



The Unix Shell

The Secure Shell



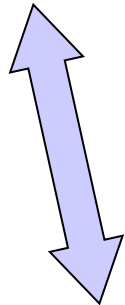
Copyright © Software Carpentry 2011

This work is licensed under the Creative Commons Attribution License

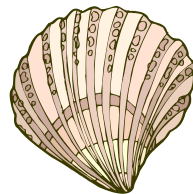
See <http://software-carpentry.org/license.html> for more information.



```
$ pwd
```

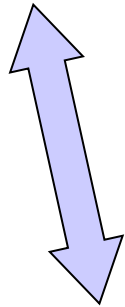


shell

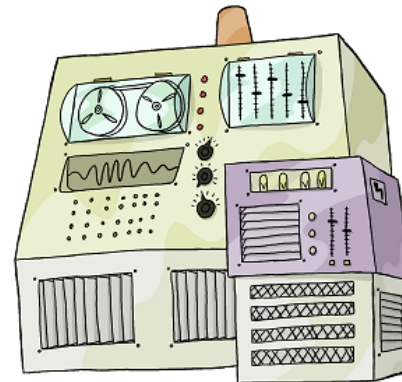
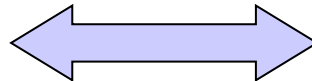
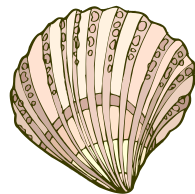




```
$ pwd  
/users/vlad  
$
```

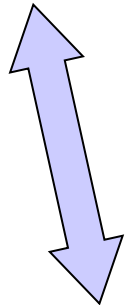


shell

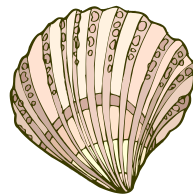




```
login as: vlad  
Password: *****
```

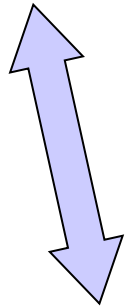


shell

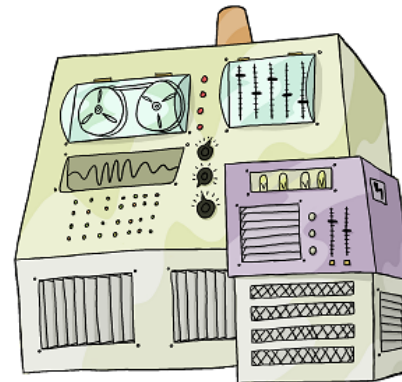
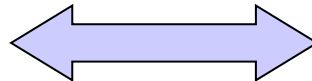
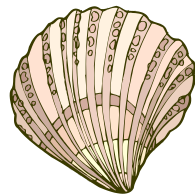




```
login as: vlad
Password: ****
$
```

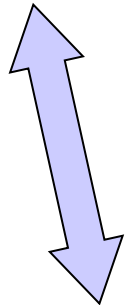


shell

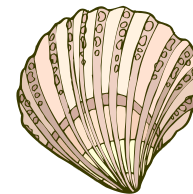
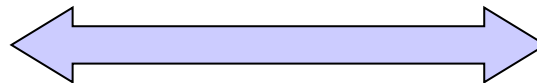
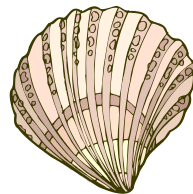




```
login as: vlad  
Password: *****  
moon>
```



shell



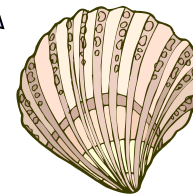
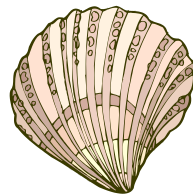
remote shell



```
login as: vlad  
Password: *****  
moon>
```



shell



remote shell

```
$ pwd
```

```
/users/vlad
```

```
$ ssh vlad@moon
```

Password:

\$ pwd

/users/vlad

\$ ssh vlad@moon

Password: ***

Access denied

Password:

```
$ pwd
```

```
/users/vlad
```

```
$ ssh vlad@moon
```

```
Password: ***
```

```
Access denied
```

```
Password: ****
```

```
moon> pwd
```

```
/home/vlad
```

```
moon> ls -F
```

```
bin/          cheese.txt    dark_side/    rocks.cfg
```

```
$ pwd
```

```
/users/vlad
```

```
$ ssh vlad@moon
```

```
Password: ***
```

```
Access denied
```

```
Password: ****
```

```
moon> pwd
```

```
/home/vlad
```

```
moon> ls -F
```

```
bin/          cheese.txt    dark_side/    rocks.cfg
```

```
moon> exit
```

```
$ pwd
```

```
/users/vlad
```

```
$ ssh vlad@moon
```

```
Password: ****
```

```
moon> pwd
```

```
/home/vlad
```

```
moon> ls -F
```

```
bin/ cheese.txt dark_side/ rocks.cfg
```

```
moon> exit
```

```
$ pwd
```

```
/users/vlad
```

```
$ ls -F
```

```
bin/ data/ mail/ music/
```

```
notes.txt papers/ pizza.cfg solar/
```

```
solar.pdf swc/
```

```
$ scp vlad@moon:/home/vlad/cheese.txt  
vlad@earth:/users/vlad
```

source file...

```
$ scp vlad@moon:/home/vlad/cheese.txt  
    vlad@earth:/users/vlad
```

source file...

...to destination directory

```
$ scp vlad@moon:/home/vlad/cheese.txt  
      vlad@earth:/users/vlad
```

source file...

...to destination directory

source and destination are written as
user@computer:path

```
$ scp vlad@moon:/home/vlad/cheese.txt  
      vlad@earth:/users/vlad
```

```
Password:  *****
```


```
cheese.txt                100%   9   1.0 KB/s   00:00
```



```
$ scp vlad@moon:/home/vlad/cheese.txt  
    vlad@earth:/users/vlad  
$ scp -r vlad@moon:/home/vlad/dark_side  
    vlad@earth:/users/vlad
```

-r indicates a directory and its contents

```
$ scp -r vlad@moon:/home/vlad/dark_side  
    vlad@earth:/users/vlad  
$ scp -r vlad@moon:/home/vlad/dark_side  
    /users/vlad  
$ pwd  
/users/vlad  
$ scp -r vlad@moon:/home/vlad/dark_side
```



.

same destination path

```
$ ssh vlad@moon
```

```
Password: ****
```

```
moon> df -h
```

<i>Filesystem</i>	<i>Size</i>	<i>Used</i>	<i>Avail</i>	<i>Use%</i>	<i>Mounted On</i>
<i>/dev/sda1</i>	<i>7.9G</i>	<i>2.1G</i>	<i>5.5G</i>	<i>28%</i>	<i>/</i>
<i>/dev/sda2</i>	<i>791G</i>	<i>150G</i>	<i>642G</i>	<i>19%</i>	<i>/home</i>

```
moon> df -h > usage.txt
```

```
moon> exit
```

```
$ scp vlad@moon:/home/vlad/usage.txt .
```

```
Password: ****
```

```
usage.txt          100%  134  1.0 KB/s  00:00
```

```
$ ssh vlad@moon 'df -h'
```

```
Password: ****
```

<i>Filesystem</i>	<i>Size</i>	<i>Used</i>	<i>Avail</i>	<i>Use%</i>	<i>Mounted On</i>
<i>/dev/sda1</i>	<i>7.9G</i>	<i>2.1G</i>	<i>5.5G</i>	<i>28%</i>	<i>/</i>
<i>/dev/sda2</i>	<i>791G</i>	<i>150G</i>	<i>642G</i>	<i>19%</i>	<i>/home</i>

```
$ ssh vlad@moon 'df -h'
```

```
Password: ****
```

