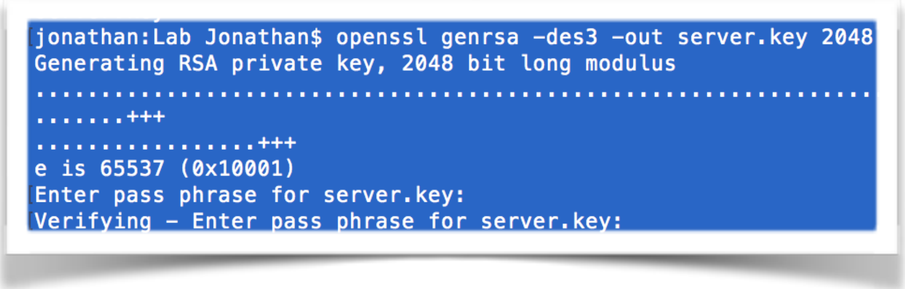
Advance Security
Lab 9
Student Name: Jonathan Riordan
Student ID: C13432152

Self Signed Certificate

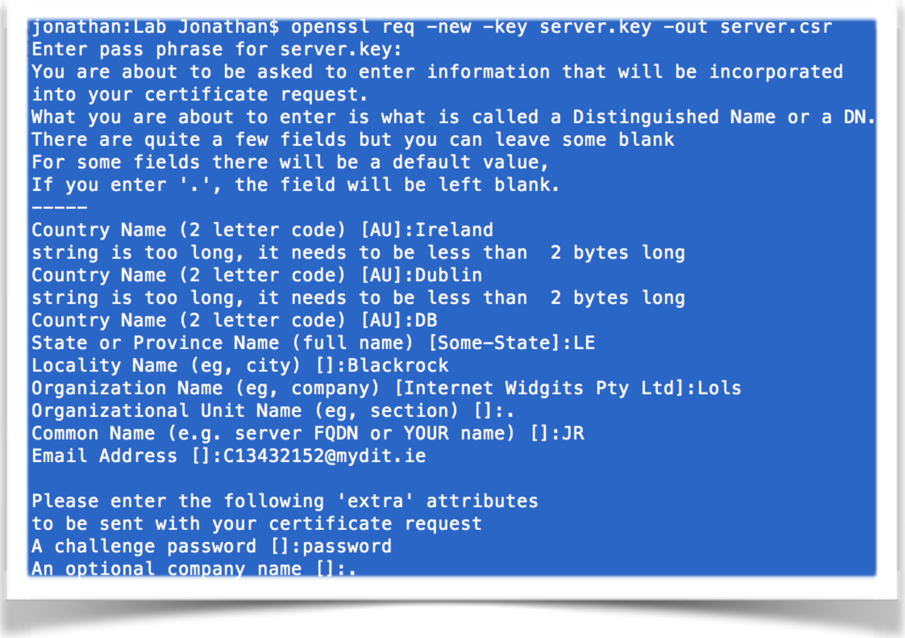
Question 1

Part 1.



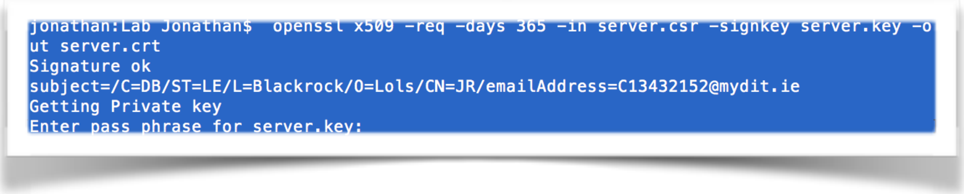
```
jonathan:Lab Jonathan$ openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus
................................................................
.......+++
.................+++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
```

Part 2.



