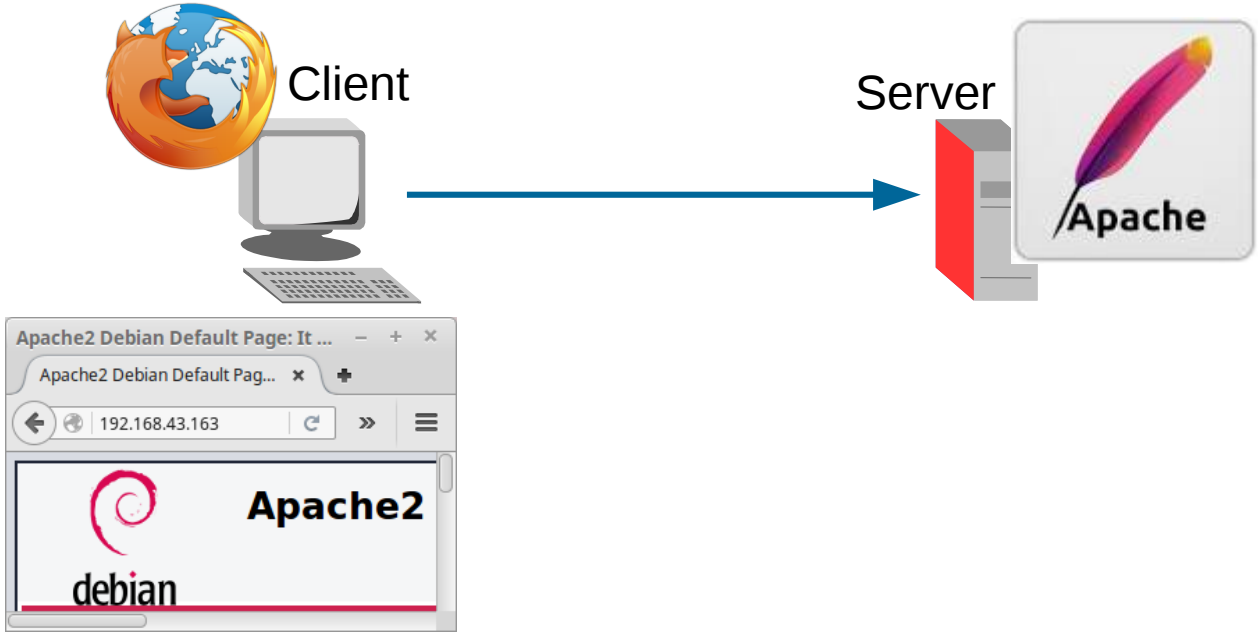
CMP3214
Computer
Communication Networks

Lecture 9

Applications



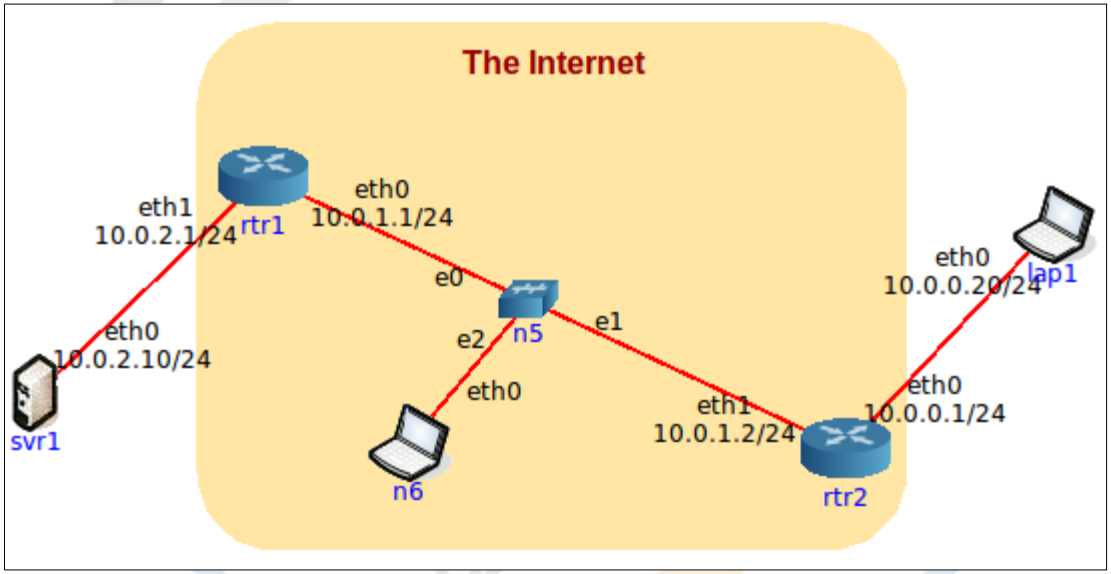
Diarmuid Ó Briain
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

Client-Server architecture

- A Client-Server architecture involves two parties:
  - Server which is a daemon running on a computer with an open Transport Layer port.
  - A Client requiring service connect to that port and depending on the application share information.

Client-Server example



CCN Course daemon (ccnd.py)

- Simple TCP Socket.

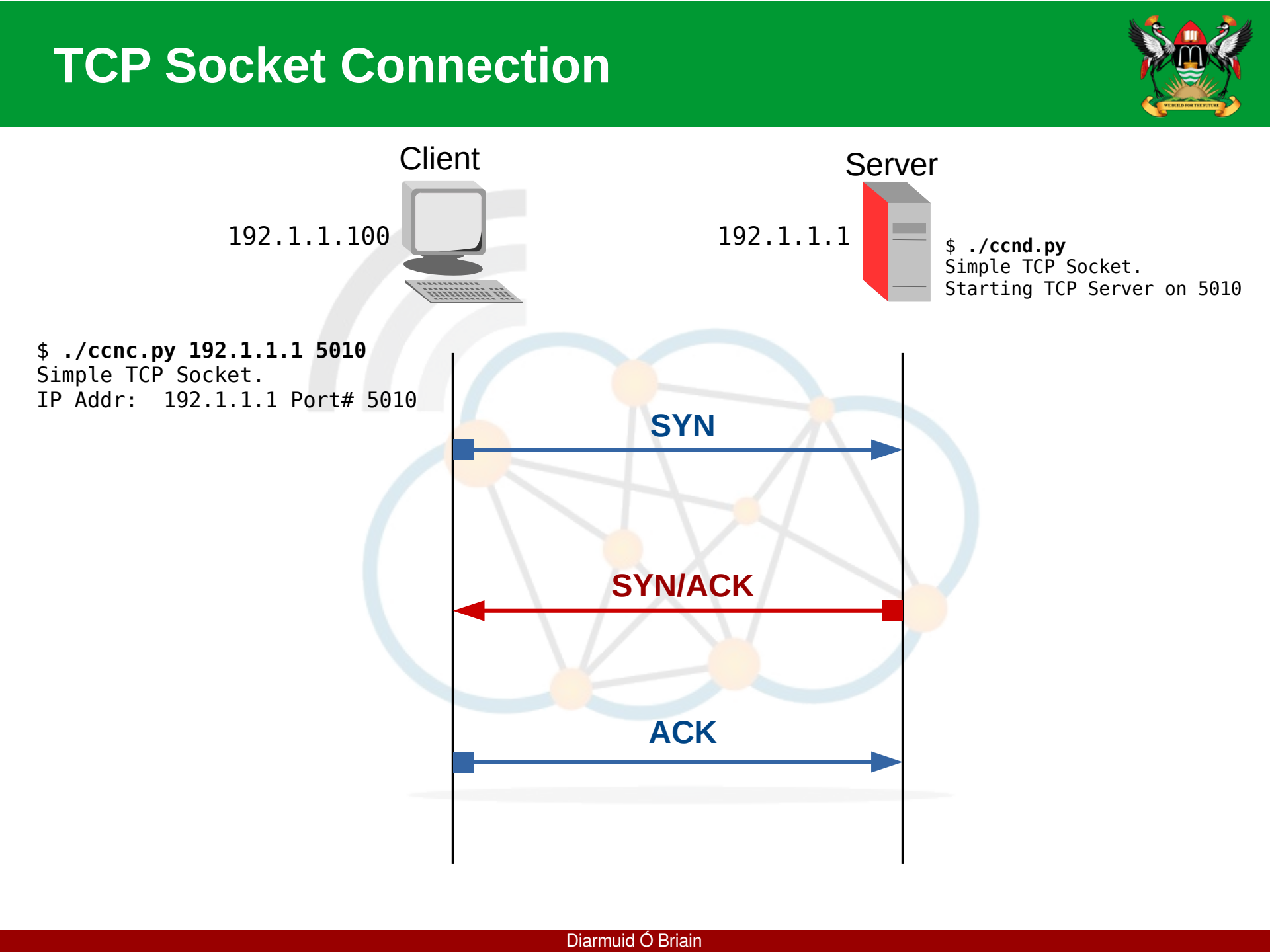
```
root@svr1:/tmp/pycore.46202/svr1.conf# cd /home/nte/TEL-3214-exercises/code
root@svr1:/home/nte/TEL-3214-exercises/code# ./ccnd.py 5050

Welcome to the TEL-3214 Computer Communication Networks daemon (ccnd)
This program is designed to give students an understanding of the
client/server architecture through a simple TCP Socket.

Starting TCP Server on 5050

Connected by ('10.0.0.20', 49655)
Sending received data back to ccnc at 10.0.0.20
```