<i>Filesystem</i>	<i>Size</i>	<i>Used</i>	<i>Avail</i>	<i>Use%</i>	<i>Mounted On</i>
<i>/dev/sda1</i>	<i>7.9G</i>	<i>2.1G</i>	<i>5.5G</i>	<i>28%</i>	<i>/</i>
<i>/dev/sda2</i>	<i>791G</i>	<i>150G</i>	<i>642G</i>	<i>19%</i>	<i>/home</i>

```
$ ssh vlad@moon 'df -h' >> usage.log
```

```
Password: ****
```

```
$ ls -F
```

<i>bin/</i>	<i>data/</i>	<i>mail/</i>	<i>music/</i>
<i>notes.txt</i>	<i>papers/</i>	<i>pizza.cfg</i>	<i>solar/</i>
<i>solar.pdf</i>	<i>swc/</i>	<i>usage.log</i>	<i>usage.txt</i>

same result

character stream

\$ echo "open sesame, please" | ssh
vlad@moon 'cat > magic.txt'

Password: ****

character stream

remote shell receives stream from pipe

```
$ echo "open sesame, please" | ssh  
  vlad@moon 'cat > magic.txt'
```

Password: *****

remote command receives input piped to ssh
cat repeats input stream as output

redirection within remote shell

The diagram illustrates the flow of a character stream. A local shell (Unix Shell) sends a character stream to an SSH connection. The SSH connection then sends the stream to a remote shell (Secure Shell). The remote shell receives the stream from the pipe and redirects it to a file (magic.txt) using the cat command. The local shell also sends a password to the remote shell.

before

```
$ ssh vlad@moon 'ls -F /home/vlad'
```

```
Password: *****
```

```
bin/          cheese.txt    dark_side/    rocks.cfg
```

```
$ echo "open sesame, please" | ssh  
    vlad@moon 'cat > magic.txt'
```

```
Password: *****
```

```
$ ssh vlad@moon 'ls -F /home/vlad'
```

```
Password: *****
```

```
bin/          cheese.txt    dark_side/    magic.txt  
rocks.cfg
```

after

before

```
$ ssh vlad@moon 'ls -F /home/vlad'
```

```
Password: *****
```

```
bin/          cheese.txt    dark_side/    rocks.cfg
```

```
$ echo "open sesame, please" | ssh  
    vlad@moon 'cat > magic.txt'
```

```
Password: *****
```

```
$ ssh vlad@moon 'ls -F /home/vlad'
```

```
Password: *****
```

```
bin/          cheese.txt    dark_side/    magic.txt
```

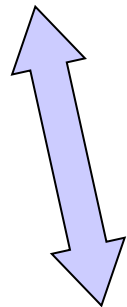
```
rocks.cfg
```

```
$ scp vlad@moon:/home/vlad/magic.txt .
```

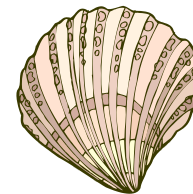
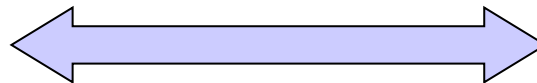
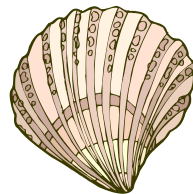
```
Password: *****
```