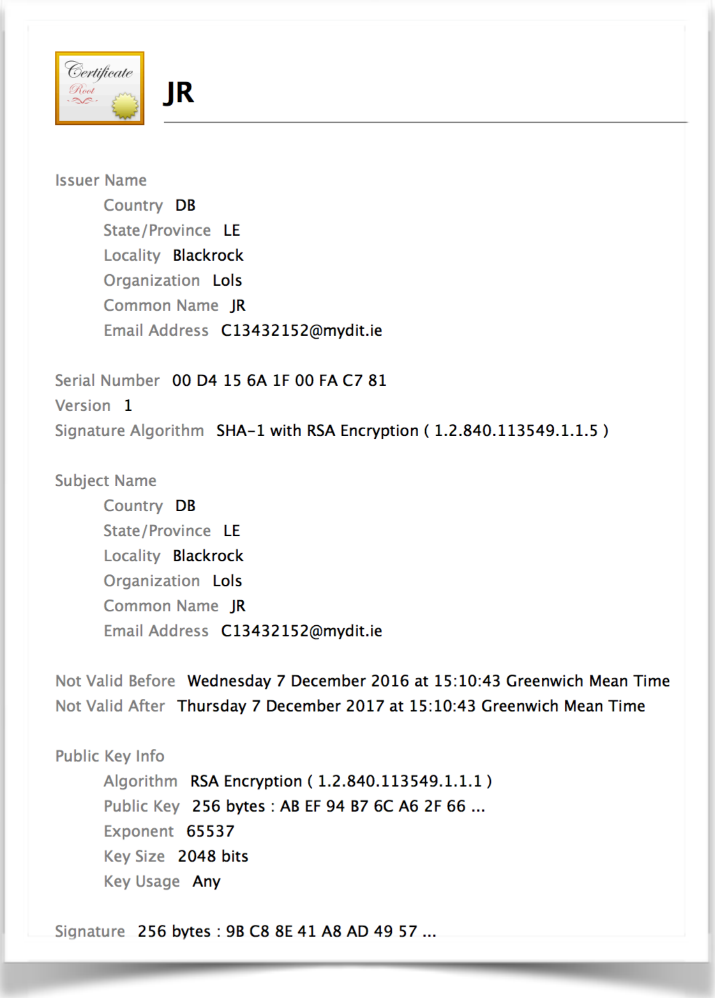
```
jonathan:Lab Jonathan$ openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:Ireland
string is too long, it needs to be less than  2 bytes long
Country Name (2 letter code) [AU]:Dublin
string is too long, it needs to be less than  2 bytes long
Country Name (2 letter code) [AU]:DB
State or Province Name (full name) [Some-State]:LE
Locality Name (eg, city) []:Blackrock
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Lols
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:JR
Email Address []:C13432152@mydit.ie

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:.
```

Part 3.



```
jonathan:Lab Jonathan$  openssl x509 -req -days 365 -in server.csr -signkey server.key -o
ut server.crt
Signature ok
subject=/C=DB/ST=LE/L=Blackrock/O=Lols/CN=JR/emailAddress=C13432152@mydit.ie
Getting Private key
Enter pass phrase for server.key:
```

Below is the screen shot of the self certificate I created. The public key 256bytes and the algorithm is RSA algorithm.



JR

Issuer Name
Country DB
State/Province LE
Locality Blackrock
Organization Lols
Common Name JR
Email Address C13432152@mydit.ie

Serial Number 00 D4 15 6A 1F 00 FA C7 81
Version 1
Signature Algorithm SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)

Subject Name
Country DB
State/Province LE
Locality Blackrock
Organization Lols
Common Name JR
Email Address C13432152@mydit.ie

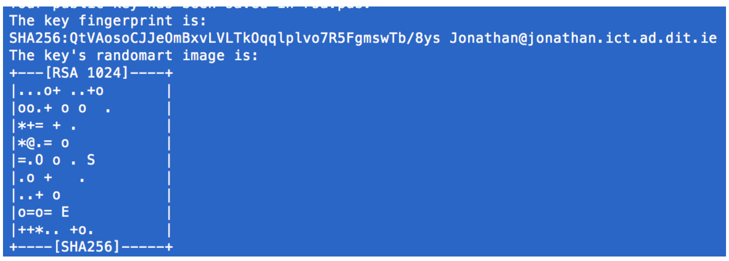
Not Valid Before Wednesday 7 December 2016 at 15:10:43 Greenwich Mean Time
Not Valid After Thursday 7 December 2017 at 15:10:43 Greenwich Mean Time

Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Public Key 256 bytes : AB EF 94 B7 6C A6 2F 66 ...
Exponent 65537
Key Size 2048 bits
Key Usage Any

Signature 256 bytes : 9B C8 8E 41 A8 AD 49 57 ...

