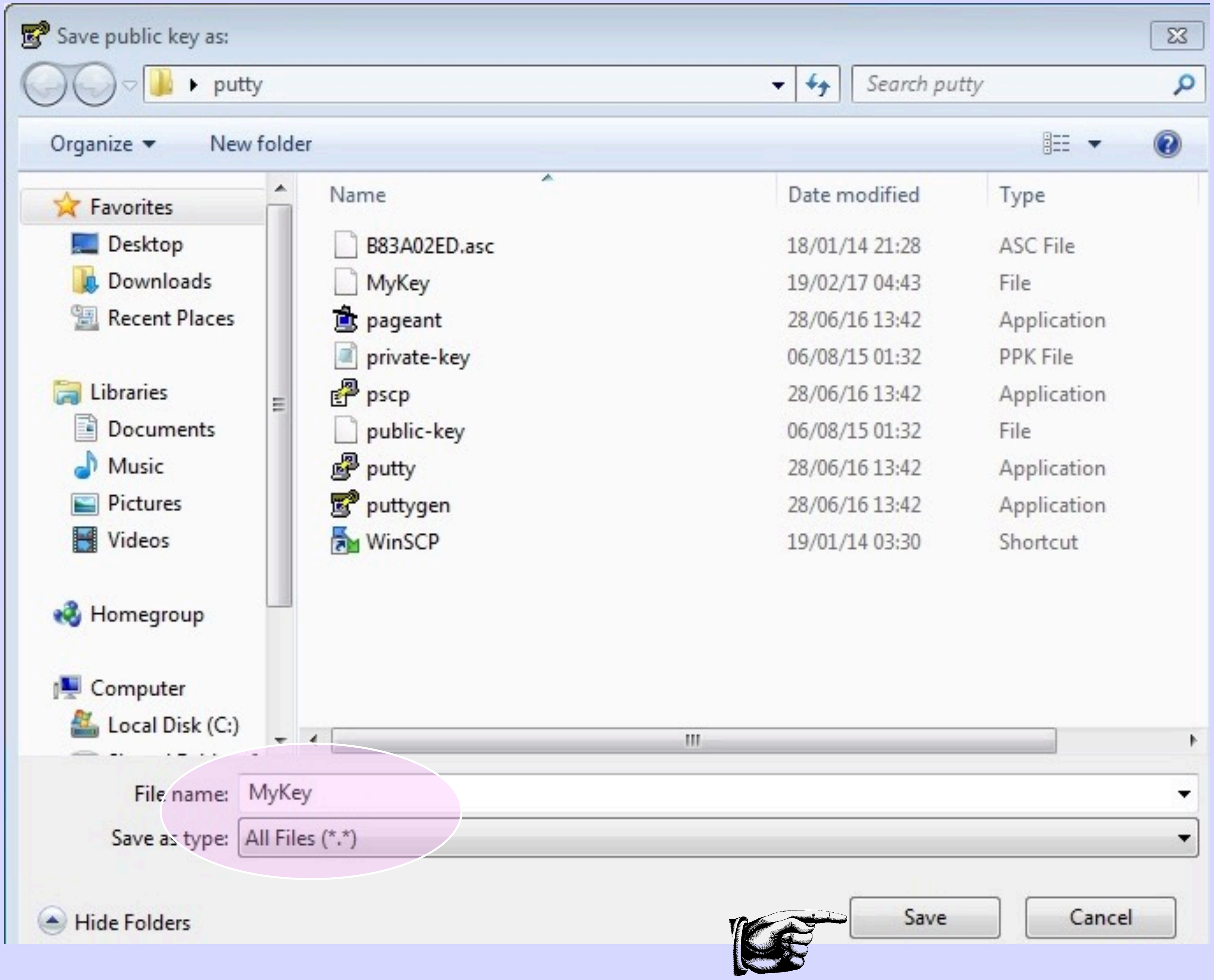
Load an existing private key file

Save the generated key

Parameters

Type of key to generate:
SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

Number of bits in a generated key: 2048


Save public key as:

putty

Search putty

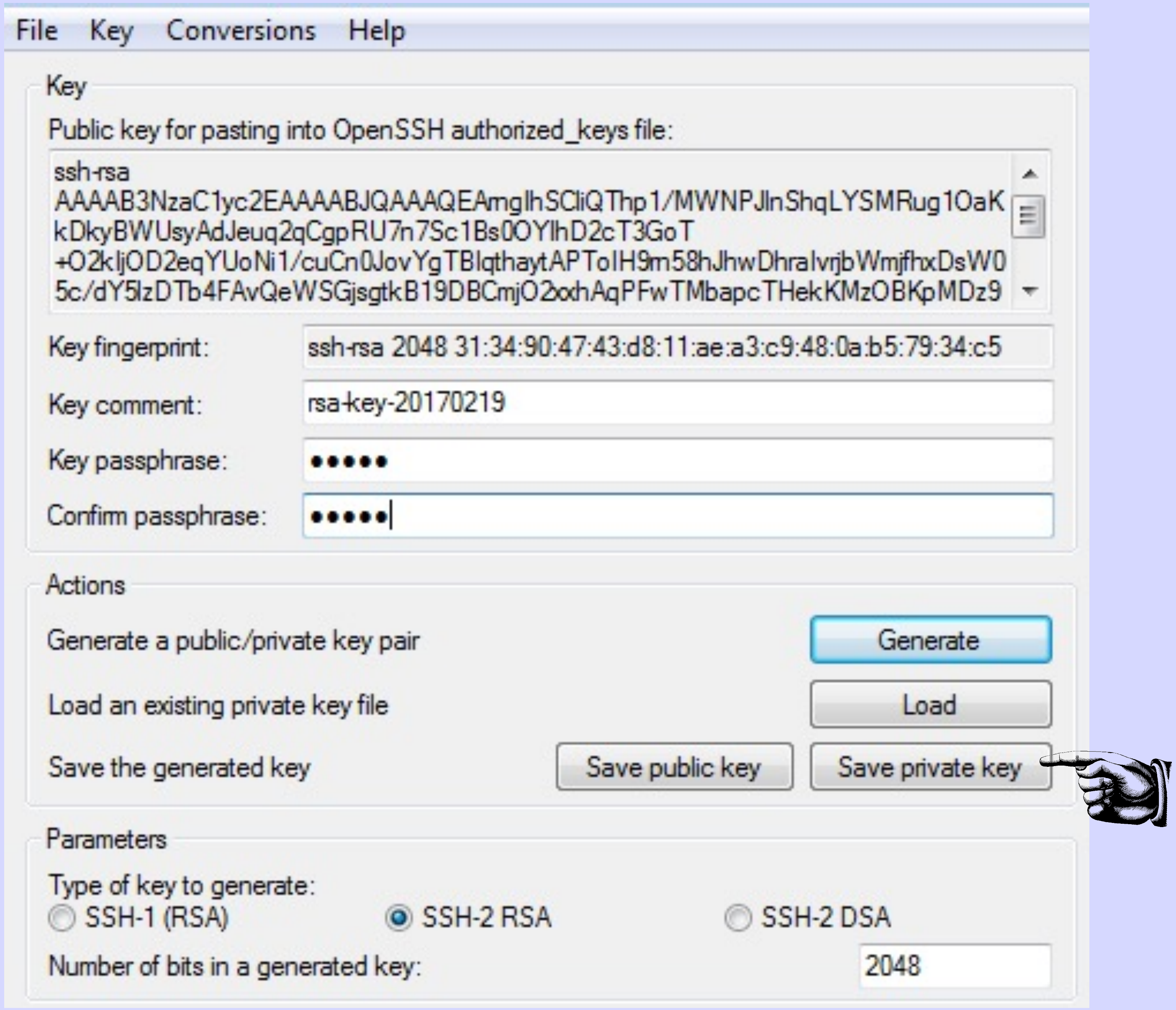
Organize ▾ New folder

Name	Date modified	Type
B83A02ED.asc	18/01/14 21:28	ASC File
MyKey	19/02/17 04:43	File
pageant	28/06/16 13:42	Application
private-key	06/08/15 01:32	PPK File
pscp	28/06/16 13:42	Application
public-key	06/08/15 01:32	File
putty	28/06/16 13:42	Application
puttygen	28/06/16 13:42	Application
WinSCP	19/01/14 03:30	Shortcut

File name: MyKey

Save as type: All Files (*.*)