CCN Course client (ccnc.py)

- Simple TCP Socket.

```
root@lap1:/tmp/pycore.46202/svr1.conf# cd /home/nte/TEL-3214-exercises/code
root@lap1:/home/nte/TEL-3214-exercises/code# ./ccnc.py 10.0.2.10 5050

Welcome to the TEL-3214 Computer Communication Networks client (ccnc)
This program is designed to give students an understanding of the
client/server architecture through a simple TCP Socket.

IP Addr:  10.0.2.10 Port# 5050

What message do you want to pass?: This is a TCP message

Sending message to server at 10.0.2.10 on port number: 5050

Received back:  This is a TCP message
```

TCP Socket Connection



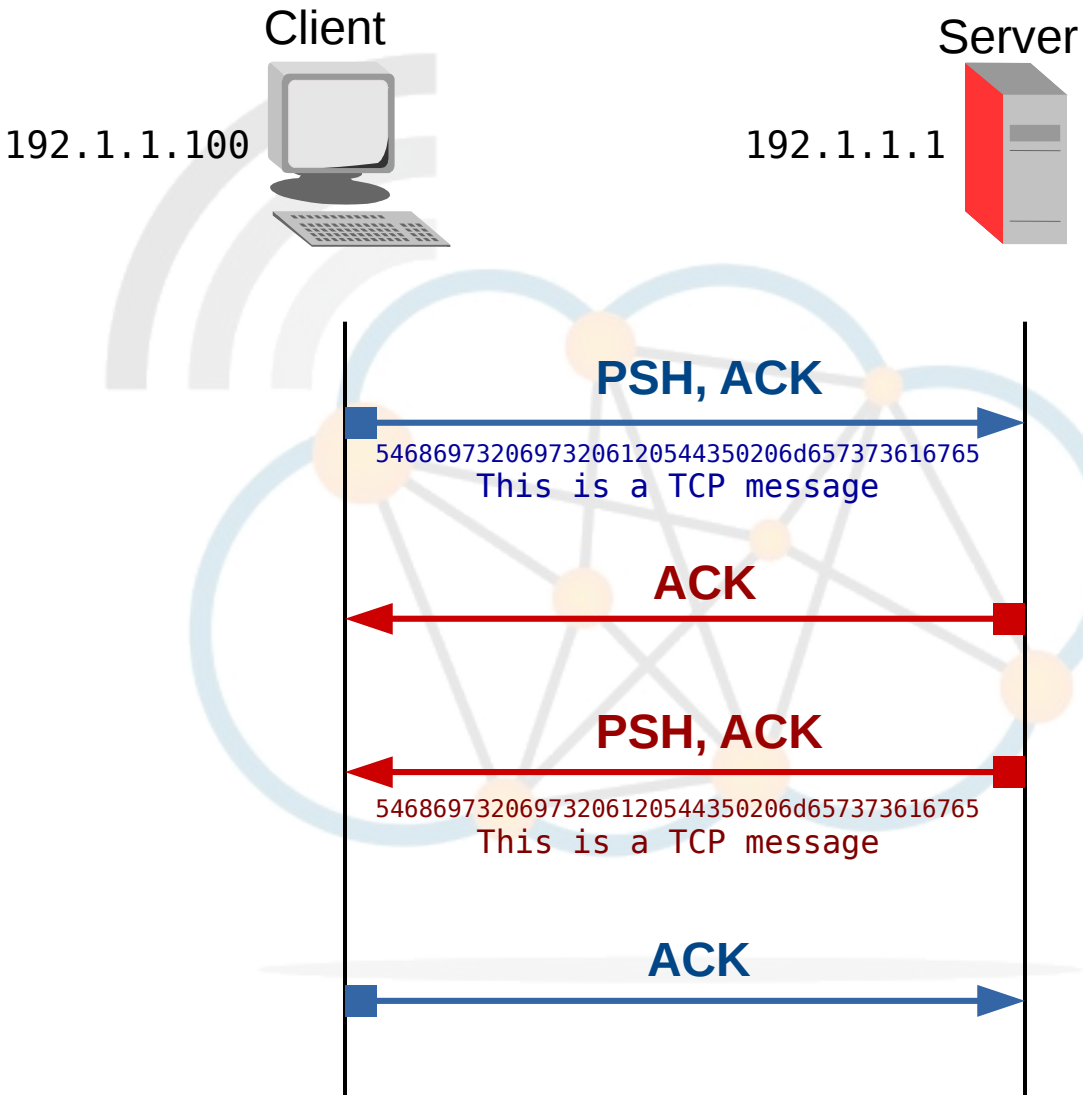
TCP Socket Connection


```
Internet Protocol Version 4, Src: 192.1.1.100, Dst: 192.1.1.1
Transmission Control Protocol, Src Port: 57567, Dst Port: 5560, Seq: 0, Len: 0
  Header length: 40 bytes
  Flags: 0x002 (SYN)
  Window size value: 29200
  Checksum: 0x1034
  Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
```


```
Internet Protocol Version 4, Src: 192.1.1.1, Dst: 192.1.1.100
Transmission Control Protocol, Src Port: 5560, Dst Port: 57567, Seq: 0, Ack: 1, Len: 0
  Header length: 40 bytes
  Flags: 0x012 (SYN, ACK)
  Window size value: 28960
  Checksum: 0x52b2
  Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
```


```
Internet Protocol Version 4, Src: 192.1.1.100, Dst: 192.1.1.1
Transmission Control Protocol, Src Port: 57567, Dst Port: 5560, Seq: 1, Ack: 1, Len: 0
    Header length: 32 bytes
    Flags: 0x010 (ACK)
    Window size value: 229
    Checksum: 0xf1b9
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```

TCP Writes and Reads



TCP Writes and Reads


```
Internet Protocol Version 4, Src: 192.1.1.100, Dst: 192.1.1.1
Transmission Control Protocol, Src Port: 57567, Dst Port: 5560, Seq: 1, Ack: 1, Len: 21
    Header length: 32 bytes
    Flags: 0x018 (PSH, ACK)
    Window size value: 229
    [Calculated window size: 29312]
    [Window size scaling factor: 128]
    Checksum: 0xf372 [validation disabled]
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
Data (21 bytes)

0000  54 68 69 73 20 69 73 20 61 20 54 43 50 20 6d 65   This is a TCP me
0010  73 73 61 67 65                                    ssage
```

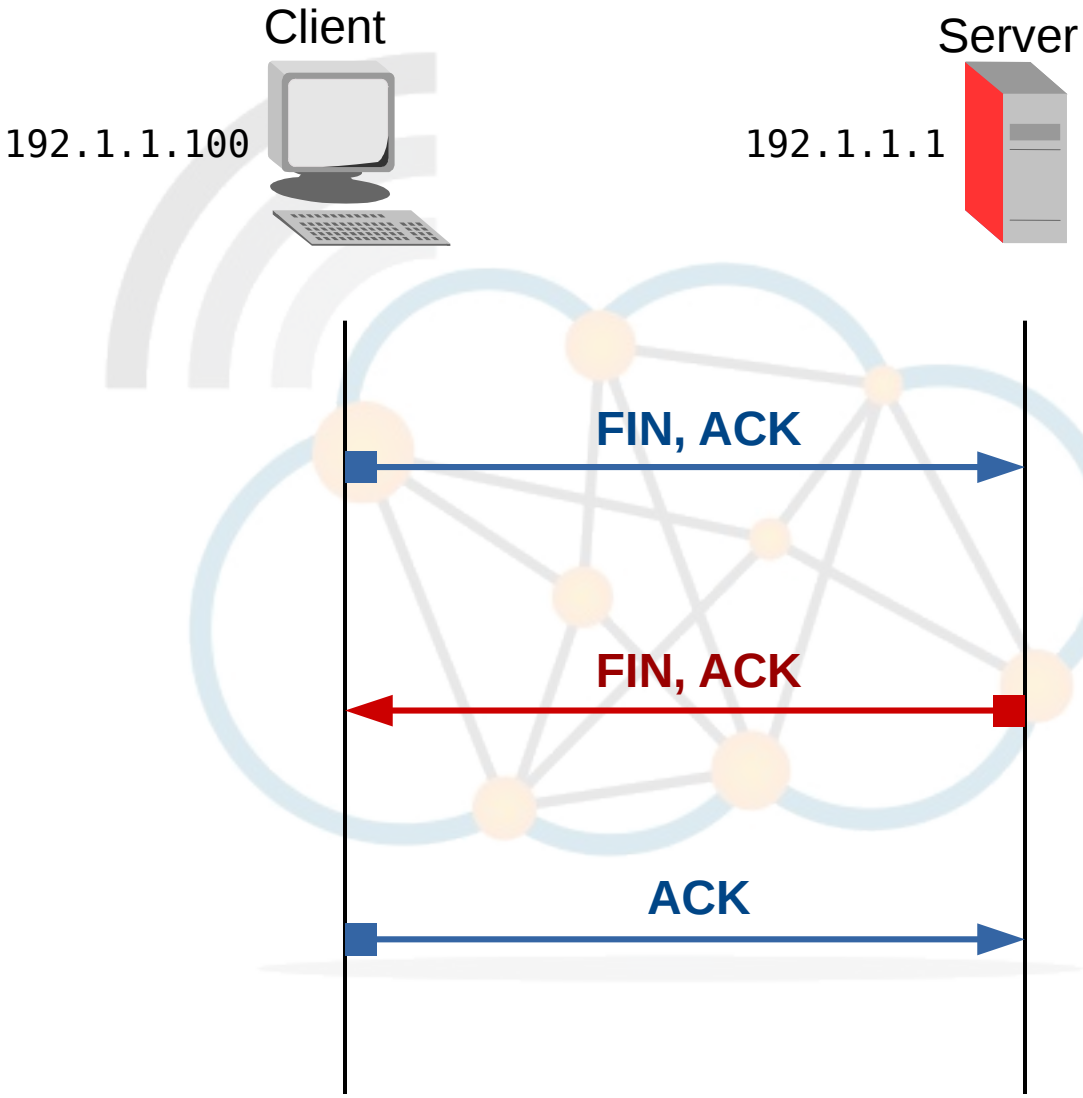