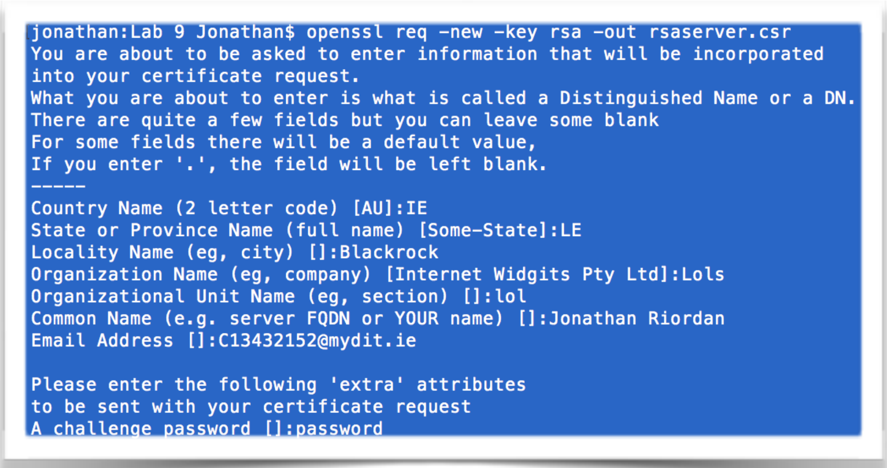
Question 2.

Part 1.
The command to create a 1024 bit key is ssh-keygen -t rsa -b 1024

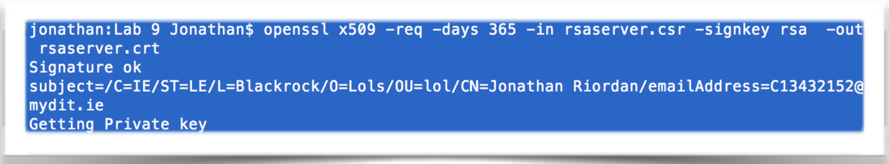


```
The key fingerprint is:
SHA256:QtVAosoCJJeOmBxvLVLTkOqqlplvo7R5FgmswTb/8ys Jonathan@jonathan.ict.ad.dit.ie
The key's randomart image is:
+---[RSA 1024]----+
|...o+ ..+o       |
|oo.+ o o  .      |
|*+= + .          |
|*@.= o           |
|=.O o . S        |
|.o +   .         |
|..+ o            |
|o=o= E           |
|++*.. +o.        |
+----[SHA256]-----+
```