Save Private Key



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAmglhSCliQThp1/MWNPJlnShqLYSMRug1OaK
kDkyBWUsyAdJeuq2qCgpRU7n7Sc1Bs0OYlhD2cT3GoT
+O2kIjOD2eqYUoNi1/cuCn0JovYgTBlqthaytAPToIH9m58hJhwDhralvrjbWmjfhxDsW0
5c/dY5lzDTb4FAvQeWSGjsgtkB19DBCmjO2oxhAqPFwTMbapcTHekKMzOBKpMDz9
```

Key fingerprint: ssh-rsa 2048 31:34:90:47:43:d8:11:ae:a3:c9:48:0a:b5:79:34:c5

Key comment: rsa-key-20170219

Key passphrase: •••••

Confirm passphrase: •••••

Actions

Generate a public/private key pair

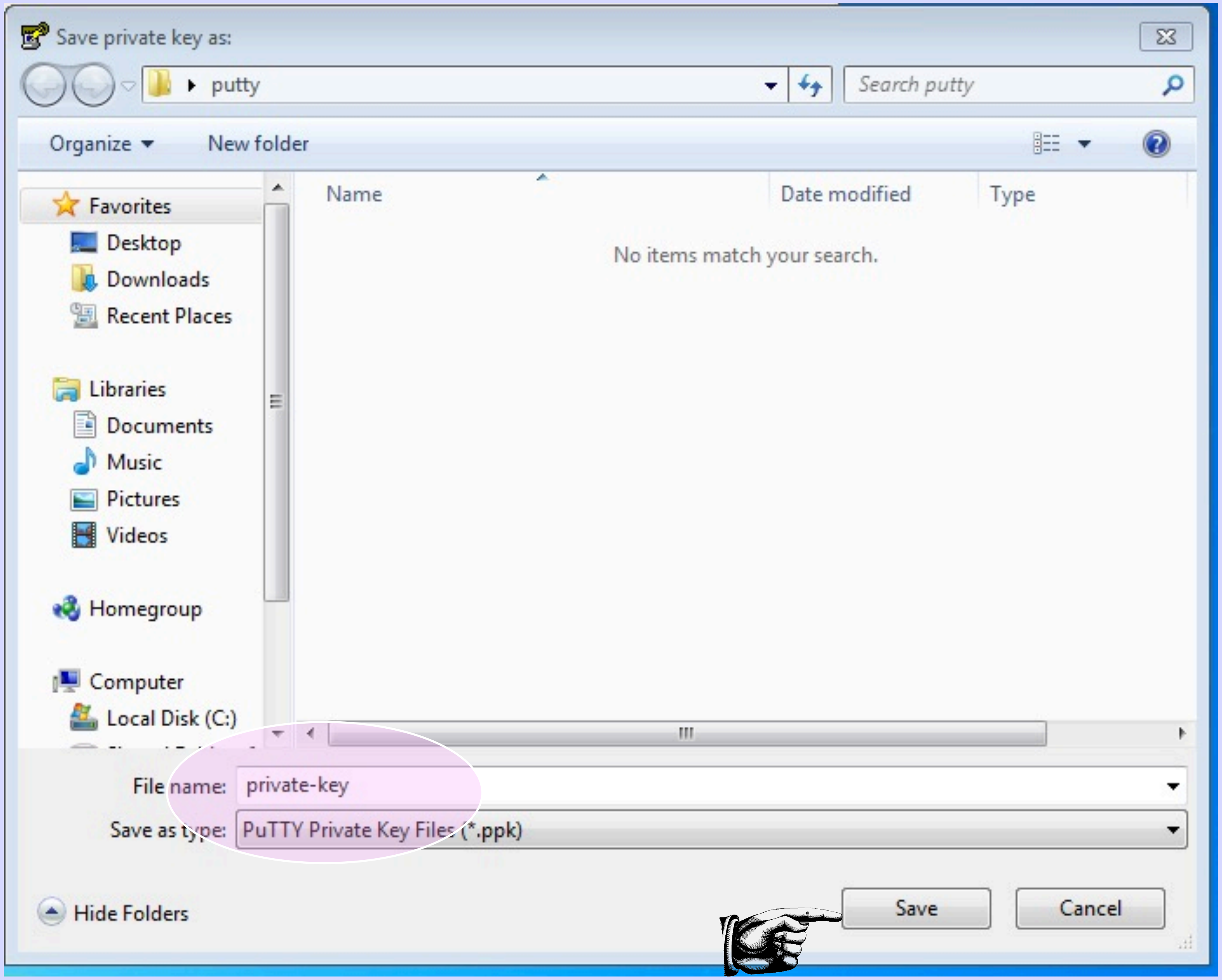
