

2-1-1 ssh Secure SHell

Using Public Key Cryptography

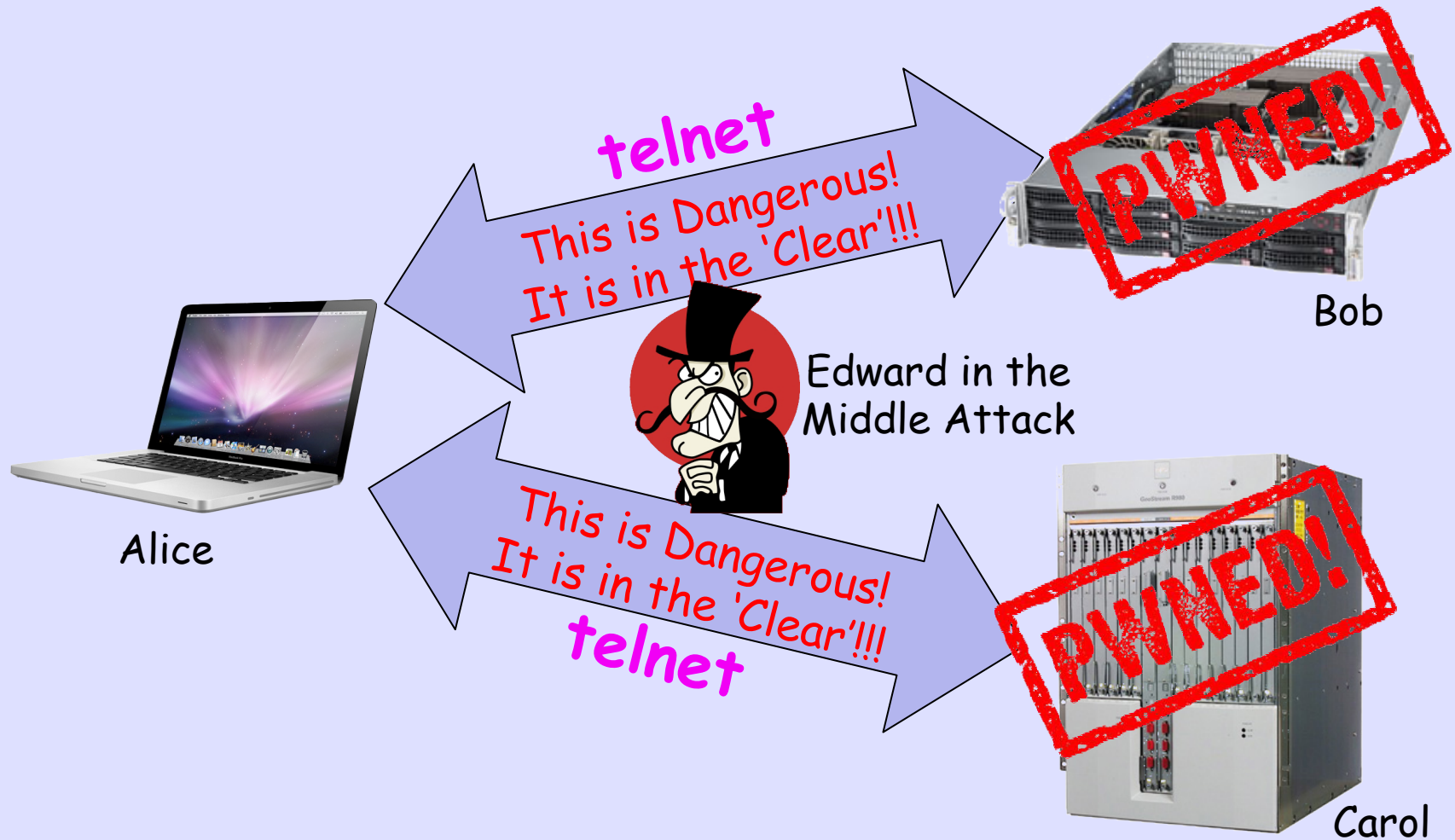
Keying, Key Exchange,
and Session Setup

Communicate Safely with Remote Systems

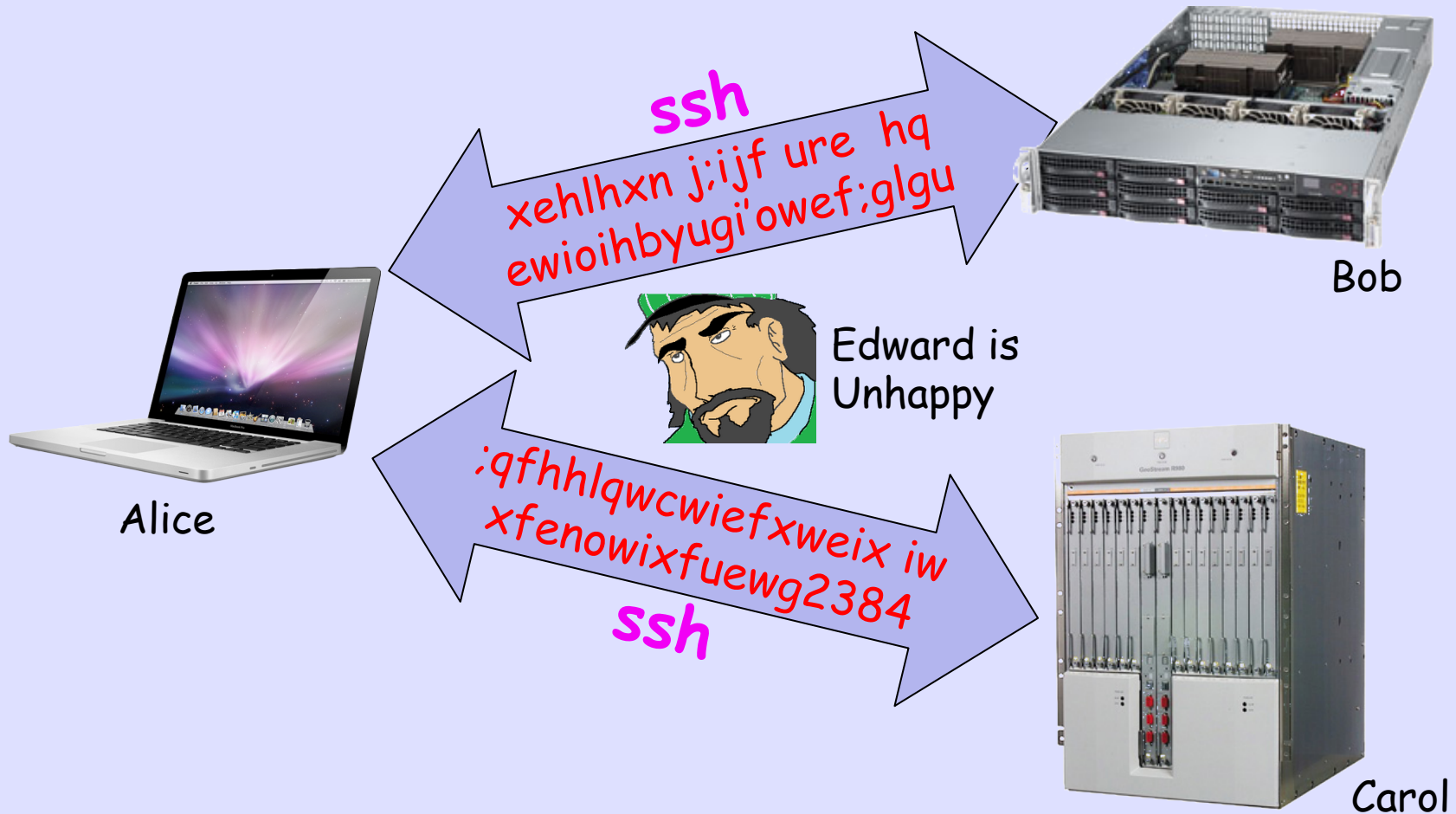
What is "Safely"