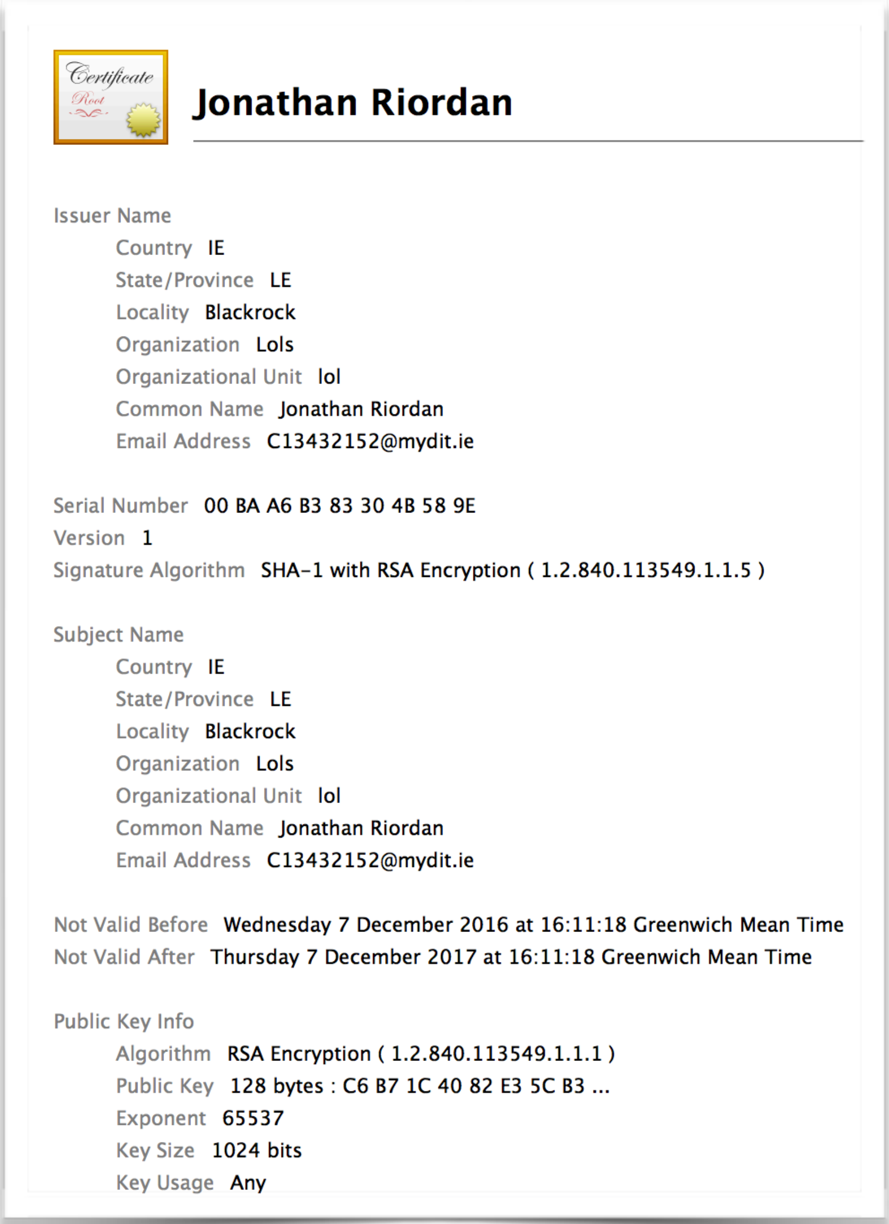
Part 2.



```
jonathan:Lab 9 Jonathan$ openssl req -new -key rsa -out rsaserver.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IE
State or Province Name (full name) [Some-State]:LE
Locality Name (eg, city) []:Blackrock
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Lols
Organizational Unit Name (eg, section) []:lol
Common Name (e.g. server FQDN or YOUR name) []:Jonathan Riordan
Email Address []:C13432152@mydit.ie

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
```



```
jonathan:Lab 9 Jonathan$ openssl x509 -req -days 365 -in rsaserver.csr -signkey rsa  -out
 rsaserver.crt
Signature ok
subject=/C=IE/ST=LE/L=Blackrock/O=Lols/OU=lol/CN=Jonathan Riordan/emailAddress=C13432152@
mydit.ie
Getting Private key
```



Jonathan Riordan

Issuer Name
Country IE
State/Province LE
Locality Blackrock
Organization Lols
Organizational Unit lol
Common Name Jonathan Riordan
Email Address C13432152@mydit.ie