Load an existing private key file

Save the generated key

Parameters

Type of key to generate:
SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

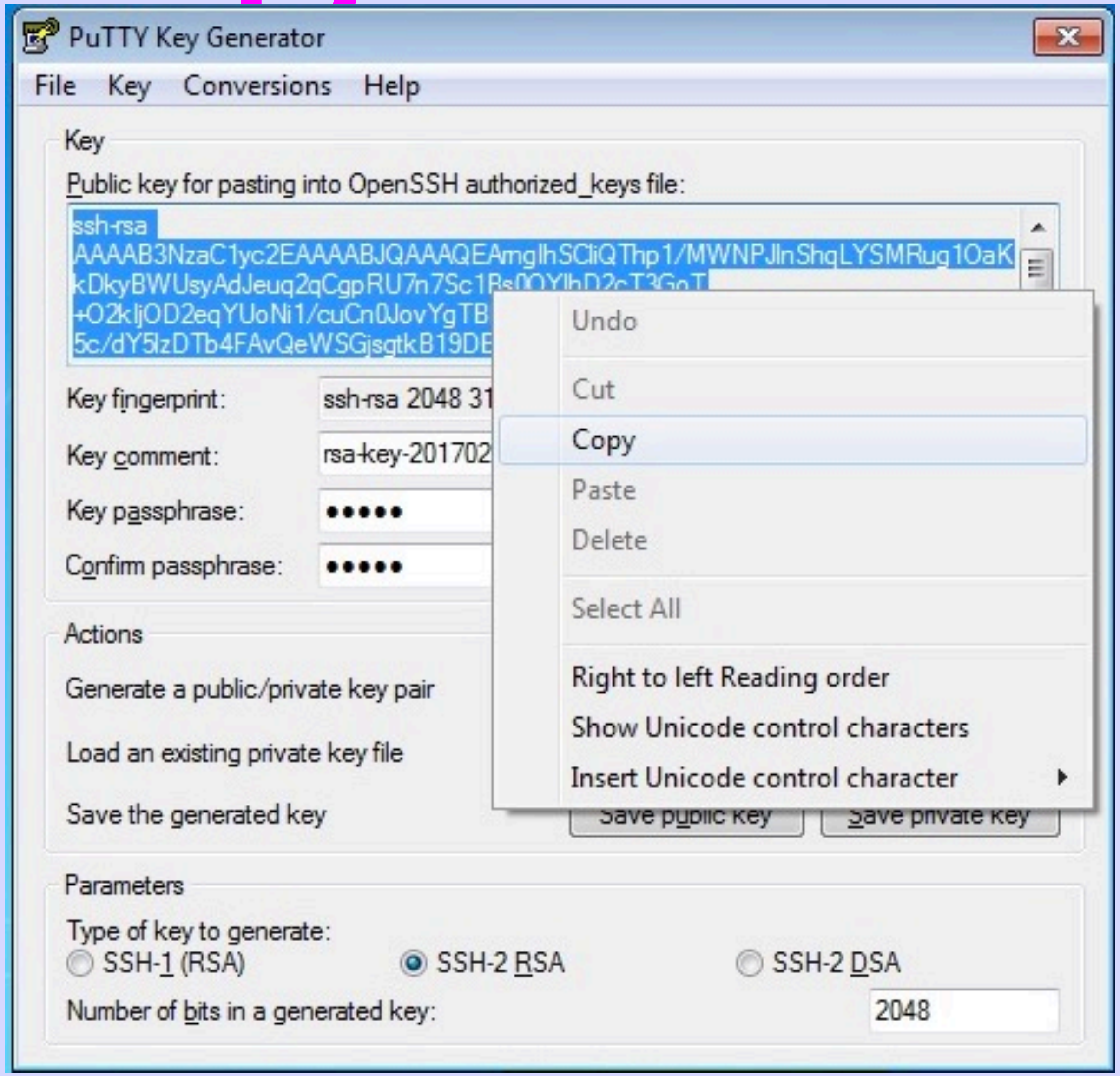
Number of bits in a generated key: 2048



File name: private-key

Save as type: PuTTY Private Key Files (*.ppk)

Copy Public Key



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAmglhSCliQThp1/MWNPJlnShqLYSMRug1OaK
kDkyBWUsyAdJeuq2qCgpRU7n7Sc1Bs0OYlhD2cT3GoT
+O2kIjOD2eqYUoNi1/cuCn0JovYgTB
5c/dY5lzDTb4FAvQeWSGjsgtkB19DE
```