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

Traditional



Encrypted



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

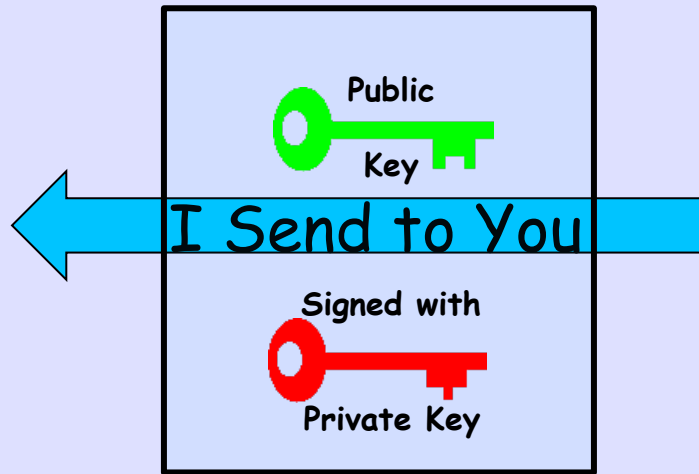
Think of SSH as
a Bit Like
PGP where the Other
End is a Computer,
Not a Human

But PGP is
Object Security
SSH is
Channel/Transport
Security

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 My Public Key

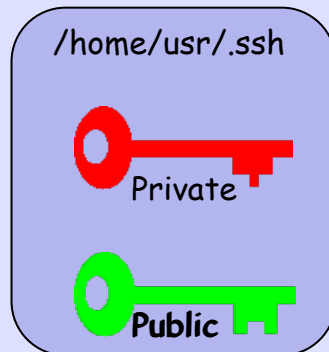
**If It Verifies, Then You Know That
I Must Have The Private Key**