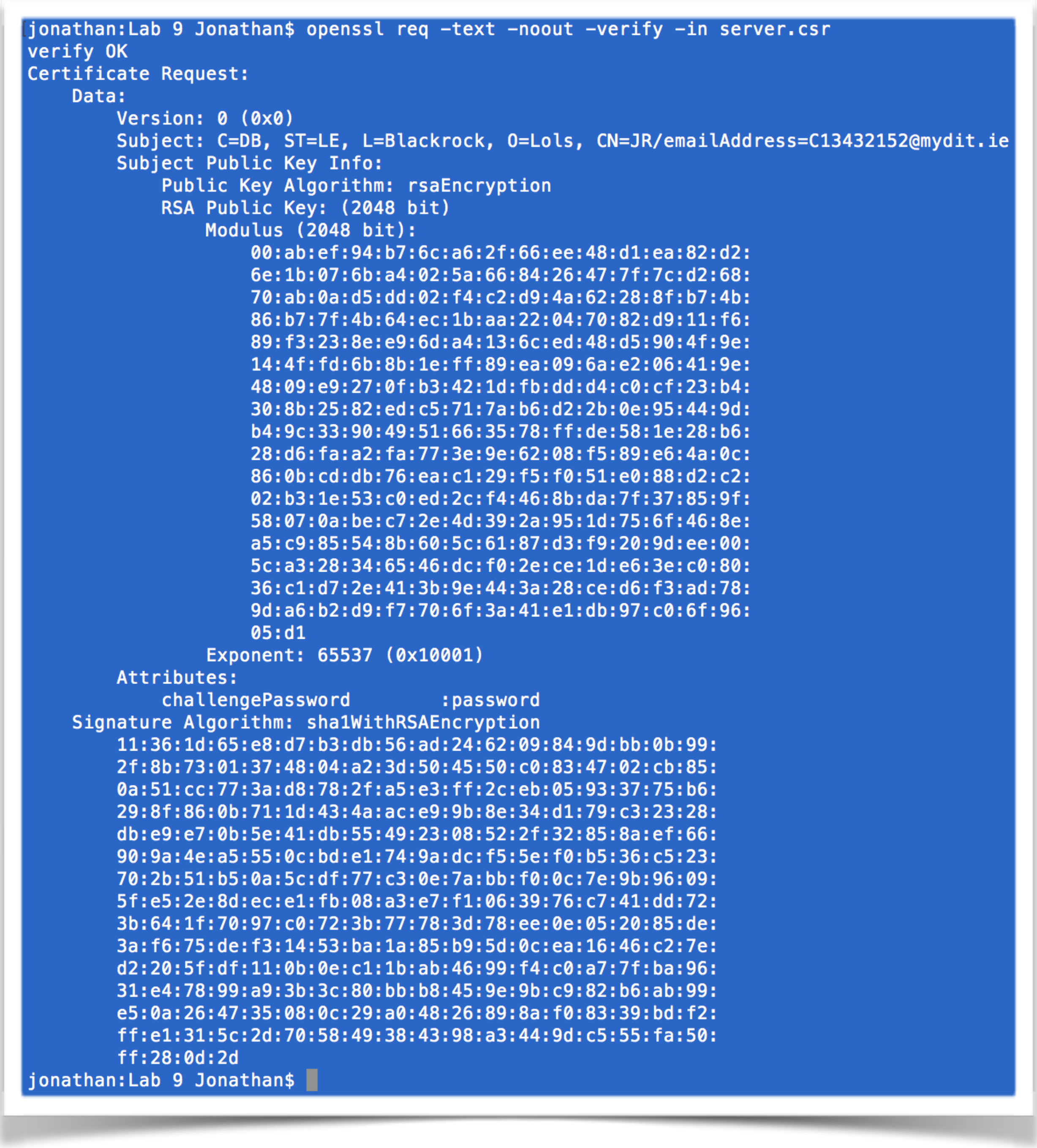
Serial Number 00 BA A6 B3 83 30 4B 58 9E
Version 1
Signature Algorithm SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)

Subject Name
Country IE
State/Province LE
Locality Blackrock
Organization Lols
Organizational Unit lol
Common Name Jonathan Riordan
Email Address C13432152@mydit.ie

Not Valid Before Wednesday 7 December 2016 at 16:11:18 Greenwich Mean Time
Not Valid After Thursday 7 December 2017 at 16:11:18 Greenwich Mean Time

Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Public Key 128 bytes : C6 B7 1C 40 82 E3 5C B3 ...
Exponent 65537
Key Size 1024 bits
Key Usage Any