Key fingerprint: ssh-rsa 2048 31

Key comment: rsa-key-201702

Key passphrase: •••••

Confirm passphrase: •••••

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Parameters

Type of key to generate:

SSH-1 (RSA) [✓ SSH-2 RSA] SSH-2 DSA

Number of bits in a generated key: 2048

Putting the Key on the Target Host

Now, you need to paste the copied public key in the file ~/.ssh/authorized_keys on your server.

Log in to your destination server using putty with username **apricot** / password **_ask_instructor_**

If your SSH folder does not yet exist, create it manually:

```
$ mkdir ~/.ssh
$ chmod 0700 ~/.ssh
$ touch ~/.ssh/authorized_keys
$ chmod 0644 ~/.ssh/authorized_keys
```

Putting the Key on the Target Host

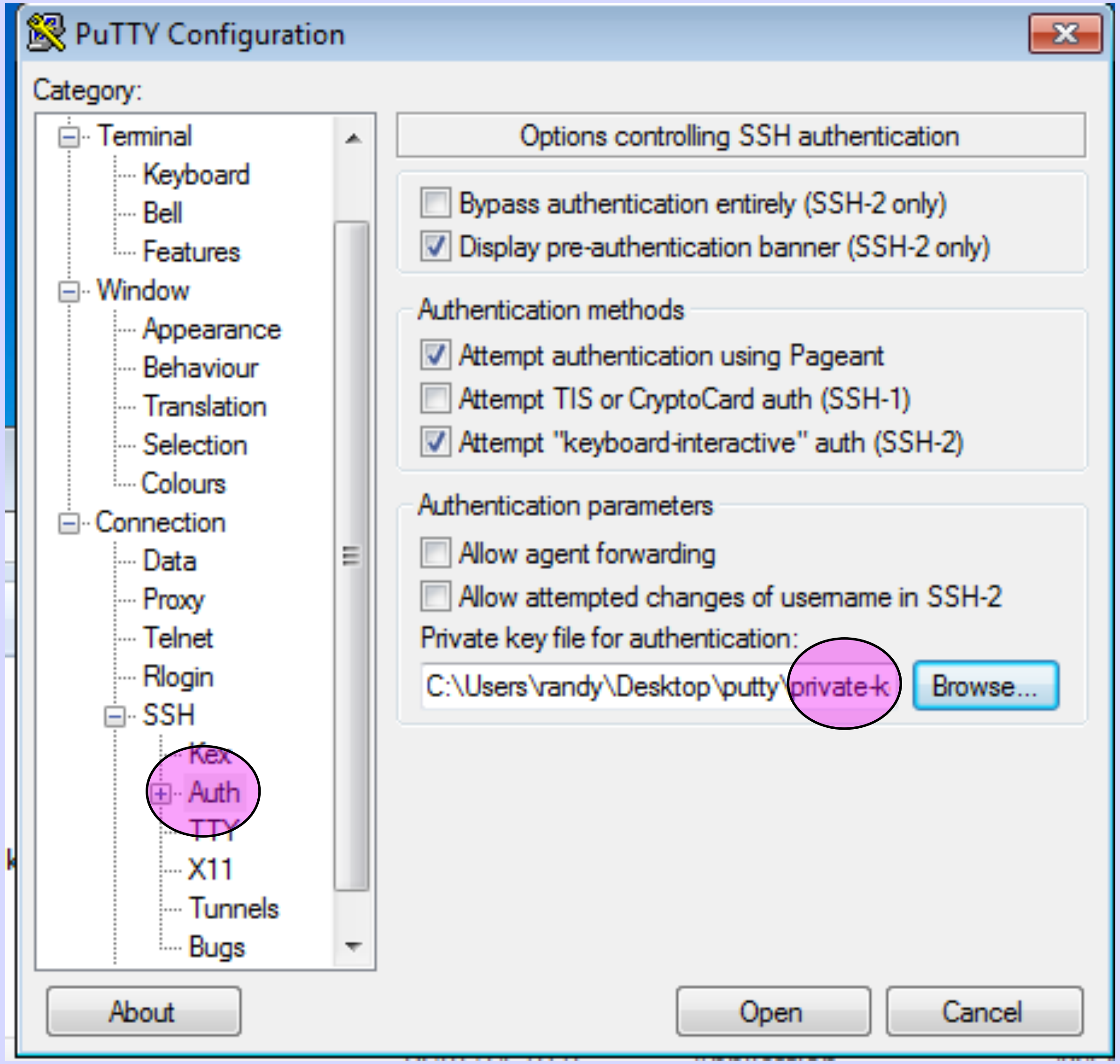
Paste the SSH public key into your ~/.ssh/authorized_keys file:

```
$ vi ~/.ssh/authorized_keys
```

Tap the i key on your keyboard & right-click your mouse to paste.

To save, tap the following keys on your keyboard (in this order): Esc, :wq Enter.

Load Key in Putty



Options controlling SSH authentication

- [ ] Bypass authentication entirely (SSH-2 only)
- [✓] Display pre-authentication banner (SSH-2 only)

Authentication methods

- [✓] Attempt authentication using Pageant
- [ ] Attempt TIS or CryptoCard auth (SSH-1)
- [✓] Attempt "keyboard-interactive" auth (SSH-2)