**And You Know You Have My
Corresponding Public Key**

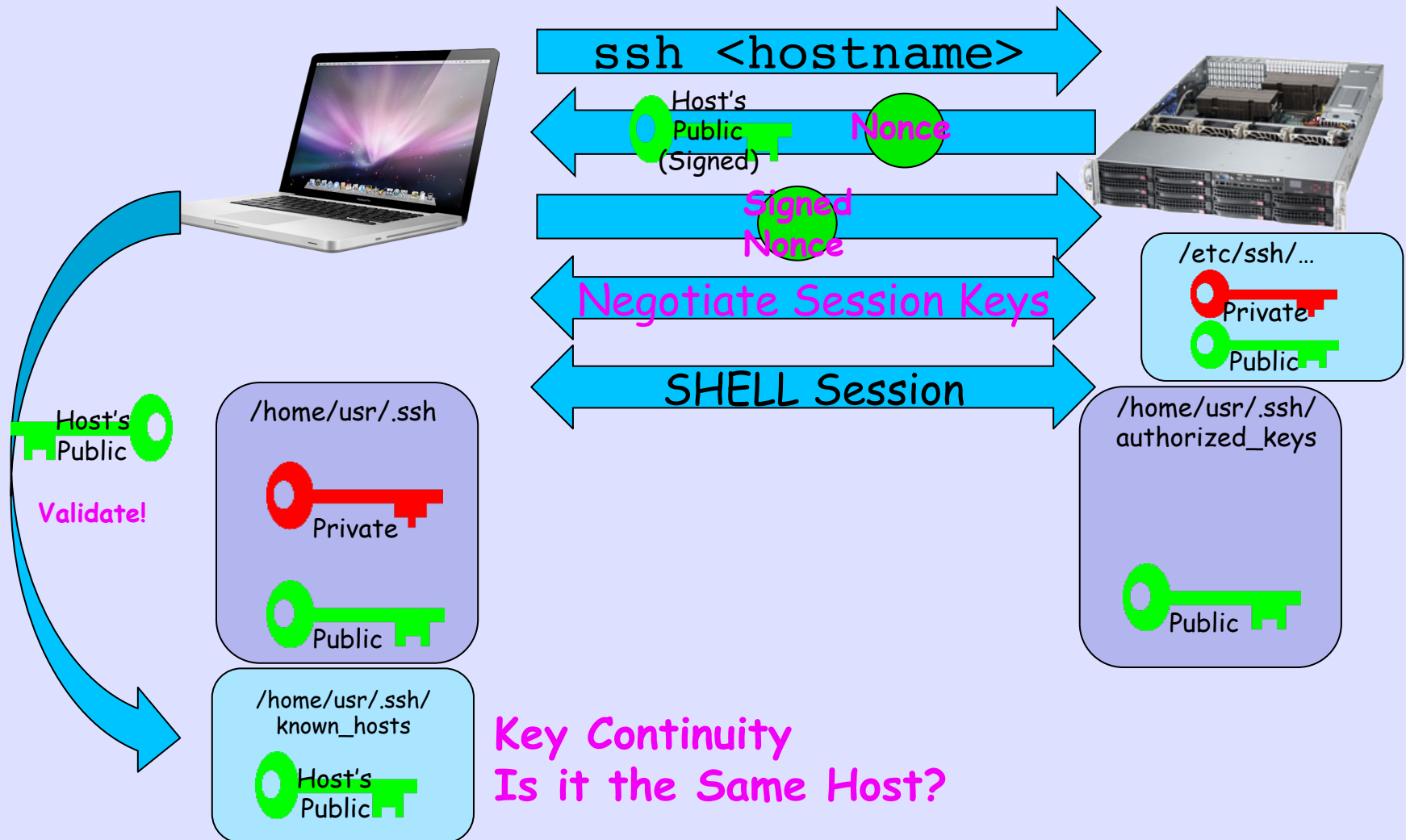
ssh - Keying Setup



`ssh-keygen -t rsa`



2-Way Authentication



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 +          |
+-----+

```

Elliptical Curve Key

```
/usr/home/foo> ssh-keygen -t ecdsa
```

```
Generating public/private ecdsa key pair.
```

```
Enter file in which to save the key (/usr/home/foo/.ssh/id_ecdsa):
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /usr/home/foo/.ssh/id_ecdsa.
```

```
Your public key has been saved in /usr/home/foo/.ssh/id_ecdsa.pub.
```

```
The key fingerprint is:
```

```
7a:9d:c5:05:5e:39:95:ae:f7:87:0a:43:66:67:2d:45 foo@psg.com
```

```
The key's randomart image is:
```

```
+--[ECDSA 256]---+
|
|      . Eoo|
|      . +o.|
|      . +.|
|      . + .|
|      S + * o|
|      . = = o .|
|      . . = . . .|
|      . o . o|
|      . . .|
+-----+

```

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.psq.com

The key's randomart image is:

```
+--[RSA1 2048]-----+
```

Diagram illustrating a 2D hexagonal lattice structure. The central hexagon is highlighted in light blue and contains the following labels:

- Top: $O + O B \dots$
- Right: $\dots = + \dots$
- Bottom-right: $\dots O O$
- Bottom: $B \cdot O$
- Left: $O S O \dots =$
- Bottom-left: $+ O + E$
- Bottom: $O \cdot O$

The lattice is enclosed in a dashed rectangular frame with a '+' sign at the bottom-left corner.

ssh v1 is for
2511s and other
antiques

Use Keys Not Passwords

- In `/etc/ssh/sshd_config`
`PermitRootLogin` without-password
`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

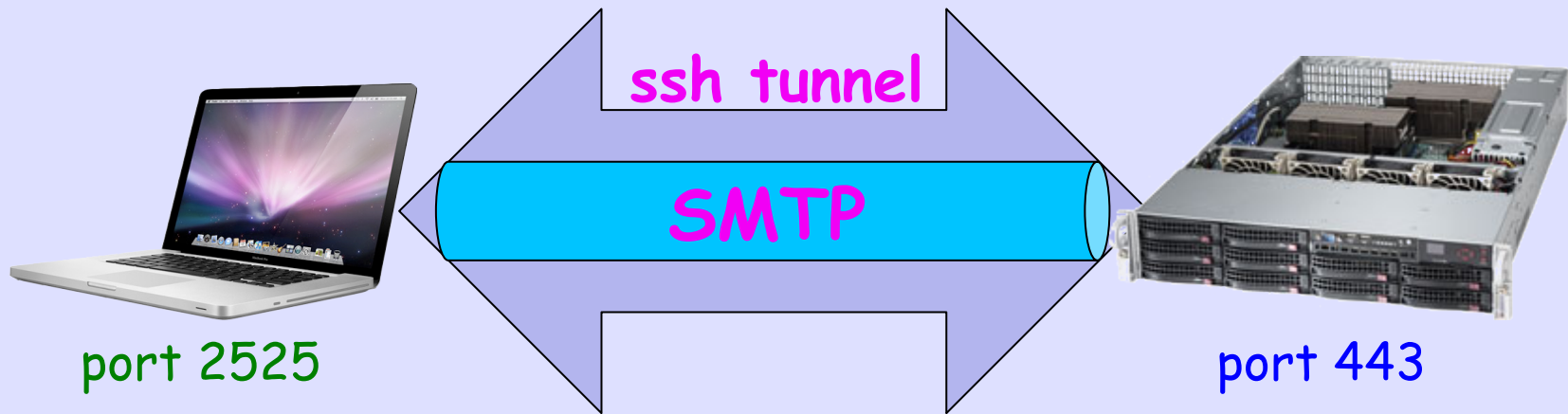
The Only Compromise
I Have Had to My
Infrastructure was a
Researcher who Stored
Their Private Key on a
Shared University Host

Private Key Protection

- FreeBSD Repository Compromise Two 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