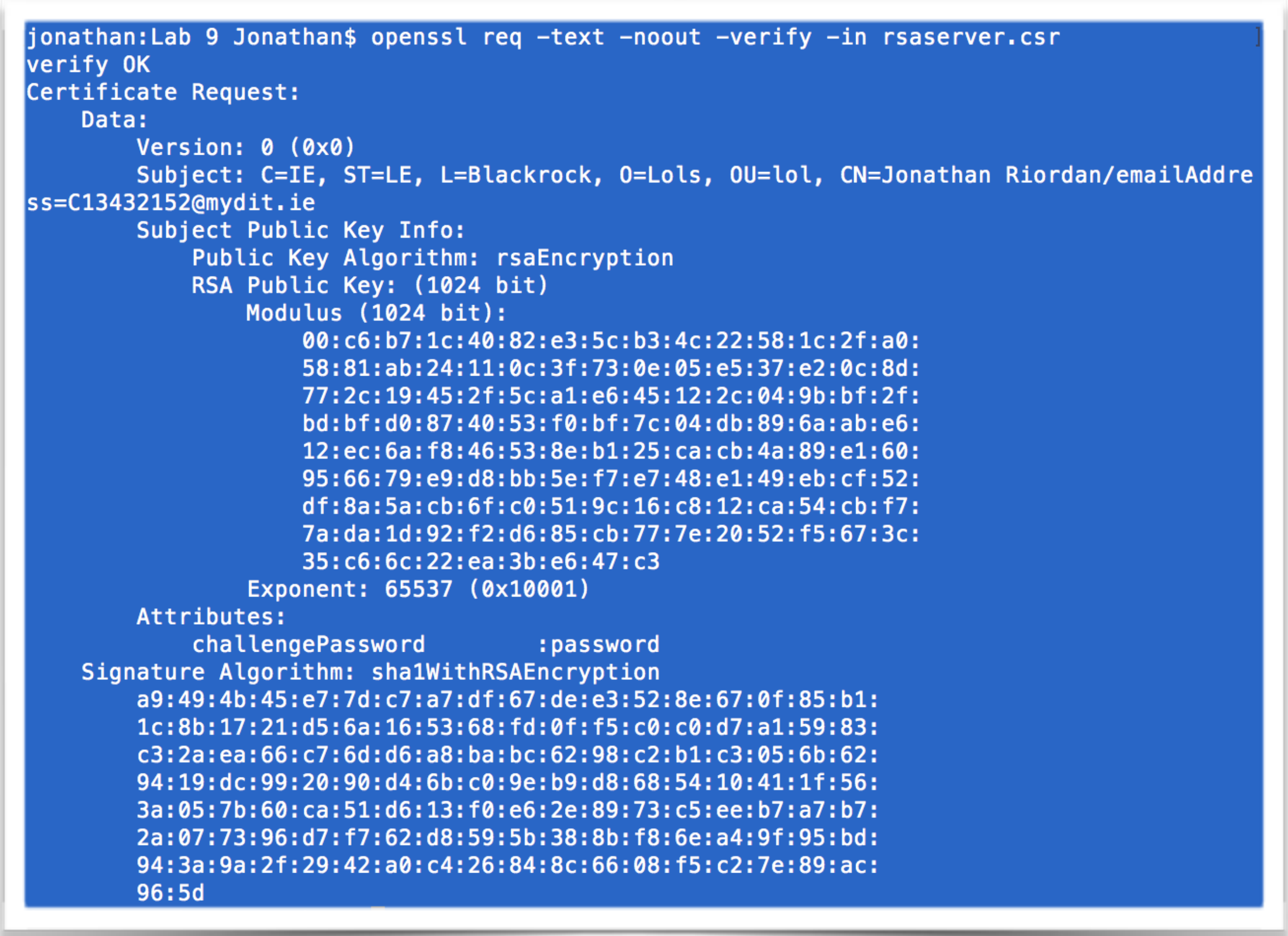
Question 3.
Validate certificates.
The first certificate. We check to verify the server.csr. The verification comes back as “Ok”. The key is 2048 bit and the algorithm is RSA.



```
jonathan:Lab 9 Jonathan$ openssl req -text -noout -verify -in server.csr
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=DB, ST=LE, L=Blackrock, O=Lols, CN=JR/emailAddress=C13432152@mydit.ie
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:ab:ef:94:b7:6c:a6:2f:66:ee:48:d1:ea:82:d2:
                    6e:1b:07:6b:a4:02:5a:66:84:26:47:7f:7c:d2:68:
                    70:ab:0a:d5:dd:02:f4:c2:d9:4a:62:28:8f:b7:4b:
                    86:b7:7f:4b:64:ec:1b:aa:22:04:70:82:d9:11:f6:
                    89:f3:23:8e:e9:6d:a4:13:6c:ed:48:d5:90:4f:9e:
                    14:4f:fd:6b:8b:1e:ff:89:ea:09:6a:e2:06:41:9e:
                    48:09:e9:27:0f:b3:42:1d:fb:dd:d4:c0:cf:23:b4:
                    30:8b:25:82:ed:c5:71:7a:b6:d2:2b:0e:95:44:9d:
                    b4:9c:33:90:49:51:66:35:78:ff:de:58:1e:28:b6:
                    28:d6:fa:a2:fa:77:3e:9e:62:08:f5:89:e6:4a:0c:
                    86:0b:cd:db:76:ea:c1:29:f5:f0:51:e0:88:d2:c2:
                    02:b3:1e:53:c0:ed:2c:f4:46:8b:da:7f:37:85:9f:
                    58:07:0a:be:c7:2e:4d:39:2a:95:1d:75:6f:46:8e:
                    a5:c9:85:54:8b:60:5c:61:87:d3:f9:20:9d:ee:00:
                    5c:a3:28:34:65:46:dc:f0:2e:ce:1d:e6:3e:c0:80:
                    36:c1:d7:2e:41:3b:9e:44:3a:28:ce:d6:f3:ad:78:
                    9d:a6:b2:d9:f7:70:6f:3a:41:e1:db:97:c0:6f:96:
                    05:d1
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :password
    Signature Algorithm: sha1WithRSAEncryption
        11:36:1d:65:e8:d7:b3:db:56:ad:24:62:09:84:9d:bb:0b:99:
        2f:8b:73:01:37:48:04:a2:3d:50:45:50:c0:83:47:02:cb:85:
        0a:51:cc:77:3a:d8:78:2f:a5:e3:ff:2c:eb:05:93:37:75:b6:
        29:8f:86:0b:71:1d:43:4a:ac:e9:9b:8e:34:d1:79:c3:23:28:
        db:e9:e7:0b:5e:41:db:55:49:23:08:52:2f:32:85:8a:ef:66:
        90:9a:4e:a5:55:0c:bd:e1:74:9a:dc:f5:5e:f0:b5:36:c5:23:
        70:2b:51:b5:0a:5c:df:77:c3:0e:7a:bb:f0:0c:7e:9b:96:09:
        5f:e5:2e:8d:ec:e1:fb:08:a3:e7:f1:06:39:76:c7:41:dd:72:
        3b:64:1f:70:97:c0:72:3b:77:78:3d:78:ee:0e:05:20:85:de:
        3a:f6:75:de:f3:14:53:ba:1a:85:b9:5d:0c:ea:16:46:c2:7e:
        d2:20:5f:df:11:0b:0e:c1:1b:ab:46:99:f4:c0:a7:7f:ba:96:
        31:e4:78:99:a9:3b:3c:80:bb:b8:45:9e:9b:c9:82:b6:ab:99:
        e5:0a:26:47:35:08:0c:29:a0:48:26:89:8a:f0:83:39:bd:f2:
        ff:e1:31:5c:2d:70:58:49:38:43:98:a3:44:9d:c5:55:fa:50:
        ff:28:0d:2d
jonathan:Lab 9 Jonathan$ 
```