```
Internet Protocol Version 4, Src: 192.1.1.1, Dst: 192.1.1.100
Transmission Control Protocol, Src Port: 5560, Dst Port: 57567, Seq: 1, Ack: 22, Len: 0
    Header length: 32 bytes
    Flags: 0x010 (ACK)
    Window size value: 227
    Checksum: 0xf1a6 [validation disabled]
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```

TCP Writes and Reads


```
Internet Protocol Version 4, Src: 192.1.1.1, Dst: 192.1.1.100
Transmission Control Protocol, Src Port: 5560, Dst Port: 57567, Seq: 1, Ack: 22, Len: 21
  Header length: 32 bytes
  Flags: 0x018 (PSH, ACK)
  Window size value: 227
  Checksum: 0xf35f
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
Data (21 bytes)

0000  54 68 69 73 20 69 73 20 61 20 54 43 50 20 6d 65   This is a TCP me
0010  73 73 61 67 65                                    ssage
```


```
Internet Protocol Version 4, Src: 192.1.1.100, Dst: 192.1.1.1
Transmission Control Protocol, Src Port: 57567, Dst Port: 5560, Seq: 22, Ack: 22, Len: 0
  Header length: 32 bytes
  Flags: 0x010 (ACK)
  Window size value: 229
  Checksum: 0xf18f
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```

TCP Close



TCP Close


```
Internet Protocol Version 4, Src: 192.1.1.100, Dst: 192.1.1.1
Transmission Control Protocol, Src Port: 57567, Dst Port: 5560, Seq: 22, Ack: 22, Len: 0
  Header length: 32 bytes
  Flags: 0x011 (FIN, ACK)
  Window size value: 229
  Checksum: 0xf18e [validation disabled]
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```


```
Internet Protocol Version 4, Src: 192.1.1.1, Dst: 192.1.1.100
Transmission Control Protocol, Src Port: 5560, Dst Port: 57567, Seq: 22, Ack: 23, Len: 0
  Header length: 32 bytes
  Flags: 0x011 (FIN, ACK)
  Window size value: 227
  Checksum: 0xf18f [validation disabled]
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```


```
Internet Protocol Version 4, Src: 192.1.1.100, Dst: 192.1.1.1
Transmission Control Protocol, Src Port: 57567, Dst Port: 5560, Seq: 23, Ack: 23, Len: 0
  Header length: 32 bytes
  Flags: 0x010 (ACK)
  Window size value: 229
  Checksum: 0xf18d [validation disabled]
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
```

SSH

- Server side
  - SSH Daemon (sshd)
  - SFTP Server (sftp-server)
  - ssh-agent
- Remote operations
  - ssh for remote login
  - Sftp
  - Secure Copy (scp).
- Key management
  - ssh-add
  - ssh-keysign
  - ssh-keyscan
  - ssh-keygen.

SSH

```
root@lap1:/root#  ssh nte@10.0.2.10
The authenticity of host '10.0.2.10 (10.0.2.10)' can't be established.
RSA key fingerprint is 53:c2:a2:55:1d:ed:de:16:2a:33:1a:42:7c:e9:4a:84.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.10' (RSA) to the list of known hosts.
nte@10.0.2.10's password: nte

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

nte@svr1:~$ id
uid=1001(nte) gid=1001(nte) groups=1001(nte),27(sudo),128(wireshark)

nte@svr1:~$ ip -4 addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
31: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    group default qlen 1000
    inet 10.0.2.10/24 scope global eth0
       valid_lft forever preferred_lft forever
```

SFTP

```
root@lap1:/root# dd if=/dev/zero of=10mb_file_test_file bs=10485760 count=1
1+0 records in
1+0 records out
10485760 bytes (10 MB) copied, 0.0131757 s, 796 MB/s

root@lap1:/root# ls -la | grep 10mb_file_test_file
-rw-rw-rw-  1 root root 10485760 Feb 26 18:04 10mb_file_test_file

Connect to the Server svr1, review the available commands.


root@lap1:/root# sftp nte@10.0.2.10
nte@10.0.2.10's password: nte
Connected to 10.0.2.10.
```

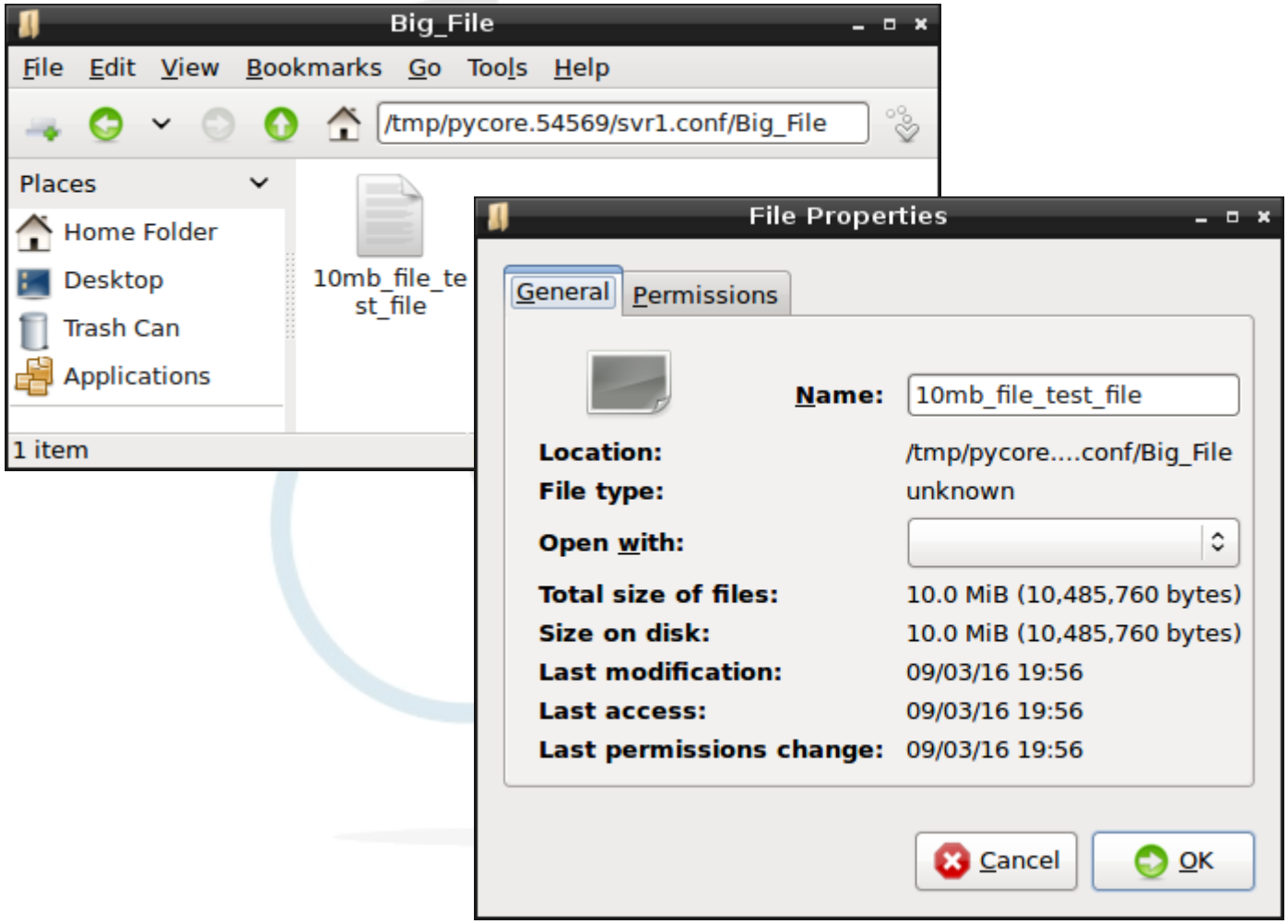
SFTP