```
$ 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

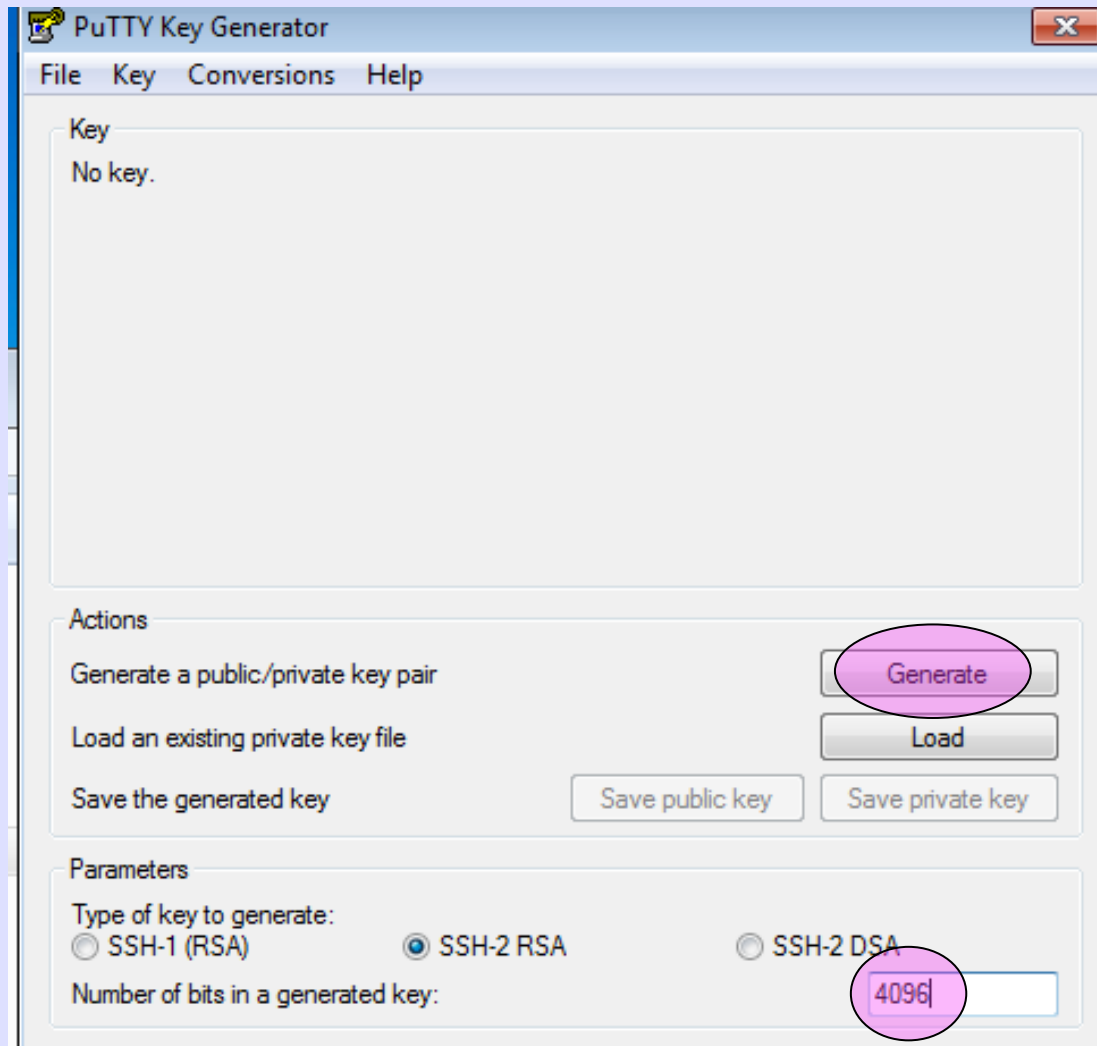
[http://www.chiark.greenend.org.uk/
~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