The second certificate.
Validating for the rsaserver.csr. The validation comes back as ‘Ok”. The key is 1024 bit and the algorithm used is RSA.

Question 4.

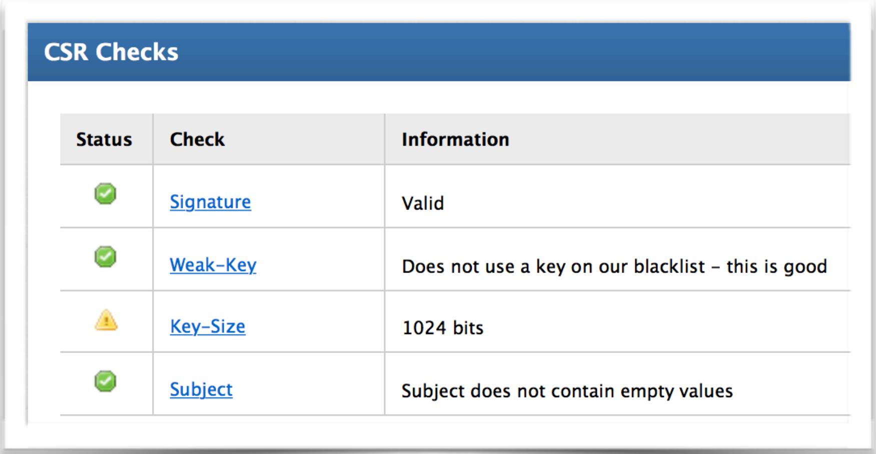


```
jonathan:Lab 9 Jonathan$ openssl req -text -noout -verify -in rsaserver.csr
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=IE, ST=LE, L=Blackrock, O=Lols, OU=lol, CN=Jonathan Riordan/emailAddre
ss=C13432152@mydit.ie
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c6:b7:1c:40:82:e3:5c:b3:4c:22:58:1c:2f:a0:
                    58:81:ab:24:11:0c:3f:73:0e:05:e5:37:e2:0c:8d:
                    77:2c:19:45:2f:5c:a1:e6:45:12:2c:04:9b:bf:2f:
                    bd:bf:d0:87:40:53:f0:bf:7c:04:db:89:6a:ab:e6:
                    12:ec:6a:f8:46:53:8e:b1:25:ca:cb:4a:89:e1:60:
                    95:66:79:e9:d8:bb:5e:f7:e7:48:e1:49:eb:cf:52:
                    df:8a:5a:cb:6f:c0:51:9c:16:c8:12:ca:54:cb:f7:
                    7a:da:1d:92:f2:d6:85:cb:77:7e:20:52:f5:67:3c:
                    35:c6:6c:22:ea:3b:e6:47:c3
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :password
    Signature Algorithm: sha1WithRSAEncryption
        a9:49:4b:45:e7:7d:c7:a7:df:67:de:e3:52:8e:67:0f:85:b1:
        1c:8b:17:21:d5:6a:16:53:68:fd:0f:f5:c0:c0:d7:a1:59:83:
        c3:2a:ea:66:c7:6d:d6:a8:ba:bc:62:98:c2:b1:c3:05:6b:62:
        94:19:dc:99:20:90:d4:6b:c0:9e:b9:d8:68:54:10:41:1f:56:
        3a:05:7b:60:ca:51:d6:13:f0:e6:2e:89:73:c5:ee:b7:a7:b7:
        2a:07:73:96:d7:f7:62:d8:59:5b:38:8b:f8:6e:a4:9f:95:bd:
        94:3a:9a:2f:29:42:a0:c4:26:84:8c:66:08:f5:c2:7e:89:ac:
        96:5d
```