```
sftp> ?
Available commands:
bye                                Quit sftp
cd path                            Change remote directory to 'path'
chgrp grp path                     Change group of file 'path' to 'grp'
chmod mode path                    Change permissions of file 'path' to 'mode'
chown own path                     Change owner of file 'path' to 'own'
df [-hi] [path]                         Display statistics for current directory or
                                   filesystem containing 'path'
exit                               Quit sftp
get [-Ppr] remote [local]          Download file
reget remote [local]               Resume download file
reput [local] remote               Resume upload file
help                               Display this help text
lcd path                           Change local directory to 'path'
lls [ls-options [path]]            Display local directory listing
lmkdir path                        Create local directory
ln [-s] oldpath newpath            Link remote file (-s for symlink)
lpwd                               Print local working directory
ls [-1afhlnrSt] [path]             Display remote directory listing
lumask umask                       Set local umask to 'umask'
mget                               As per get but wildcards for multiple downloads
mkdir path                         Create remote directory
mput                               As per put but wildcards for multiple uploads
progress                           Toggle display of progress meter
put [-Ppr] local [remote]          Upload file
pwd                                Display remote working directory
quit                               Quit sftp
rename oldpath newpath             Rename remote file
rm path                            Delete remote file
rmdir path                         Remove remote directory
symlink oldpath newpath            Symlink remote file
version                            Show SFTP version
!command                           Execute 'command' in local shell
!                                  Escape to local shell
?                                  Synonym for help
```

SFTP

```
sftp> cd Desktop
sftp> mkdir Big_File
sftp> cd Big_File
sftp> pwd
Remote working directory: /home/nte/Desktop/Big_File

sftp> lpwd
/root
sftp> lls
10mb_file_test_file

sftp> put 10mb_file_test_file
Uploading 10mb_file_test_file to /home/nte/Desktop/Big_File/10mb_file_test_file
10mb_file_test_file

sftp> ls
10mb_file_test_file

sftp> exit
root@lap1:/root#
 100%   10MB  10.0MB/s   00:00
```

10 MB file


Big_File
File Edit View Bookmarks Go Tools Help
/tmp/pycore.54569/svr1.conf/Big_File
Places
Home Folder
Desktop
Trash Can
Applications
10mb_file_test_file
1 item
File Properties
General
Permissions
Name:
10mb_file_test_file
Location:
/tmp/pycore....conf/Big_File
File type:
unknown
Open with:
Total size of files:
10.0 MiB (10,485,760 bytes)
Size on disk:
10.0 MiB (10,485,760 bytes)
Last modification:
09/03/16 19:56
Last access:
09/03/16 19:56
Last permissions change:
09/03/16 19:56
Cancel
OK

Archiving and compressing

Diarmuid Ó Briain
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

Tape Archive (TAR)

- Archiving utility.

```
root@svr1# ls Big_File/
10mb_file_test_file

root@svr1# tar -cf Big_File.tar Big_File

root@svr1# file Big_File.tar
Big_File.tar: POSIX tar archive (GNU)


root@svr1# tar -tf Big_File.tar
Big_File/
Big_File/10mb_file_test_file


root@svr1# rm -r Big_File


root@svr1# tar -xf Big_File.tar

root@svr1# ls Big_File
10mb_file_test_file
```

GNU ZIP (GZIP)

- Compression tool, compress an archive or file.

```
root@svr1# gzip Big_File.tar

root@svr1# ls -l | grep Big_File
drwxr-xr-x 2 root     root  4096 Mar  9 19:56 Big_File
-rw-rw-rw- 1 root     root 10346 Mar  9 20:08 Big_File.tar.gz
```

To reverse this process use the **gunzip** command.

```
root@svr1# gunzip Big_File.tar.gz

root@svr1# ls -l | grep Big_File
drwxr-xr-x 2 root     root     4096 Mar  9 19:56 Big_File
-rw-rw-rw- 1 root     root 10496000 Mar  9 20:08 Big_File.tar
```

BZIP2

- Compression tool, compress an archive or file.

```
root@svr1# bzip2 Big_File.tar

root@svr1# ls -l | grep Big_File
drwxr-xr-x 2 root     root 4096 Mar  9 19:56 Big_File
-rw-rw-rw- 1 root     root  180 Mar  9 20:08 Big_File.tar.bz2
```

The reverse process is similar to what has been seen for **gunzip**.

```
root@svr1# bunzip2 Big_File.tar.bz2

root@svr1# ls -l | grep Big_File
drwxr-xr-x 2 root     root     4096 Mar  9 19:56 Big_File
-rw-rw-rw- 1 root     root 10496000 Mar  9 20:08 Big_File.tar
```

xz

- Compression tool, compress an archive or file.

```
root@svr1# xz Big_File.tar

root@svr1# ls -l | grep Big_File
drwxr-xr-x 2 root     root 4096 Mar  9 19:56 Big_File
-rw-rw-rw- 1 root     root 1764 Mar  9 20:08 Big_File.tar.xz
```

The reverse process is similar to what has been seen for **gunzip**.

```
root@svr1# unxz Big_File.tar.xz

root@svr1# ls -l | grep Big_File
drwxr-xr-x 2 root     root     4096 Mar  9 19:56 Big_File
-rw-rw-rw- 1 root     root 10496000 Mar  9 20:08 Big_File.tar
```

TAR/GZIP/BZIP2/XZ

- TAR can archive and compress.

```
root@svr1# tar -czvf Big_File.tar.gz Big_File
Big_File/
Big_File/10mb_file_test_file

root@svr1# file Big_File.tar.gz
Big_File.tar.gz: gzip compressed data, last modified: Wed Mar  9 20:23:18
2016, from Unix

root@svr1# tar -cjvf Big_File.tar.bz2 Big_File
Big_File/
Big_File/10mb_file_test_file

root@svr1# file Big_File.tar.bz2
Big_File.tar.bz2: bzip2 compressed data, block size = 900k

root@svr1# tar -cJvf Big_File.tar.xz Big_File
Big_File/
Big_File/10mb_file_test_file

root@svr1# file Big_File.tar.xz
Big_File.tar.xz: XZ compressed data
```

ZIP

- ZIP utility, same as WinZip, archive and compress.

Recursively archive and compress all the file in **Big_File** directory.

```
root@svr1# zip -r Big_File.zip Big_File
updating: Big_File/ (stored 0%)
updating: Big_File/10mb_file_test_file (deflated 100%)
```

To recover the files from the compressed archive use **unzip**.

```
root@svr1# unzip Big_File.zip
Archive:  Big_File.zip
   creating: Big_File/
  inflating: Big_File/10mb_file_test_file
```

Compare

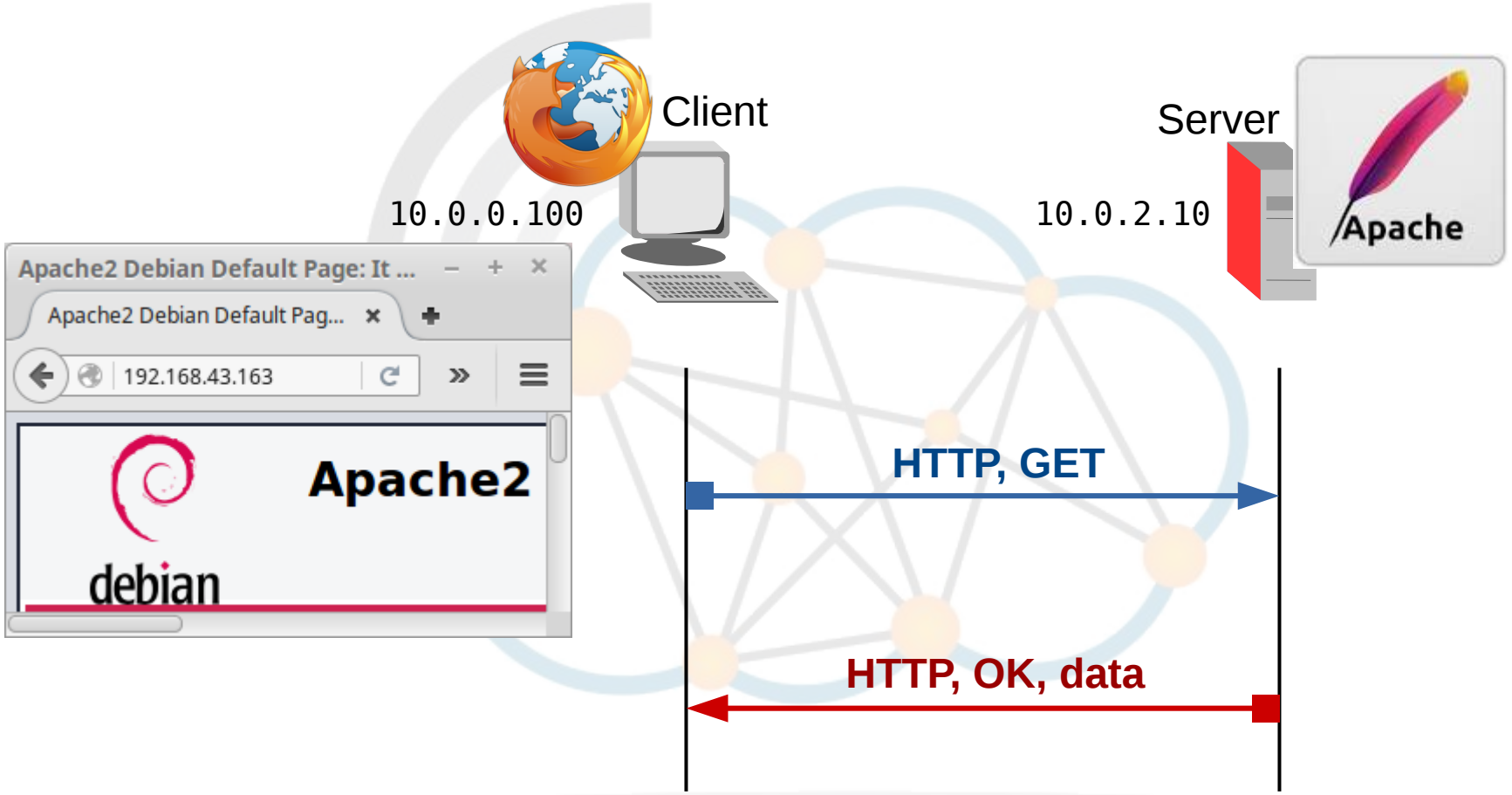
```
root@svr1# ls -la | grep Big_File.
-rw-rw-rw-  1 root     root 10496000 Mar  9 20:08 Big_File.tar
-rw-rw-rw-  1 root     root      180 Mar  9 20:24 Big_File.tar.bz2
-rw-rw-rw-  1 root     root    10333 Mar  9 20:23 Big_File.tar.gz
-rw-rw-rw-  1 root     root     1752 Mar  9 20:26 Big_File.tar.xz
-rw-rw-rw-  1 root     root    10542 Mar  9 20:28 Big_File.zip

root@lap1# sftp root@10.0.2.10
root@10.0.2.10's password: root
Connected to 10.0.2.10.