PuTTY: `putty.exe`

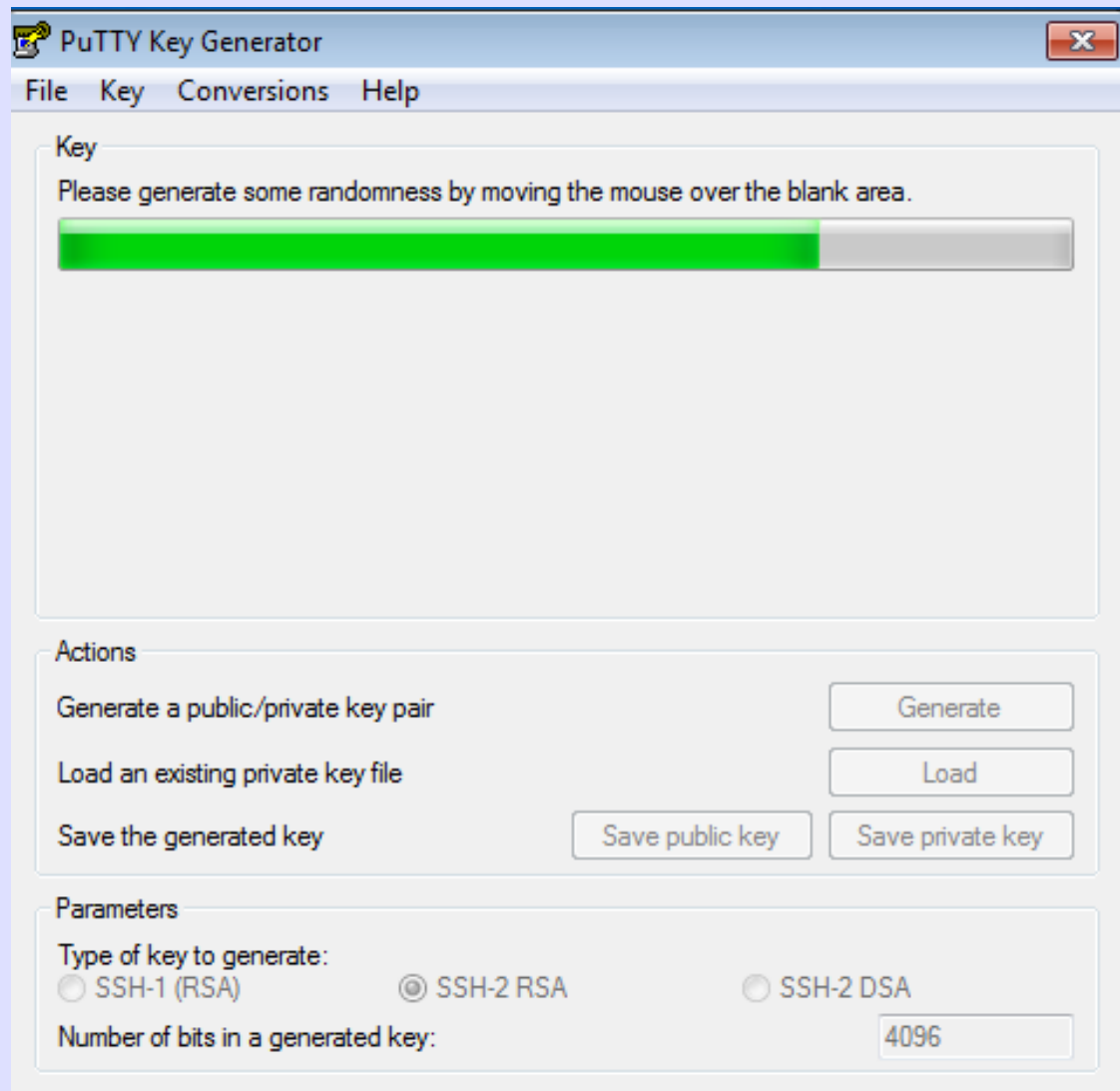
Pageant: `pageant.exe`

PuTTYgen: `puttygen.exe`

PuttyGen



Generate Key



Enter Passphrase & Save Key

PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAAnFDinOYLGUOn5sQxkoUldPhgsWwWRRLSN
U4QH187O8M4Ry864RnUBJAoknClwE
+0g2uPgQBn5s0796RdvzDS2mbAfukIXTMG46uileV
+5y9UPMLc5j8AGavVqu2uMksdoRFdTZTTzZ
```

Key fingerprint: ssh-rsa 4096 f4:c1:60:77:86:02:32:1d:41:83:8d:c1:ca:47:9c:26

Key comment: rsa-key-20140118

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: 4096

Putting the Key on the Target Host

- Mail the **Public** key to your sysadmin: (randy@psg.com) and he will install it
- He will then create the .ssh directory in your home directory
- And put the **public** key in a file called authorized_keys

```
mkdir ~username/.ssh
```

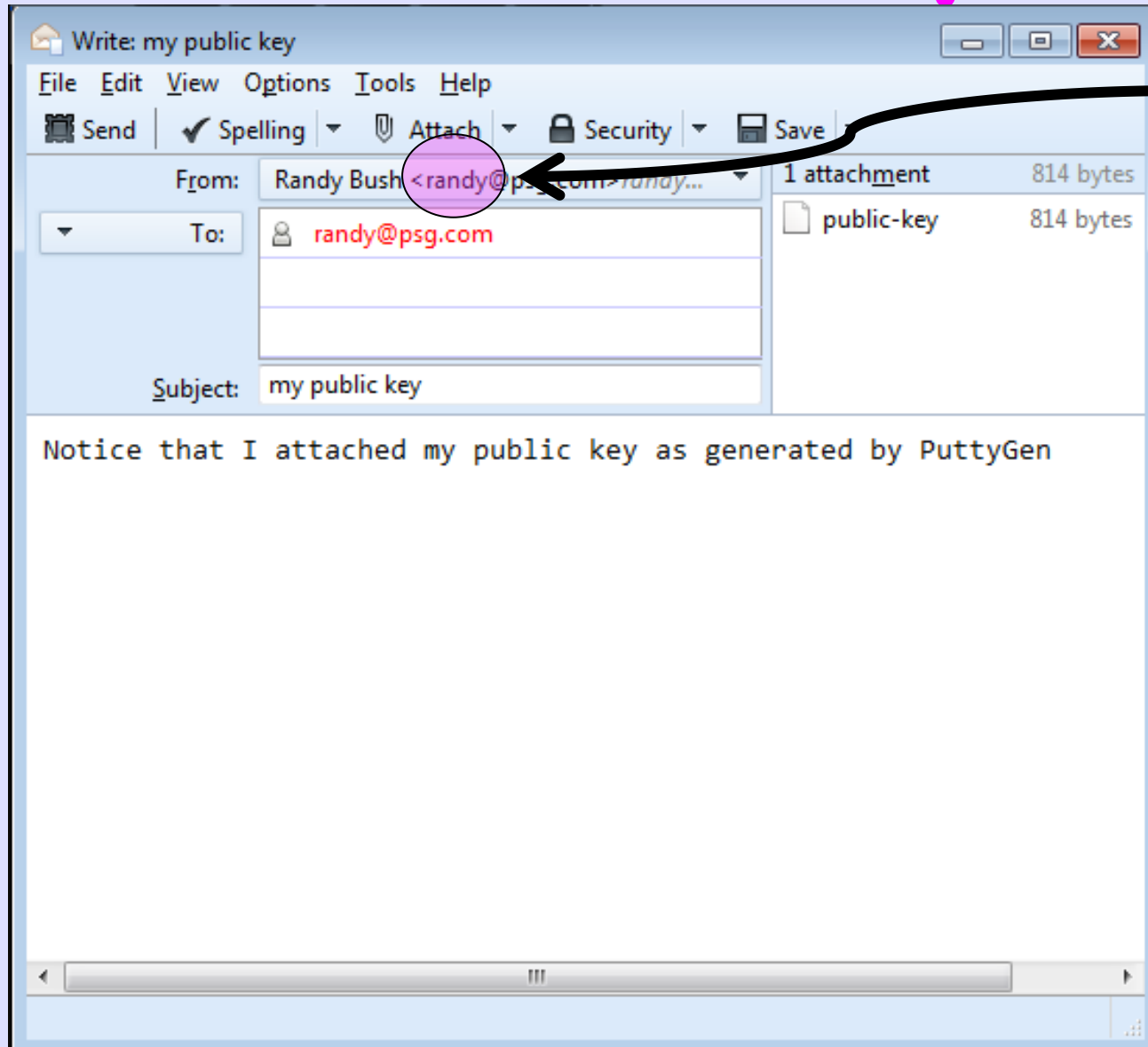
```
cat id_rsa.pub >> ~username/.ssh/authorized_keys
```

- Permissions have to be non world readable