Question 4.

I used the website https://certlogik.com/decoder/ to check the strength and to get the configuration of my certificates.

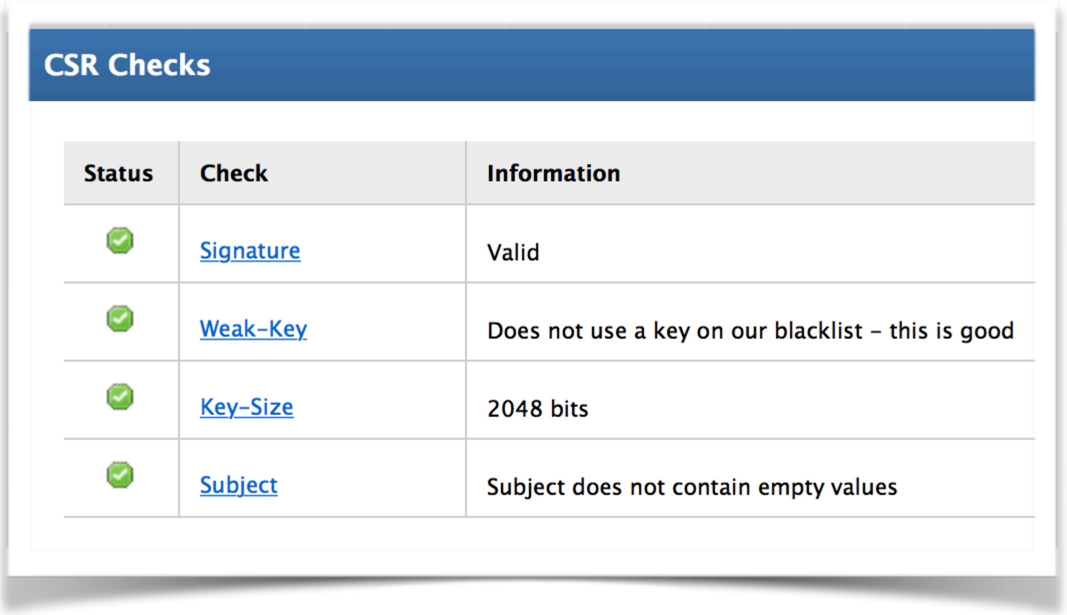
Testing the strength of my rsaserver certificate, the results are as follows. The key size provides a warning as the key is not big enough. The signature is valid. other information I can receive from this service is the configuration of the certificate, the md5 checksum and the SHA-1.



CSR Checks

Status	Check	Information
✓	Signature	Valid
✓	Weak-Key	Does not use a key on our blacklist - this is good
⚠	Key-Size	1024 bits
✓	Subject	Subject does not contain empty values

I also used this website to check the strength and configuration of the other certificate I created. As above, I can retrieve information about the key size, the md5 checksum and SHA-1. The results are as follows for my server.csr certificate.



CSR Checks

Status	Check	Information
✓	Signature	Valid
✓	Weak-Key	Does not use a key on our blacklist - this is good
✓	Key-Size	2048 bits
✓	Subject	Subject does not contain empty values