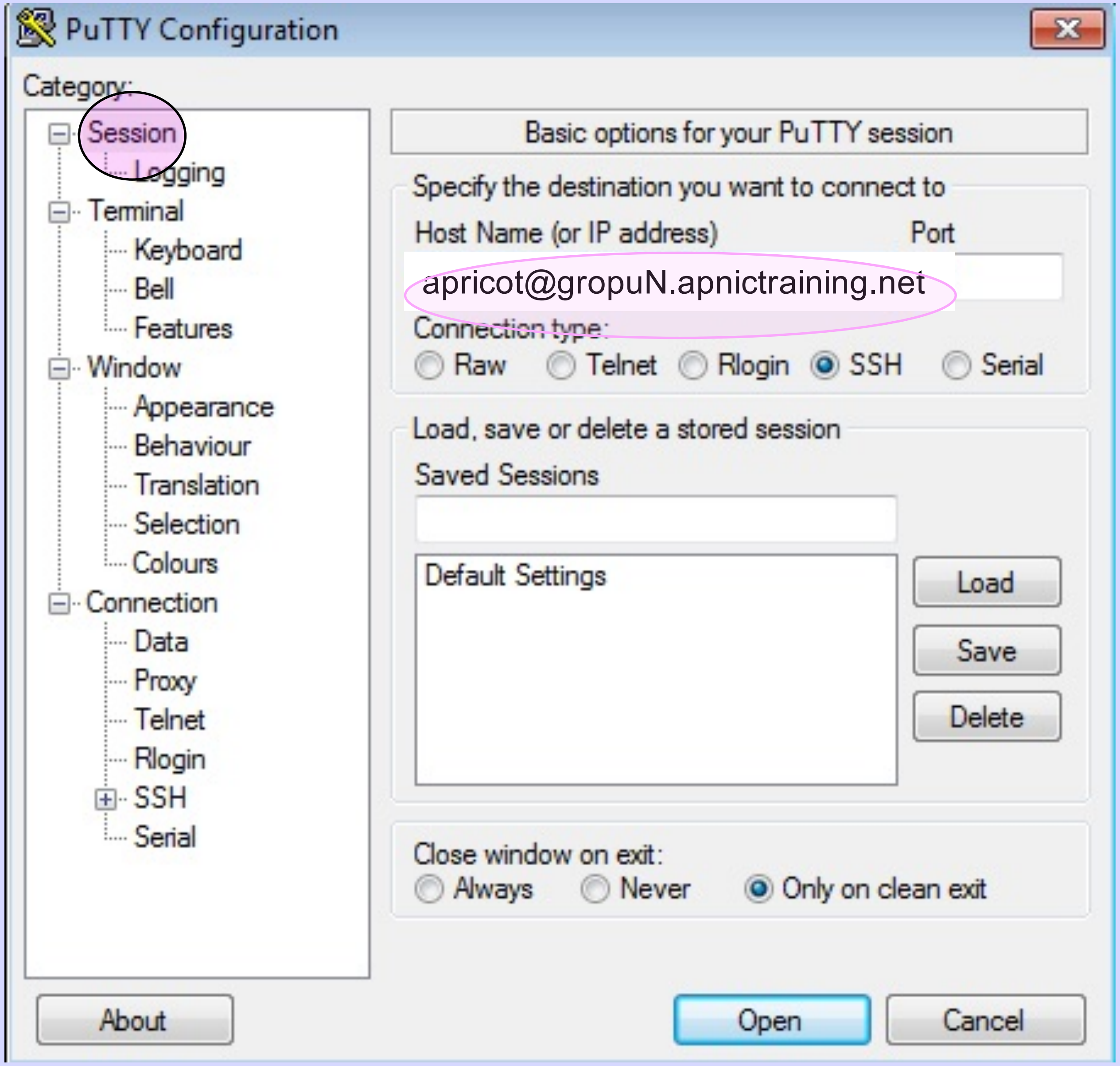
Authentication parameters

- [ ] Allow agent forwarding
- [ ] Allow attempted changes of username in SSH-2

Private key file for authentication:

C:\Users\randy\Desktop\putty\private-k

ssh to Host



PuTTY Configuration

Category:

- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - Telnet
  - Rlogin
  - SSH
  - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): apricot@gropuN.apnictraining.net

Port:

Connection type: Raw / Telnet / Rlogin / [✓ SSH] / Serial

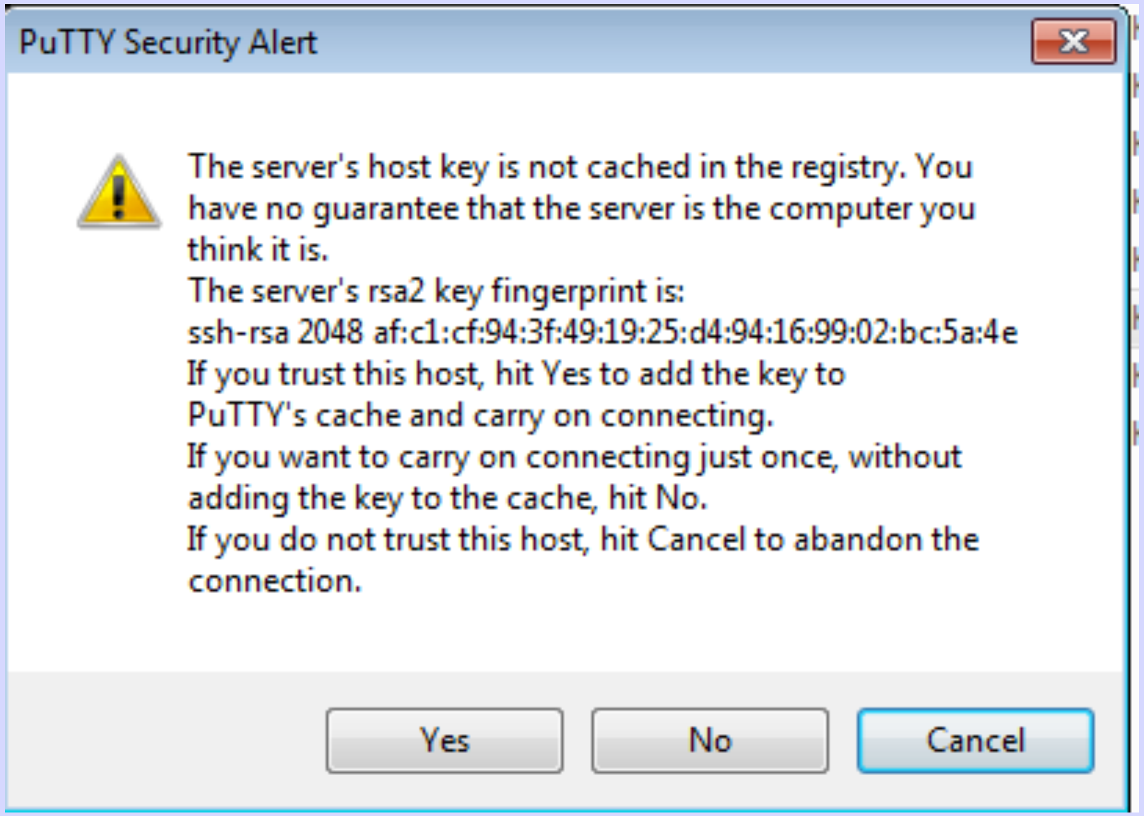
Load, save or delete a stored session

Saved Sessions:

Default Settings

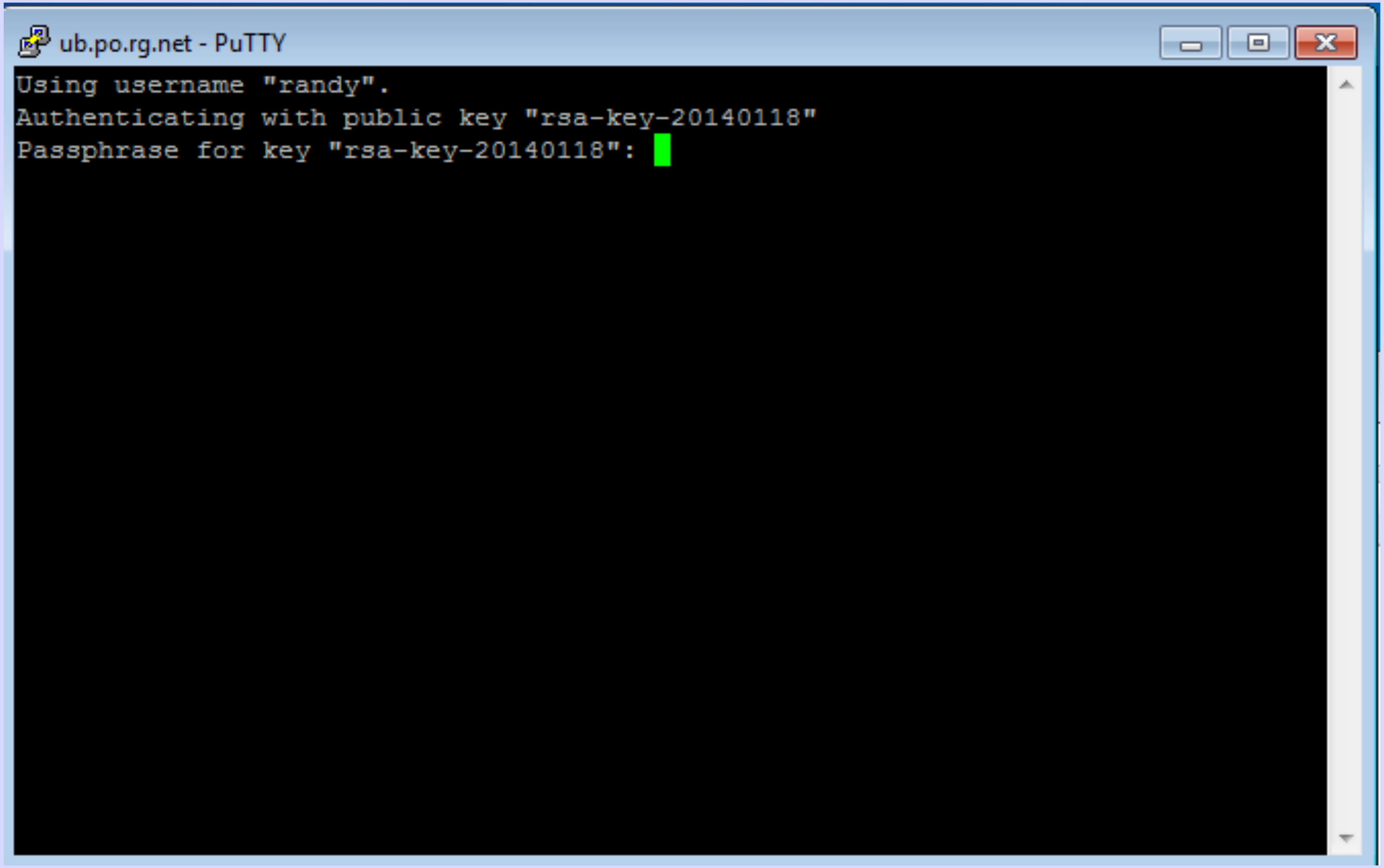
Close window on exit: Always / Never / [✓ Only on clean exit]