sftp> mget Big_File.*
Fetching Big_File.tar to /tmp/pycore.54569/svr1.conf/Big_File.tar
Big_File.tar                                  100%   10MB  10.0MB/s   00:00
Fetching Big_File.tar.bz2 to /tmp/pycore.54569/svr1.conf/Big_File.tar.bz2
Big_File.tar.bz2                              100% 1764     1.7KB/s   00:00
Fetching Big_File.tar.gz to /tmp/pycore.54569/svr1.conf/Big_File.tar.gz
Big_File.tar.gz                               100%   10KB  10.1KB/s   00:00
Fetching Big_File.tar.xz to /tmp/pycore.54569/svr1.conf/Big_File.tar.xz
Big_File.tar.xz                               100% 1764     1.7KB/s   00:00
Fetching Big_File.zip to /tmp/pycore.54569/svr1.conf/Big_File.zip
Big_File.zip                                  100%   10KB  10.3KB/s   00:00
```

Hypertext Transfer Protocol (HTTP)

Diarmuid Ó Briain
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

HTTP


Client
10.0.0.100
Server
10.0.2.10
Apache
Apache2 Debian Default Page: It ...
Apache2 Debian Default Pag...
192.168.43.163
Apache2
debian
HTTP, GET
HTTP, OK, data

HTTP

```
root@lap1:/tmp/pycore.46202/lap1.conf# lynx 10.0.2.10
Looking up  '10.0.2.10' first

  Your Terminal type is unknown!
  Enter a terminal type: [vt100] <CR>


                                svr1 web server

   This is the default web page for this server.

   The web server software is running but no content has been added, yet.

     Eth0 - ['10.0.2.10/24', '2001:2::10/64']
```

HTTP GET

```
Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.2.10
Transmission Control Protocol,
  Src Port: 54117, Dst Port: 80, Seq: 1, Ack: 1, Len: 253
Hypertext Transfer Protocol
  GET / HTTP/1.0\r\n
  Host: 10.0.2.10\r\n
    Accept: text/html, text/plain, text/sgml, text/css, app/xhtml+xml, */*;q=0.01\r\n
    Accept-Encoding: gzip, compress, bzip2\r\n
    Accept-Language: en\r\n
    User-Agent: Lynx/2.8.9dev.1 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/3.3.8\r\n
    \r\n
```

HTTP OK

```
Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.0.20
Transmission Control Protocol, Src Port: 80, Dst Port: 54117, Seq: 1, Ack: 254, Len: 500
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 26 Feb 2016 18:33:12 GMT\r\n
    Server: Apache/2.4.10 (Debian)\r\n
    Last-Modified: Fri, 26 Feb 2016 18:32:10 GMT\r\n
    ETag: "115-52cb08464d460"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 277\r\n
    Connection: close\r\n
    \r\n
    Data (277 bytes)