after

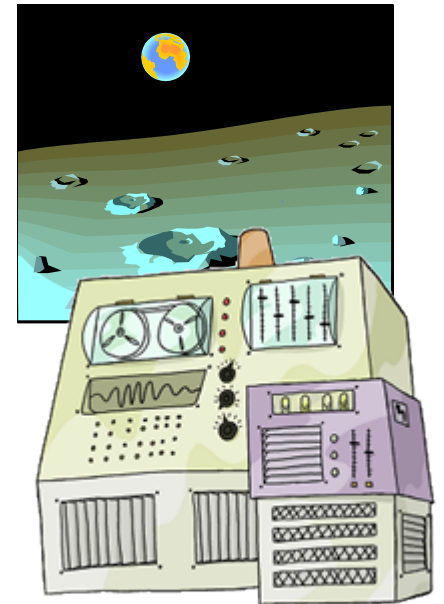
```
login as: vlad  
Password: ****
```



shell

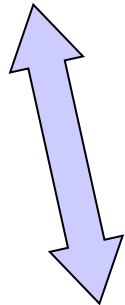


remote shell

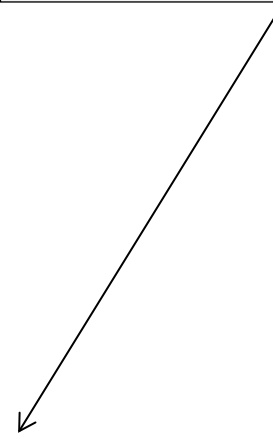
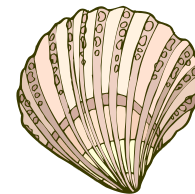
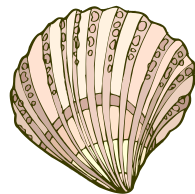




```
login as: vlad  
Password: thriller
```



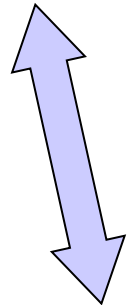
shell



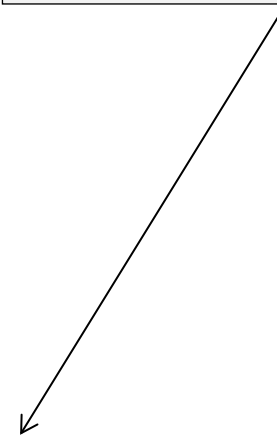
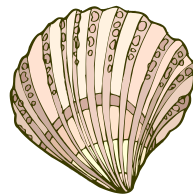
remote shell



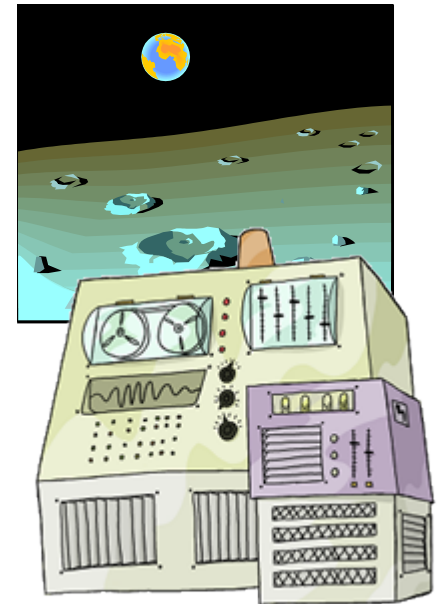
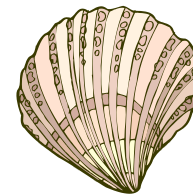
```
login as: vlad  
Password: thriller
```

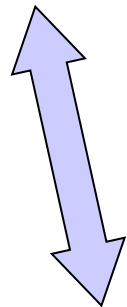


shell

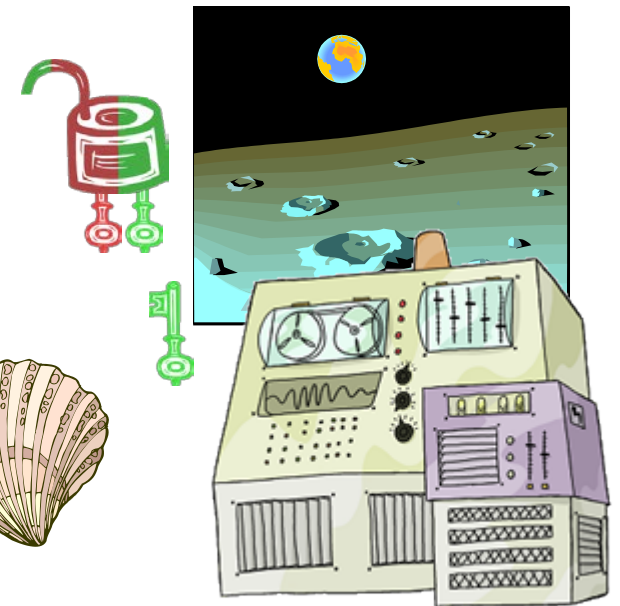
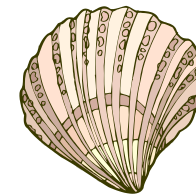
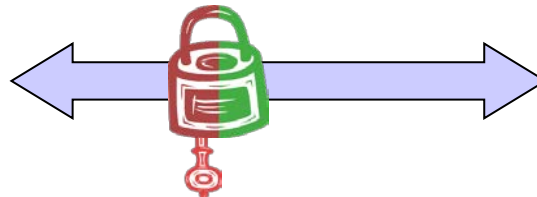
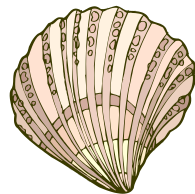