```
chown -R username:guest ~username/.ssh
```

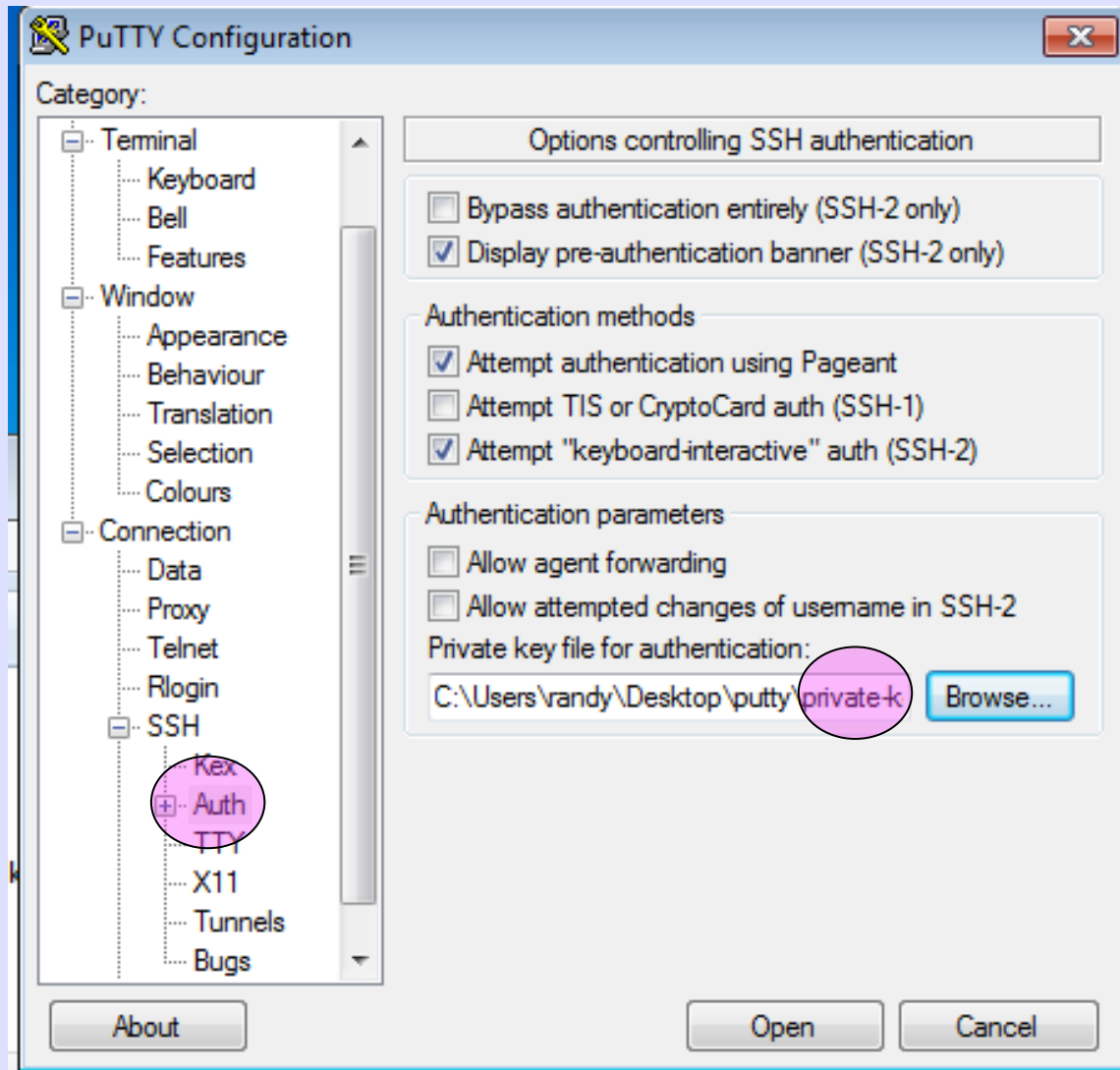
```
chmod -R go-rwx ~username/.ssh
```

Mail the Key

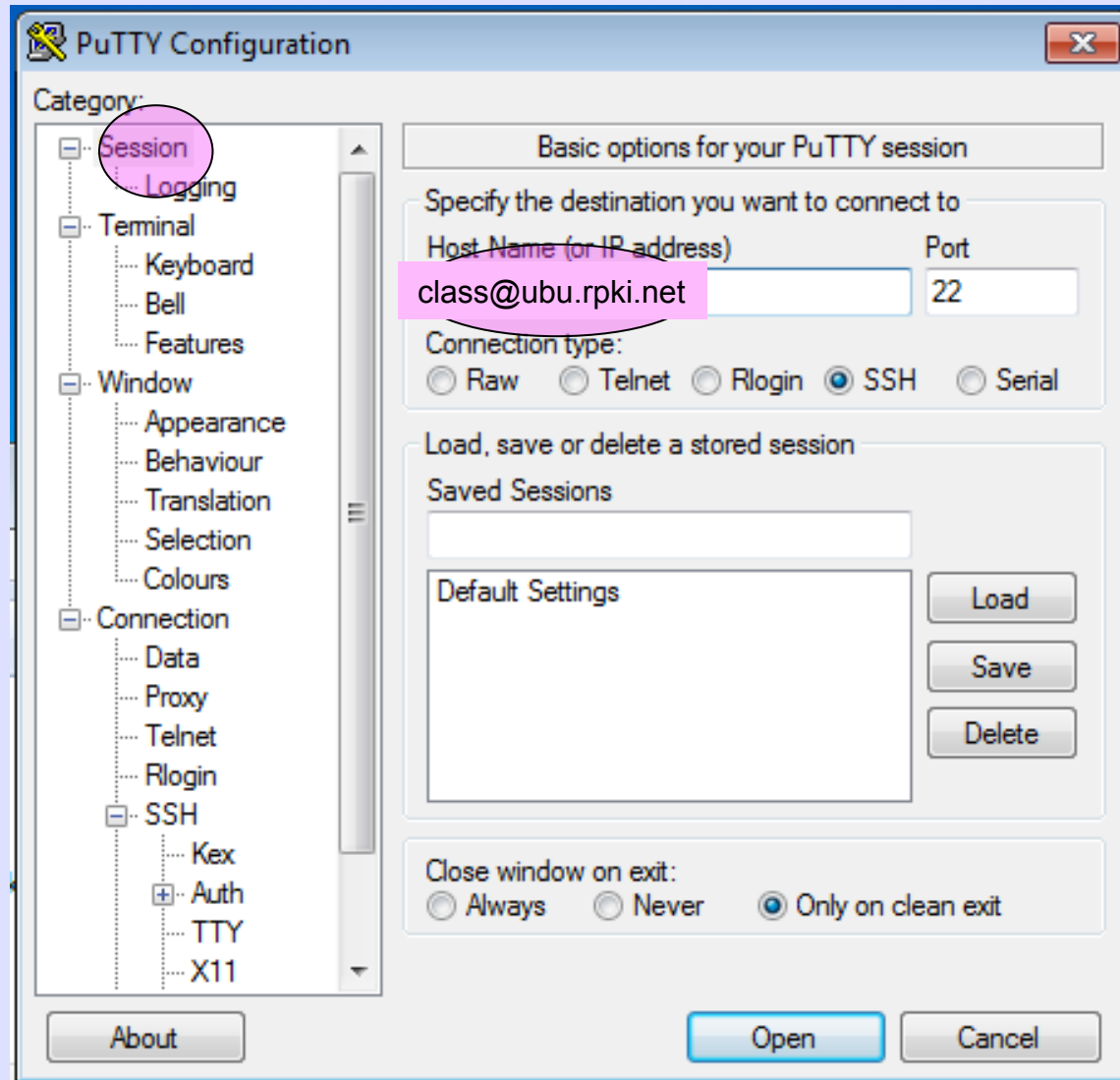


Your
User
Name

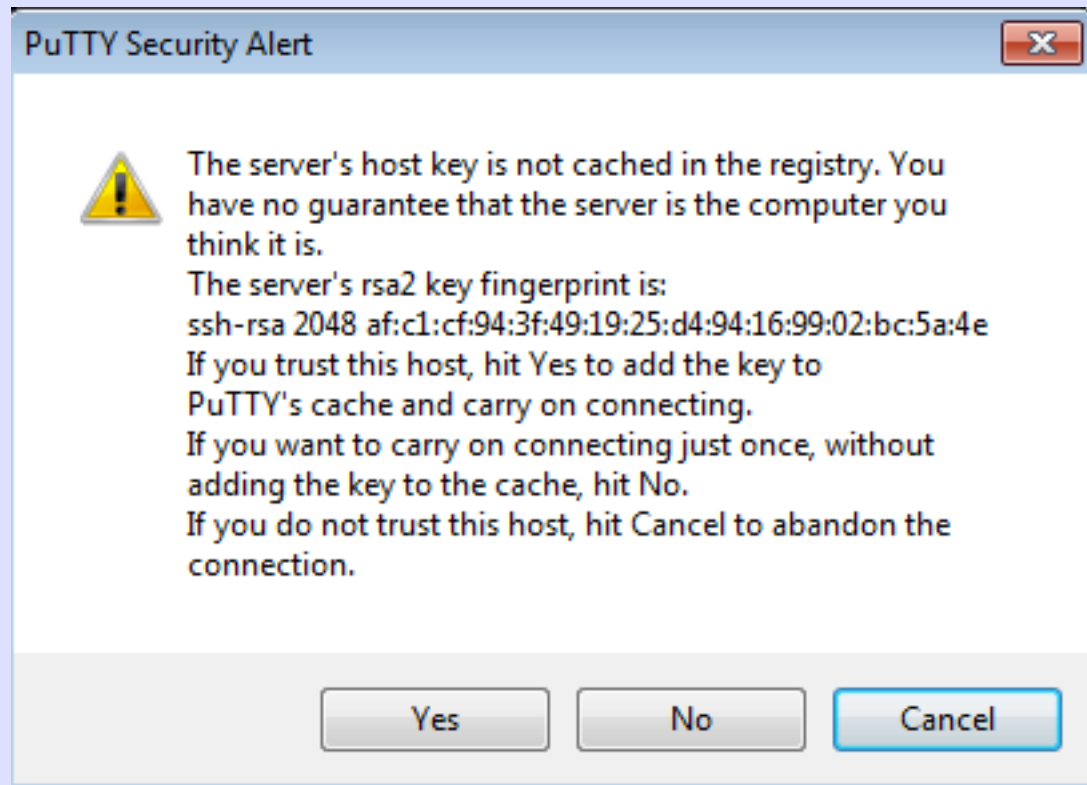
Load Key in Putty



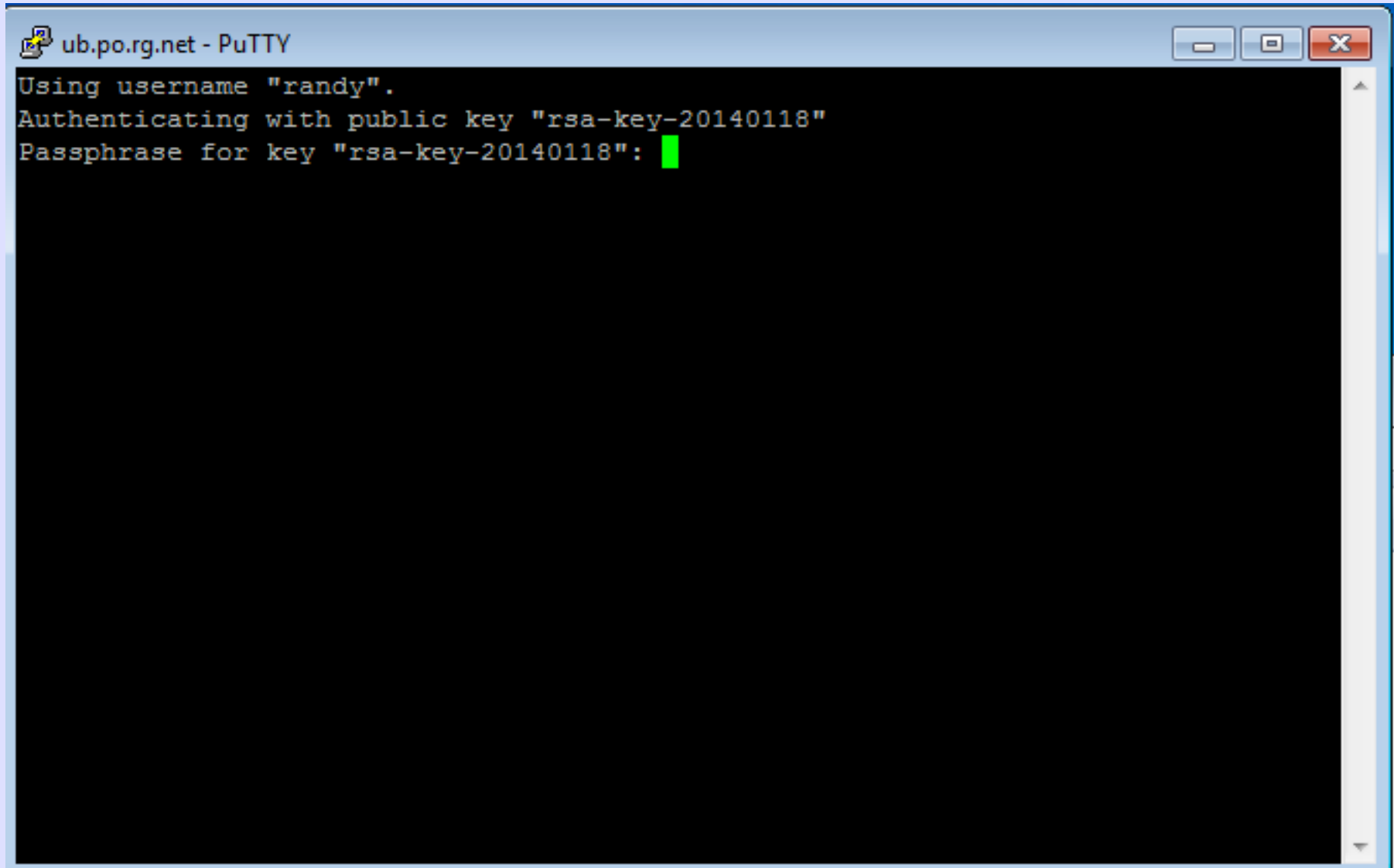
ssh to Host



Accept Host's Key



Passphrase for Key

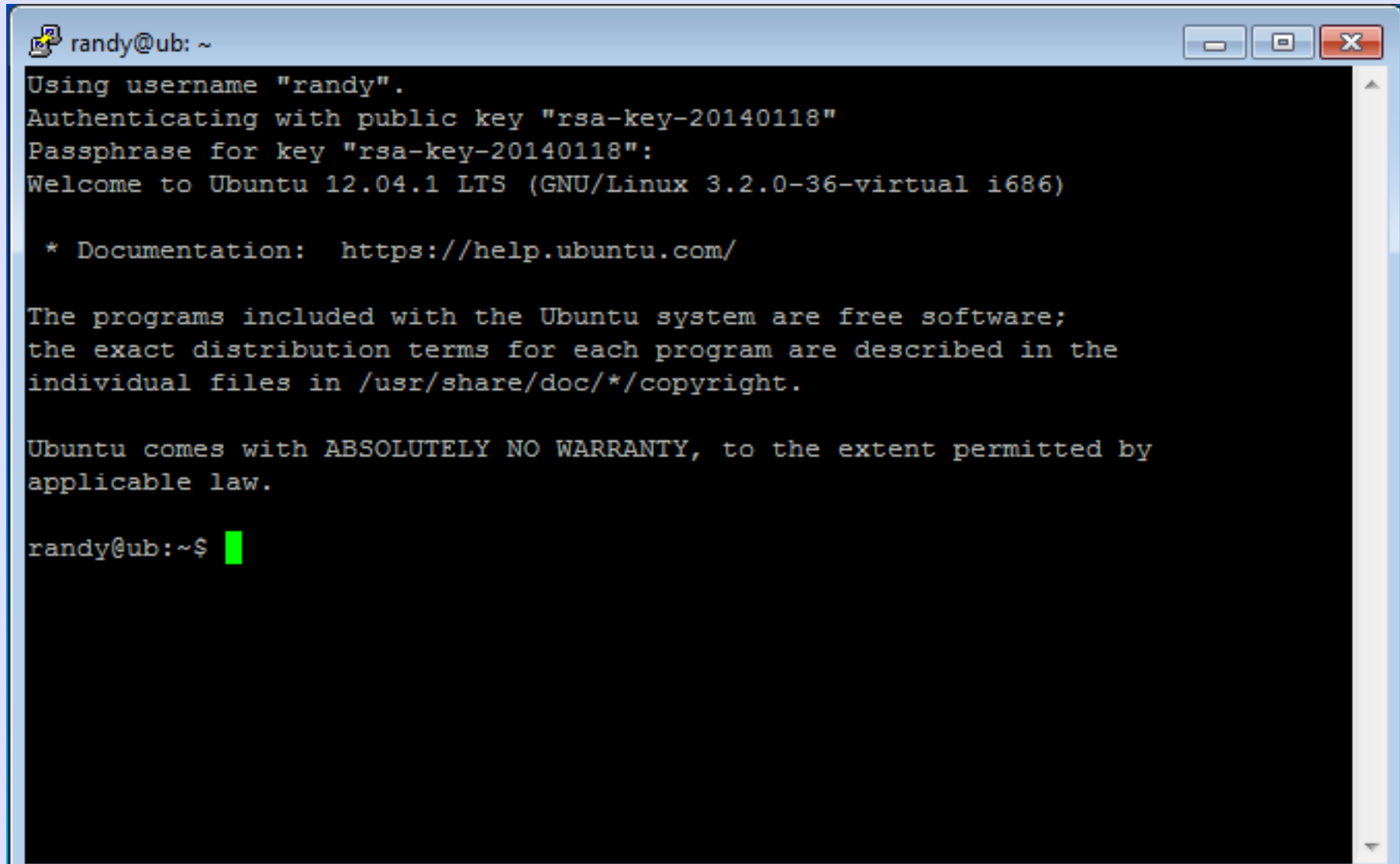


A screenshot of a PuTTY terminal window titled "ub.po.rg.net - PuTTY". The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner. The terminal output shows the following text:

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

The text is displayed in a monospaced font on a black background. A green cursor is visible at the end of the third line, indicating where the user should enter the passphrase.

You Are In!

A terminal window titled 'randy@ub: ~' with standard window controls. The terminal text shows a successful login for user 'randy' using a public key. It displays the Ubuntu version (12.04.1 LTS) and provides documentation and warranty information. The prompt 'randy@ub:~\$' is followed by a green cursor.

```
randy@ub: ~  
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:~$
```


ssh - Shell Session

```
$ ssh username@ubu.rpki.net
```