0000  3c 68 74 6d 6c 3e 3c 62 6f 64 79 3e 3c 21 2d 2d   <html><body><!--
0010  20 67 65 6e 65 72 61 74 65 64 20 62 79 20 75 74    generated by ut
0020  69 6c 69 74 79 2e 70 79 3a 48 74 74 70 53 65 72   ility.py:HttpSer
0030  76 69 63 65 20 2d 2d 3e 0a 3c 68 31 3e 73 76 72   vice -->.<h1>svr
0040  31 20 77 65 62 20 73 65 72 76 65 72 3c 2f 68 31   1 web server</h1
0050  3e 0a 3c 70 3e 54 68 69 73 20 69 73 20 74 68 65   >.<p>This is the
0060  20 64 65 66 61 75 6c 74 20 77 65 62 20 70 61 67    default web pag
0070  65 20 66 6f 72 20 74 68 69 73 20 73 65 72 76 65   e for this serve
0080  72 2e 3c 2f 70 3e 0a 3c 70 3e 54 68 65 20 77 65   r.</p>.<p>The we
0090  62 20 73 65 72 76 65 72 20 73 6f 66 74 77 61 72   b server softwar
00a0  65 20 69 73 20 72 75 6e 6e 69 6e 67 20 62 75 74   e is running but
00b0  20 6e 6f 20 63 6f 6e 74 65 6e 74 20 68 61 73 20    no content has
00c0  62 65 65 6e 20 61 64 64 65 64 2c 20 79 65 74 2e   been added, yet.
00d0  3c 2f 70 3e 0a 3c 6c 69 3e 65 74 68 30 20 2d 20   </p>.<li>eth0 -
00e0  5b 27 31 30 2e 30 2e 32 2e 31 30 2f 32 34 27 2c   ['10.0.2.10/24',
00f0  20 27 32 30 30 31 3a 32 3a 3a 31 30 2f 36 34 27    '2001:2::10/64'
0100  5d 3c 2f 6c 69 3e 0a 3c 2f 62 6f 64 79 3e 3c 2f   ]</li>.</body></
0110  68 74 6d 6c 3e                                    html>
```

Asymmetric Key Cryptography

Diarmuid Ó Briain
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

Asymmetric Key Cryptography

- Asymmetric key cryptography / public key cryptography uses asymmetric key algorithms instead of or in addition to symmetric key algorithms.
- Unlike symmetric key algorithms does not require a secure initial exchange of one or more secret keys to both sender and receiver.
- A mathematically related key pair is created, a secret private key and a public key the latter which is published.
- These keys allow protection of the **authenticity** of a message by creating a digital signature of a message using the private key, which can be validated using the public key.
- It also allows for the protection of the messages **confidentiality** and **integrity**, by public key encryption, encrypting the message using the public key, which can only be decrypted using the private key.
- Public key cryptography is employed by many cryptographic algorithms and cryptosystems.
- It is used in standards such as TLS/SSL, PGP, and GnuPG.

Key pairs

- The generation of key pairs requires the use of intractable problems called trapdoor functions which are functions that are easy to compute in one direction, yet believed to be difficult to compute in the opposite direction without special information, called the "trapdoor".
- An intractable problem is a problem for which there is no efficient means of solving.
- The public key cryptographic intractable problems used to date are based either on factoring prime numbers or discrete logarithms.

Example, discrete logarithm problem

- Take a prime number $m = 29$ as the modulus (public key).
- Primitive roots of 29 are: 2, 3, 8, ***10,*** 11, 14, 15, 18, 19, 21, 26, 27.
- So taking and a base $b = 10$ (the trapdoor)
- Alice chooses a secret $y = 8$ (Private Key).
- Alice sends Bob $w = b^y \bmod m = 10^8 \bmod 29 = 25$
- Bob chooses a secret $z = 11$ (Private Key).
- Bob sends Alice $x = b^z \bmod m = 10^{11} \bmod 29 = 2$
- Alice computes $s = x^y \bmod m = 2^8 \bmod 29 = 24$
- Bob computes $s = w^z \bmod m = 25^{11} \bmod 29 = 24$
- Alice and Bob now share a secret, in this case 24 without it being transferred across the transmission path and without either Alice or Bob sharing their private keys.

Example, discrete logarithm problem

- The trapdoor can remain small but the public and private keys are typically over 300 digits.

- For example:

4305 0241 20D9 18FA DF8D EC2D EFD5 FD35 89C9 E069
EA95 FD20 5E35 F3B5 ED31 D4FC D6E4 4811 5D86 CD8F
CAFA 362F 922C F01C 2F40 D544 2654 D0D2 2881 D653
DA2C 4203 D266 E2D2 DC02 0301 2001

Key exchange protocols

- Diffie-Hellman key protocol
  - In 1976 Whitfield Diffie and Martin Hellman, who, influenced by Ralph Merkle's work on public-key distribution went down the discrete log route when developing what became known as Diffie-Hellman key exchange method.
- El Gamal
  - El Gamal is based on Diffie-Hellman method. It was described by Taher Elgamal in 1985.
  - It is used in the free GNU Privacy Guard software, recent versions of PGP, and other crypto-systems.

RSA

- In 1977 Ronald Rivest, Adi Shamir and Len Adleman developed an algorithm using factoring of prime numbers. This algorithm became known as RSA.
- Taking two large prime numbers we will call ‘B’ and ‘Q’. Multiply these numbers to generate ‘N’:
  - ***N*** = B * Q
- Select another number ‘***e***’ such that:
  - ***e*** < ***N***
  - ***e*** and (***N*** -1)(Q – 1) are relatively prime (no common factors except 1)
- Find a number ‘***p***’ such that:
  - (***ep*** - 1) mod(B - 1)(Q - 1) = 0
- Distribute ‘***e***’ and ‘***N***’ as the public key and keep ‘***p***’ as the private key.
- For Alice to send an encrypted message she sends:
  - ***{CT}*** = ***{PT}***$_e$ mod ***N***
- Bob receives and retrieves the message by:
  - ***{PT}*** = ***{CT}***$_p$ mod ***N***

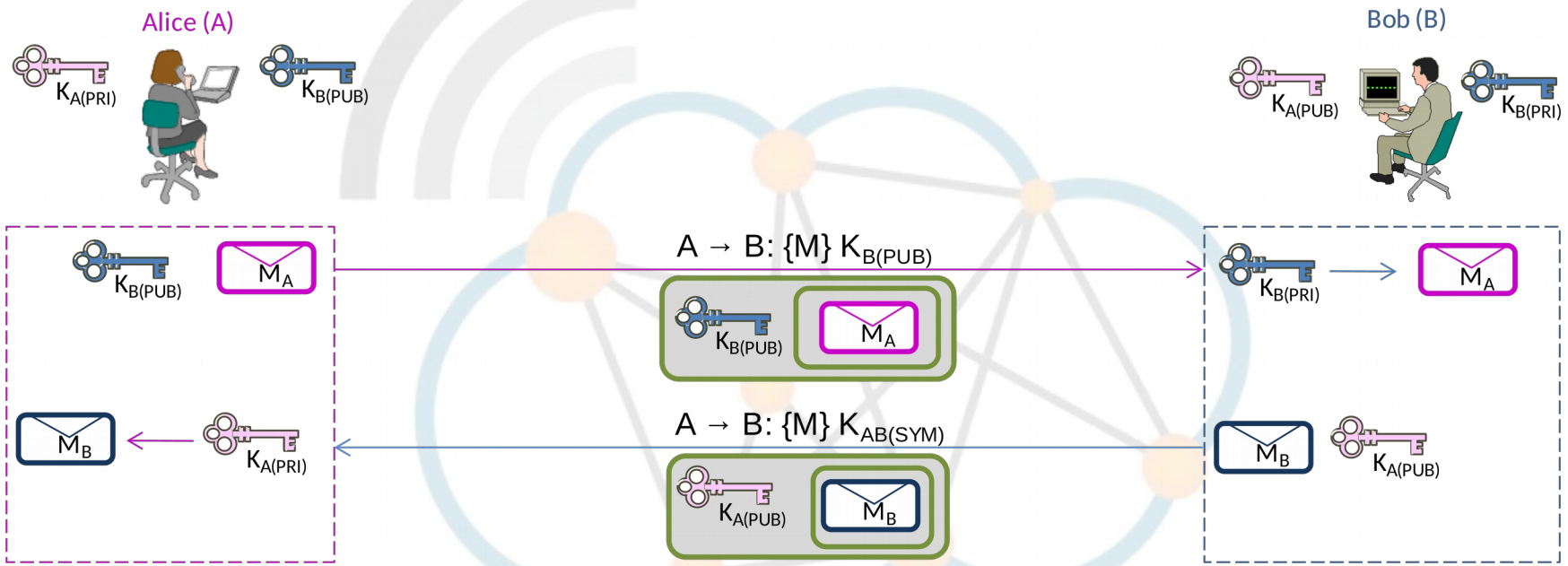
Elliptic curve cryptography (ECC)

- Another intractable problem that is used is the assumption that finding the discrete logarithm of an elliptic curve element is infeasible.
- The size of the elliptic curve determines the difficulty of the problem.
- It is believed that a smaller group can be used to obtain the same level of security as RSA-based systems.
- Using a small group reduces storage and transmission requirements.

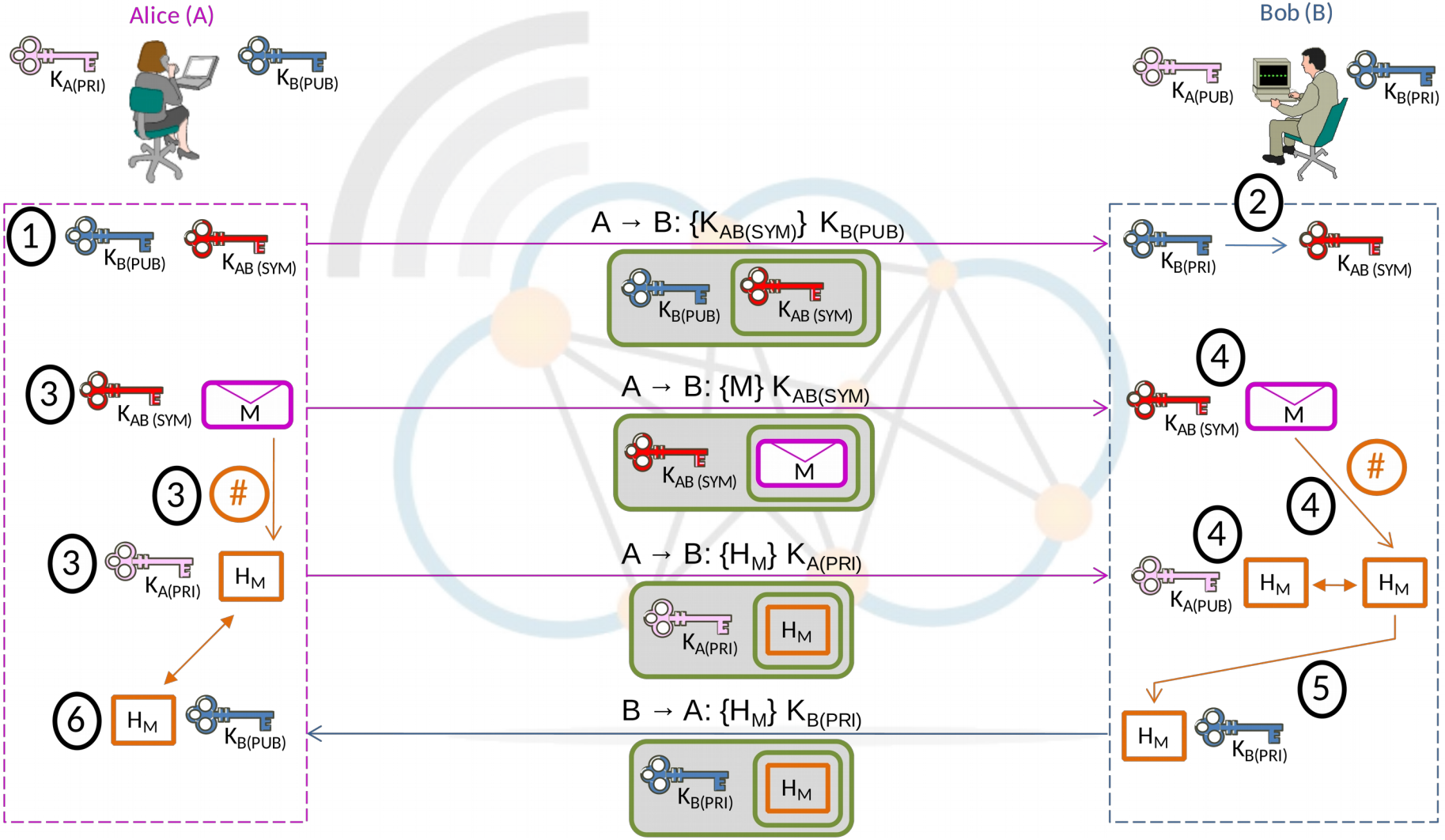
Asymmetric Key Protocol summary

Algorithm	Name	Mode	Block size	Keys	Other
RSA	Ron Rivest, Adi Shamir & Len Adleman	Factoring	Variable	1024 – 2048	
Diffie Hellamn	Whitfield Diffie & Martin Hellman	Discrete Log	Variable	Variable	Only used for key exchange
ECC	Elliptical Curve Cryptography	Discrete Log	Variable	80 → 512	160 bits key is equivalent to 1024 bits in RSA

Asymmetric Key Cryptography



The hybrid system



Key management

How to make the public keys available. For this we need a Public Key Infrastructure (PKI). This is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

VeriSign Entrust

- Certificate Authorities (CA)
  - CAs are web sites that publish the key bound to a given user.
  - This is achieved using the CA's own key, so that trust in the user key relies on one's trust in the validity of the CA's key.
  - The mechanism that binds keys to users is called the Registration Authority (RA), which might or might not be separate from the CA.
  - The key-user binding are established, depending on the level of assurance the binding has, by software or under human supervision.
  - The term trusted third party (TTP) may also be used for certificate authority (CA). Moreover, PKI is itself often used as a synonym for a CA implementation.
  - The ITU-T standard for Certificate Authority is included within the X.509 system.

Key management

- Web of Trust
  - An alternative approach to the problem of public authentication of public key information is the web of trust scheme, which uses self-signed certificates and third party attestations of those certificates.
  - PGP and GnuPG are examples of implementations of the web of trust model.
  - They allow the use of e-mail digital signatures for self-publication of public key information.
  - It is relatively easy to implement one's own Web of Trust.

Implementations

- Privacy Enhanced Mail (PEM)
  - Early IETF proposal for securing email using public key cryptography.
  - Depended on prior deployment of a hierarchical public key infrastructure (PKI)
  - Not widely deployed.
- Pretty Good Privacy (PGP)
  - Signing, encrypting and decrypting e-mails to increase the security of e-mail communications.
  - PGP follows the OpenPGP standard (RFC 4880) for encrypting and decrypting data.
  - Web of trust and Certificate Authority (CA).
  - GnuPG is the GNU project's complete and free implementation of the OpenPGP standard.
- Secure/Multipurpose Internet Mail Extensions (S/MIME)
  - MIME is the standard that extends the format of e-mail to support:
    - Text in character sets other than ASCII.
    - Non-text attachments
    - Message bodies with multiple parts
    - Header information in non-ASCII character sets
  - S/MIME standard for adding cryptographic signature & encryption services to MIME data.

GNU Privacy Guard (GPG)

- GnuPG free implementation of the OpenPGP RFC4880.
- Allows to encrypt and sign data and communication, features a versatile key management system as well as access modules for all kinds of public key directories.
- Project Gpg4win, Windows version of GnuPG.

Private Key