remote shell

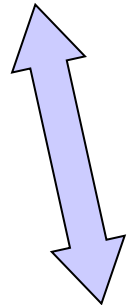




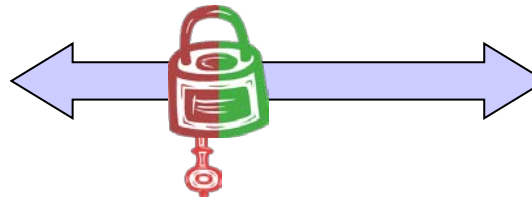
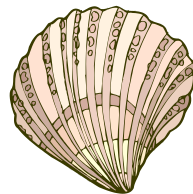
shell



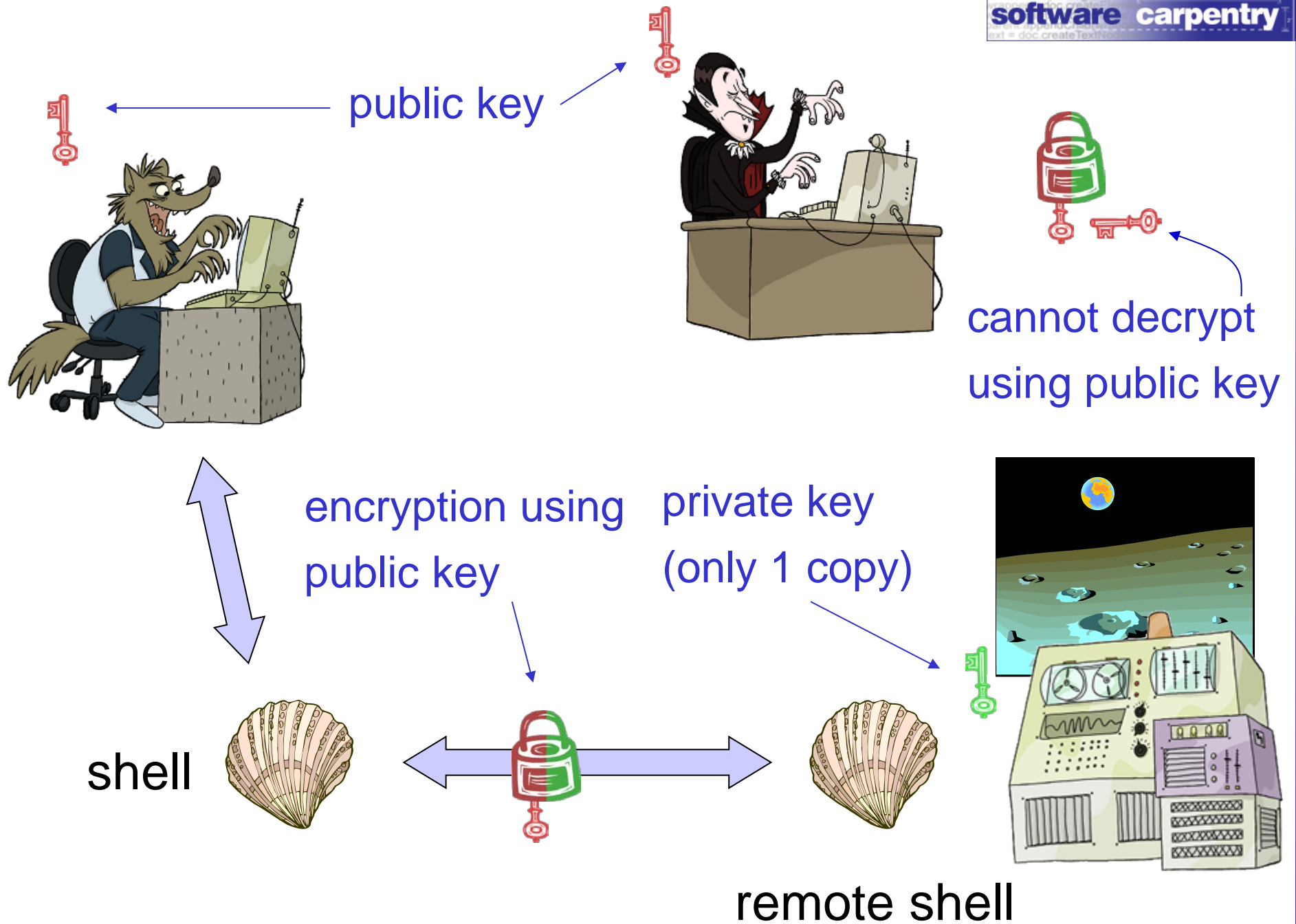
remote shell



shell



remote shell

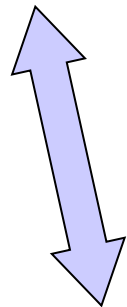


login as: `vlad`

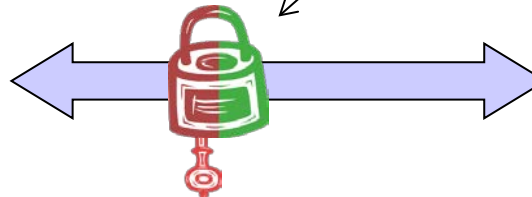
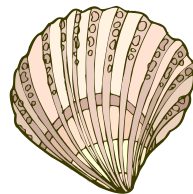
Password: `*****`

`huxyo ew: xdvw`

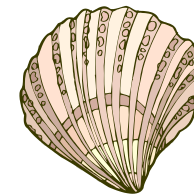
`uqfcmjbn: lhiujdbj`



shell



remote shell




```
$ ssh vlad@moon
```

```
The authenticity of host 'moon (10.1.2.3)'
can't be established.
```

```
RSA key fingerprint is
```

```
f1:68:f5:90:47:dc:a8:e9:62:df:c9:21:f0:8b:c5:39.
```

```
Are you sure you want to continue connecting
```