Accept Host's Key


PuTTY Security Alert

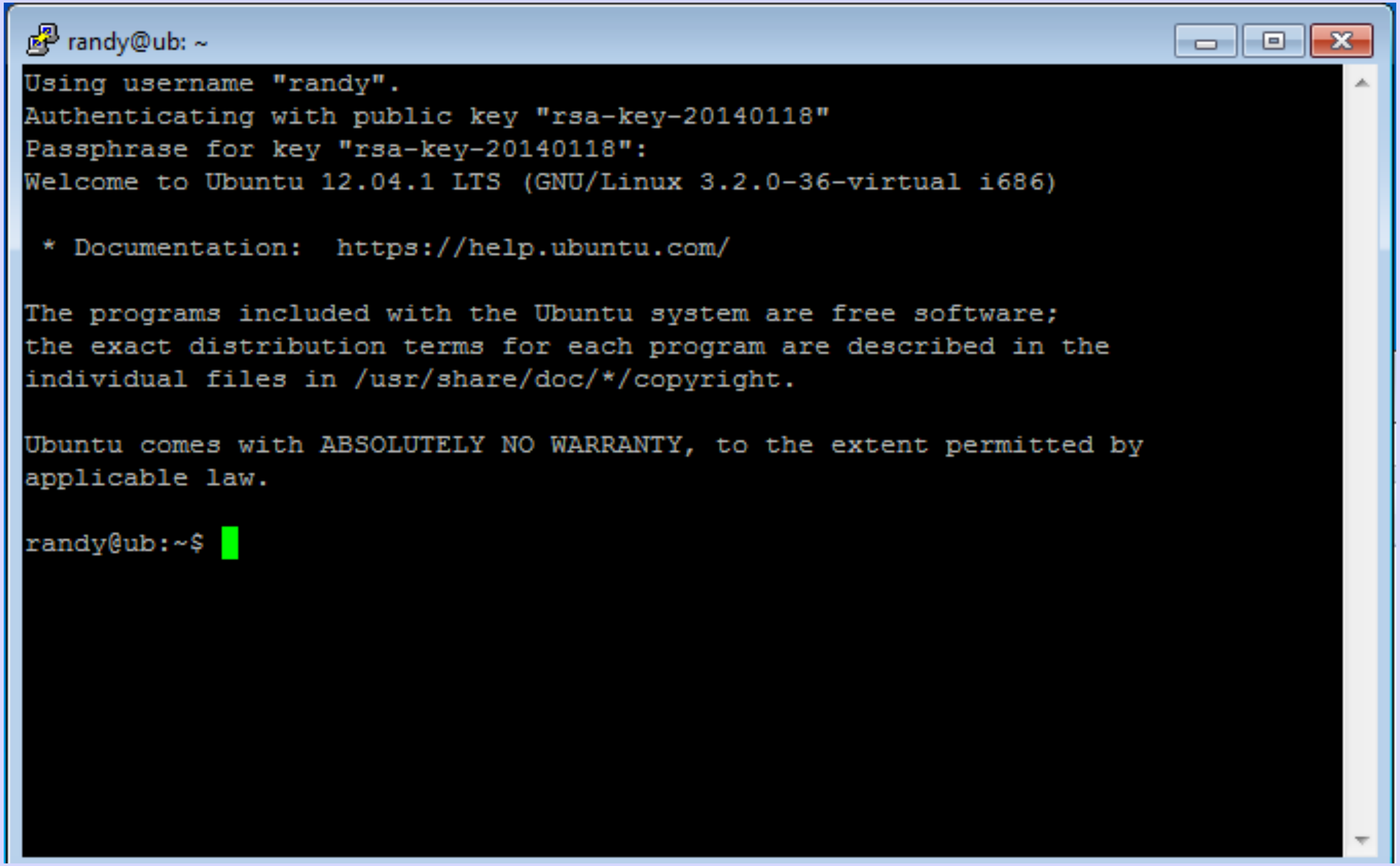
The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 af:c1:cf:94:3f:49:19:25:d4:94:16:99:02:bc:5a:4e
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

Passphrase for Key



```
Using username "randy".
Authenticating with public key "rsa-key-20140118"
Passphrase for key "rsa-key-20140118":
```

You Are In!



```
Using username "randy".
Authenticating with public key "rsa-key-20140118"
Passphrase for key "rsa-key-20140118":
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-36-virtual i686)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

randy@ub:~$ 
```

Disable Password based authentication

```
$ sudo vi /etc/ssh/sshd_config
```

In this file, set the following settings to the following values. If these settings are already in the file, set them to "no" rather than add new lines.

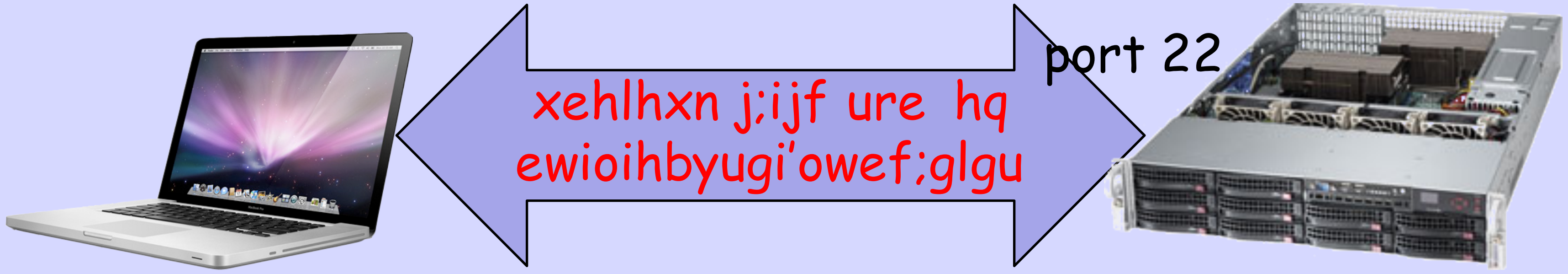
```
ChallengeResponseAuthentication no
PasswordAuthentication no
```

Once this is done, restart the SSH daemon to apply the settings.

```
$ sudo /etc/init.d/sshd restart
```

ssh - Shell Session

```
$ ssh class@ssh.derp.nz
```


xehlhxn j;ijf ure hq ewioihbyugi'owef;glgu
port 22