```
$ gpg --gen-key

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1

What keysize do you want? (2048) 2048

Key is valid for? (0) 1w
Key expires at Wed 16 Mar 2016 14:04:20 EAT
Is this correct? (y/N) y

Real name: Ada Lovelace
Email address: alovelace@mak.ac.ug
Comment: March Key

Enter passphrase: babbage
Re-enter passphrase: babbage

gpg: key 6E64AF4C marked as ultimately trusted
```

Public Key

```
$ gpg --armor --output pubkey.txt --export 'Ada Lovelace'

$ cat pubkey.txt

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1

mQENBFbgA5sBCAC12JYl3KtCjH7jOtfZWBw7gISy5Zl8Y4WMQnEsD7F/na1xQqWB2kN8ka2MzBCyUFlWQ6sJu/
8jfkzS3YGy4eCa7OZeAdQgZ4EQU+eC1rIo8cLhPA+jyL5EacQ+jG4kBDsLD+kD8AA55whmAGoapK2lZNyX8tuo
W7Ex94BkATB3EgwJ/0Q53aGrsH93BhIEesc32duvpS0uRe9xY+iEnU9ZquCE6hdCqJBXHo2HdCqs2nN8os8Alw
AfLvN7uRc4Yv7qi5tpWM5+9L30lZlm2/Ydkl2WPxotkCp6mqp+RvZmL7w6hh/dmflZ/Ts+SzrPMdt9QtE/hJA/
J/j8uOedc8jABEBAAG0K0FkYSBMb3ZlbGFjZSAoTWFyY2ggS2V5KSA8YWxvdmVsYWNlQGMycy5pZT6JAT4EEwE
CACgFAlbgA5sCGwMFCQAJOoAGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEPMVgONuZK9MqIEH/1B3BHTjJ
GsXbWsobJnKMJuJeHNaL+9ibmHkgPK3r1K77D8n0xyO4/k0rpS/BNbp2e9VhEbIpk3/tgIJOGgEhm7ckSmTZSf
OyTBi5YO0c/GzhAptNKDbk+qSRVgoV+qtIaeEPBvUWthZzpa6qRO8b24hdi7QwR354Lf2ZOnU/WY5/DBGxl4+N
GJ3BIN5wB7yL8LPeHTAyseftgN0wR6C9AwEXx7mW0kBLLFVEmwAB6sJzXvgOOsRpQ5Wr5bF5C1CWf5MG+VQ0ab
jnteP3I6YibQcDEExqDXfo1nebVzKu1Nke7bCef6jvrEJ6W2BlxAGf0Le8c2+SB5QsJFEAGmBSG5AQ0EVuADmw
EIAMGIj2BqX2OP2kl7GYXaiaV+xSxndNR0Rtv7yvehKbJ9YhmplxyHLOIXBqoFWC9YUPcoy40HmhfPrhXrvIA5
Uiema6dm/BFBOtnrpHM0JWHgBwmk3GOQWH2WKHwlRjIhc36l93wesuJYENskIy/WtLEfiuvkS4ZAfiEhw91VVp
0LFTrBahprOgYzcOhra78RhvxI8/vTS03a1GYryuOQoArCjj+TFNMhGIVgxIY2XWxOlKO7hh75VRRrbM6dhZJk
emLKPiKzqbpPfpQaCN11kytQLtJ+r0KHLrv3GjhGaChRehDSkVoSltPzsYpSslj/bG5jK0BqmTRzNnOUXQjWXv
sAEQEAAYkBJQQYAQIADwUCVuADmwIbDAUJAAk6gAAKCRDzFYDjbmSvTBUtB/0cxtlyK9jB82rlQVCNViJIsfnK
YC+wZ4h84HhoCpzyTBweRm1nnSNu06paps+rS/GXQ0yOOfT4b/NALv5iJwKANRqkShH4LsxbGYd8Ps/jMEc8lR
nSTNwlHnKGzjSco9wGnF/A2omqc2gdLAFlHZPUJDnzhG2H5jHvgwJs6Oncgs5FyjtA/FnUUqMzy+ITQCbYQEnD
Qn8CmNyBVnxz04u+Td2ajRznD3V29UvXgTaG7gU5842CNsLiezrfqqPgnNnRISxpAboy3xCpUbqBG/i8z6hNwG
DBZRGuwKROAYC5dNDE+SBugYub16SDkhe2dR5tGbURFPSeOdLpf17ONf3/=JYlA

-----END PGP PUBLIC KEY BLOCK-----
```

Key distribution

- The public key can be freely distributed, posted on a web site, or otherwise distributed.
- It can also be registered with public keyservers so that others can retrieve the key without having to contact the owner directly.

```
$ gpg --send-keys 'Ada Lovelace' --keyserver hkp://subkeys.pgp.net
```

Encrypting a file for personal use

```
$ ls -la | grep MyFile
-rw-r--r--  1 dobriain dobriain     74 Mar  9 14:23 MyFile.txt

$ cat MyFile.txt
MyFile
======

This is a file used as part of the exercise on encryption.


$ gpg --encrypt --recipient 'Ada Lovelace' MyFile.txt

$ ls -la | grep MyFile
-rw-r--r--  1 dobriain dobriain     74 Mar  9 14:23 MyFile.txt
-rw-r--r--  1 dobriain dobriain    406 Mar  9 14:23 MyFile.txt.gpg

Use the file command to interrogate the file type and the cat command to
view the contents.

$ file MyFile.txt
MyFile.txt: ASCII text

$ file MyFile.txt.gpg
MyFile.txt.gpg: data
```

Encrypting a file for personal use

```
$ cat MyFile.txt
MyFile
======

This is a file used as part of the exercise on encryption.
```

```
$ cat MyFile.txt.gpg
ï¿½#
ï¿½!ï¿½l00#ï¿½Pï¿½#ï¿½aï¿½ï¿½Eï¿½ï¿½#ï¿½ï¿½6ï¿½VU+ï¿½ï¿½vU`<ï¿½ï¿½#<"ï¿½.ï¿½i}|ï¿½ï¿½ï¿½ï¿½]
                                                                vmï¿½ï¿½#Tq<ï¿½&bï¿½ï¿½jï¿½ï¿½|
#ï¿½hï¿½ï¿½ï¿½ï¿½Gï¿½Hï¿½-dï¿½gï¿½ rï¿½ï¿½$ï¿½Oï¿½Iï¿½ï¿½UBï¿½
        ï¿½2%ï¿½ï¿½#ï¿½ï¿½ï¿½ï¿½#)0ï¿½ï¿½Oï¿½fï¿½Jï¿½Oï¿½ï¿½^#Wp#ï¿½ï¿½ï¿½0+<ï¿½ï¿½
                                                                ï¿½ï¿½[ï¿½ï¿½#6ï¿½ï¿½Xï¿½ï¿½#K?M#ï¿½Hbï¿½ï¿½#
ï¿½ï¿½~##)ï¿½4ï¿½ï¿½0  !eï¿½8'ï¿½&ï¿½ï¿½@ï¿½ï¿½Dbï¿½Xï¿½nï¿½'ï¿½9ï¿½`vï¿½#ï¿½ï¿½sNcï¿½ï¿½ï¿½Zï¿½F$^ï¿½?ï¿½ï¿½Pï¿½tUL#ï¿½^ï¿½o#_ï¿½
%ï¿½ï¿½#ï¿½ï¿½ï¿½LAï¿½ï¿½ï¿½~ï¿½ï¿½hï¿½fï¿½ï¿½#ï¿½ï¿½#ï¿½_ï¿½ï¿½
ï¿½ï¿½ï¿½Rc2
        ï¿½ï¿½ï¿½ï¿½ï¿½@#ï¿½\#ï¿½2ï¿½ï¿½&ï¿½=#,#D
ï¿½6ï¿½#Nï¿½ï¿½#ï¿½ï¿½ï¿½ï¿½cï¿½ï¿½#Z#6I
                          ï¿½ï¿½,Nï¿½rï¿½=(0ï¿½ï¿½ï¿½ï¿½#ï¿½#ï¿½
```

Decrypting a file for personal use

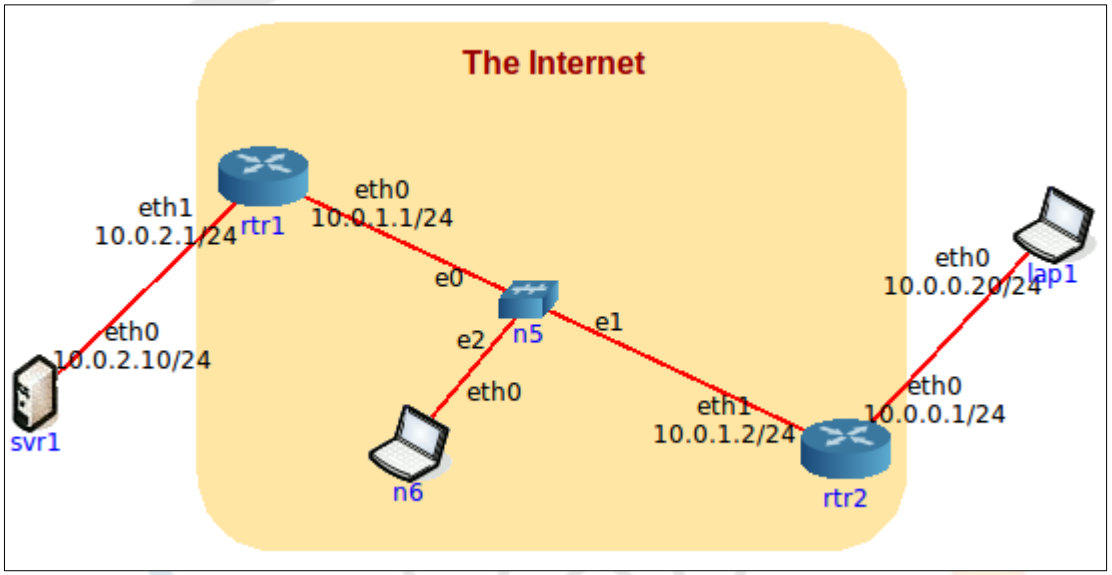
```
$ gpg --output MyFile2.txt --decrypt MyFile.txt.gpg
You need a passphrase to unlock the secret key for
user: "Ada Lovelace (March Key) <alovelace@mak.ac.ug>"
2048-bit RSA key, ID DC6CB630, created 2016-03-09 (main key ID 6E64AF4C)