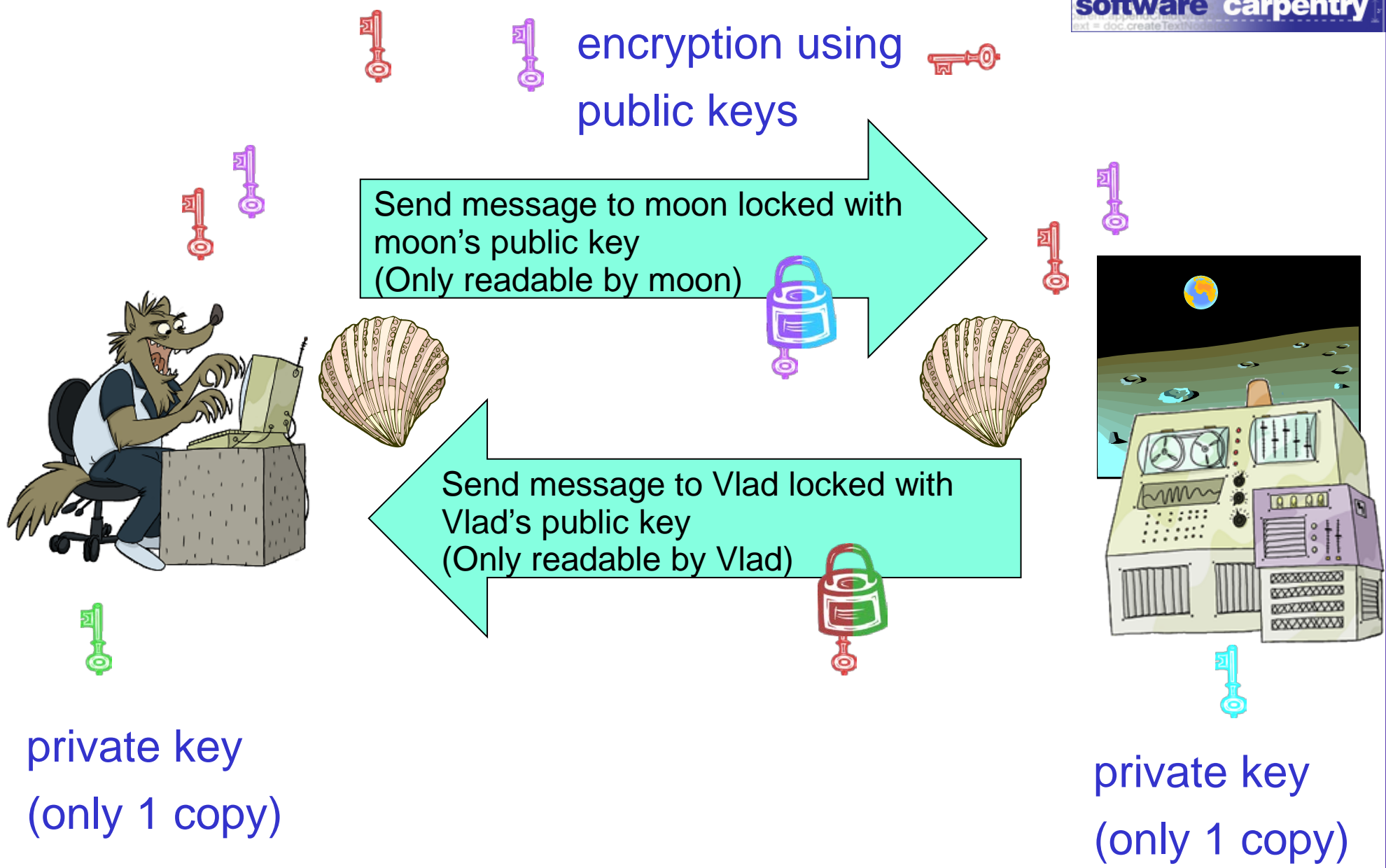
```
(yes/no)? yes
```

```
Warning: Permanently added 'moon,10.1.2.3' (RSA)
to the list of known hosts.
```

```
Password: *****
```

```
moon>
```

encryption using public keys





created by

Elango Cheran

February 2011



Copyright © Software Carpentry 2011

This work is licensed under the Creative Commons Attribution License

See <http://software-carpentry.org/license.html> for more information.

The ssh agent

- Stores secret keys in memory.
- Avoids repeated typing of the pass phrases.
- Can talk to a forwarding mechanism.
For example:
 - your workstation → jasmin-login1 → jasmin-sci1
 - jasmin-login1 does not have the private key
 - authentication traffic forwarded from end to end:
 - jasmin-sci1 sends challenge
 - workstation sends response, proving your identity

The ssh agent (continued)

- To start the agent and load your secret key:
 - Linux: session manager should start agent for you. Use `ssh-add` to load key (if not done automatically when ssh first used).
 - Windows: launch Pageant and click “add key”.
 - Enter your pass phrase.
- For authentication forwarding:
 - Linux: use “`ssh -A`” (often the default)
 - Windows: in PuTTY, go to Connection → SSH → Auth, and “Allow agent forwarding”

Exercise

1. Ssh to xuser1 @10.30.10.119.
2. Use ls, pwd and other commands to prove to yourself it's a different computer.
3. Logout (use exit or ^D)
4. Use scp to copy some files to the remote computer.
5. Login again and ls the files.
6. Logout
7. Use ssh to run df on the remote computer.