gpg: encrypted with 2048-bit RSA key, ID DC6CB630, created 2016-03-09
      "Ada Lovelace (March Key) <alovelace@mak.ac.ug>"

Enter passphrase: babbage
```

Check the new file and use the **diff** command to confirm it is identical to the original file **MyFile.txt**.

```
$ cat MyFile2.txt
MyFile
======

This is a file used as part of the exercise on encryption.

$ diff MyFile.txt MyFile2.txt
```

Passing encrypted files to another person

Decript file here



Encrypt file here

f(secretFile.txt.gpg)svr1{pri}

⇩

secretFile.txt

f(secretFile.txt)svr1{pub}

⇩

secretFile.txt.gpg

Passing encrypted files to another person

- Ensure the other person has the public key.

```
root@lap1# echo "This is my little secret" > secretFile.txt


root@lap1# cat secretFile.txt
This is my little secret


root@lap1# gpg --encrypt --recipient 'Ada Lovelace' secretFile.txt


root@lap1# ls secretFile*
secretFile.txt   secretFile.txt.gpg


root@lap1# file secretFile.txt.gpg
secretFile.txt.gpg: PGP RSA encrypted session key - keyid: 9EBC8885
982DEE5 RSA (Encrypt or Sign) 2048b .
```

Import public key from another person

- Import public key.

```
root@lap1# gpg --import pubkey.txt

gpg: key 3AAB4367: "Ada Lovelace (Ada Lovelace March key) <alovelace@cedat.mak.ac.ug>" not
changed

gpg: Total number processed: 1

gpg:               unchanged: 1



root@lap1# gpg --list-keys

/root/.gnupg/pubring.gpg

------------------------

pub   2048R/3AAB4367 2016-03-09 [expires: 2016-03-10]

uid                  Ada Lovelace (Ada Lovelace March key) <alovelace@cedat.mak.ac.ug>

sub   2048R/E5DE8209 2016-03-09 [expires: 2016-03-10]
```

Decrypted files from another person

- With the originators public key public key.

```
root@svr1# ls secret*
secretFile.txt.gpg

root@svr1# gpg --output lap1_secret.txt --decrypt secretFile.txt.gpg

You need a passphrase to unlock the secret key for
user: "Ada Lovelace (Ada Lovelace March key) <alovelace@cedat.mak.ac.ug>"
2048-bit RSA key, ID E5DE8209, created 2016-03-09 (main key ID 3AAB4367)

Enter passphrase: babbage

gpg: encrypted with 2048-bit RSA key, ID E5DE8209, created 2016-03-09
      "Ada Lovelace (Ada Lovelace March key) <alovelace@cedat.mak.ac.ug>"

root@svr1# cat lap1_secret.txt
This is my little secret
```

Digitally signing a file

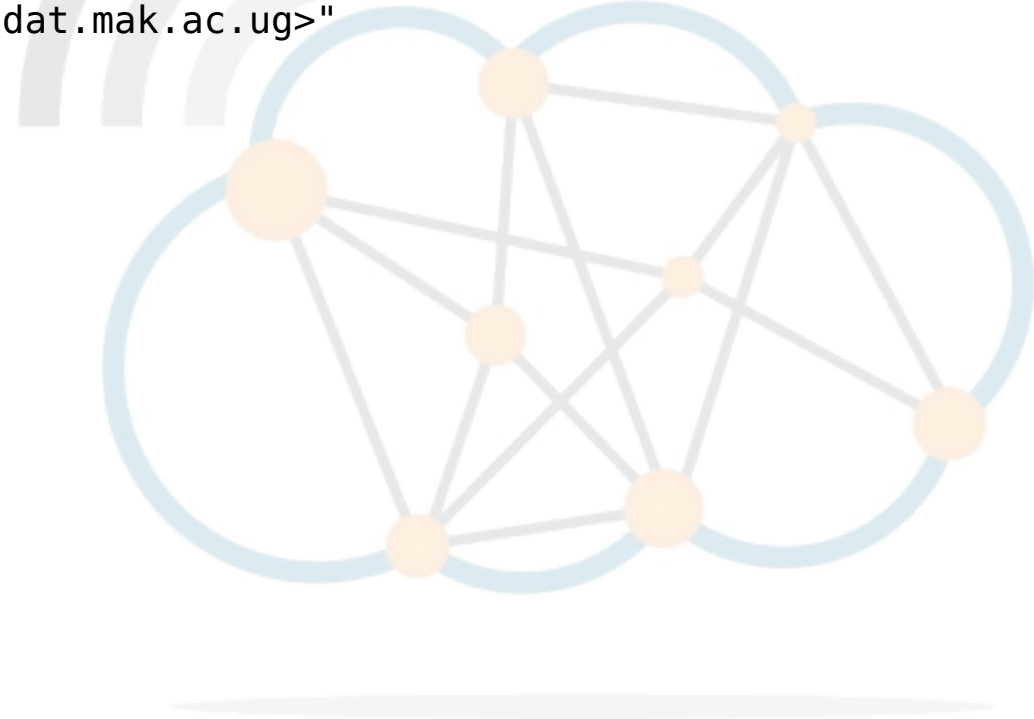
```
root@svr1# echo 'This file will be signed" > sign.txt

root@svr1# gpg --armor --detach-sign sign.txt

You need a passphrase to unlock the secret key for
user: "Ada Lovelace (Ada Lovelace March key) <alovelace@cedat.mak.ac.ug>"
2048-bit RSA key, ID 3AAB4367, created 2016-03-09
Enter passphrase: babbage

root@svr1# ls -la sign*
-rw-rw-rw- 1 root root  25 Mar  9 19:18 sign.txt
-rw-rw-rw- 1 root root 473 Mar  9 19:20 sign.txt.asc

root@svr1# file sign.txt.asc
sign.txt.asc: PGP signature Signature (old)
```

Digitally signing a file

```
root@svr1# cat sign.txt.asc
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQEcBAABAgAGBQJW4HdvAAoJENeMAnc6q0NnQekH/3bck0fGF3FSblpQSeVfrZLJ
sCpGNvLeDv+PpyPCLRtPqwHJlCGMFsP6FCh9d07EYJIYnpHuvnwSPaJCsXqS6OcX
f11vbSo24BLWYDN/T9v7Kt3ui7jEhUYqQNZQXMzlciVrRpYqU5F4vQClTChQXZ2l
9R71QuOGi98AsAZfitAXU3L3SLPxHwieJefqsuWgLqI75uuB2atoy+FvrFSQ7gdv
nW9ylvehHFLtyXwKMQUZ5OSGW/DUl0M6CRVofu4aY9BsIHV5z9yiMiQG3Vi2t5Kl
/4YAzN34jy0YHPDTFrv3qCdgGtuB/Zv9/6CkYYRjP4XyhBtJM74483lDF+hJzjU=
=SS9+
-----END PGP SIGNATURE-----
```

Verify a digitally signed file

```
# gpg --verify sign.txt.asc sign.txt

gpg: Signature made Wed 09 Mar 2016 19:20:15 GMT using RSA key ID
3AAB4367

gpg: Good signature from "Ada Lovelace (Ada Lovelace March key)
<alovelace@cedat.mak.ac.ug>"
```



Applications Lab

Diarmuid Ó Briain
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

Thank You

Diarmuid Ó Briain
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com