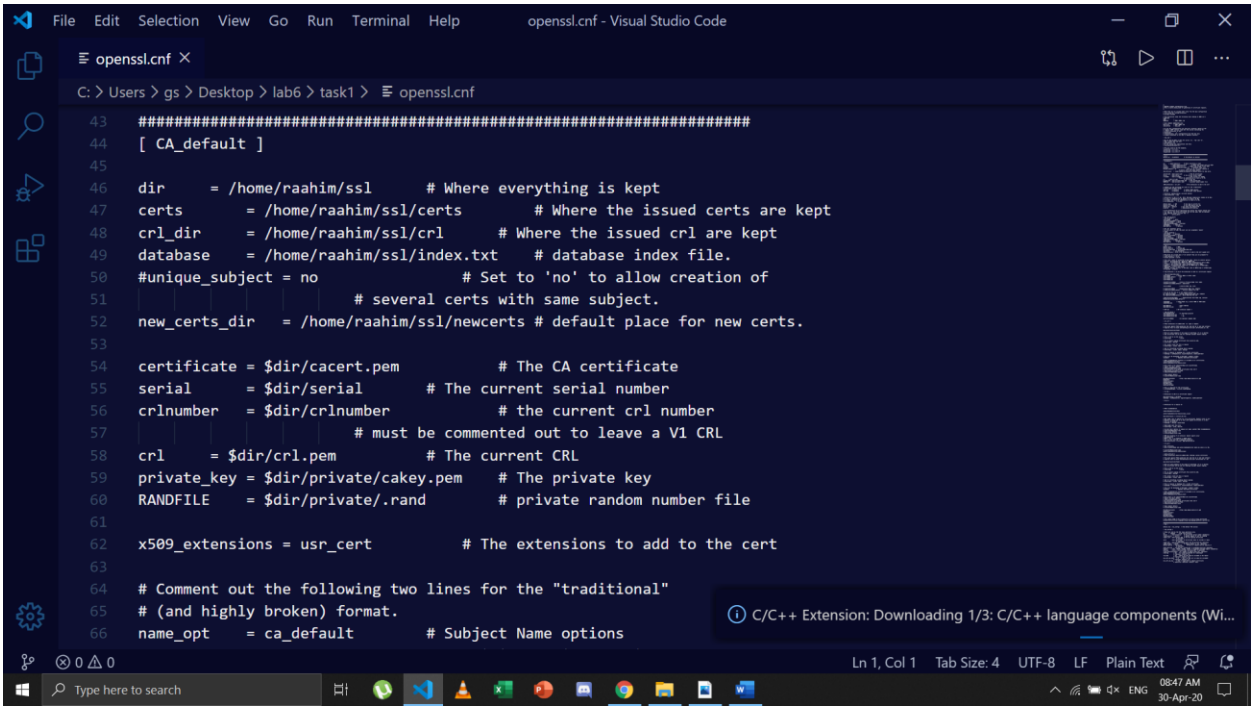
Network Security

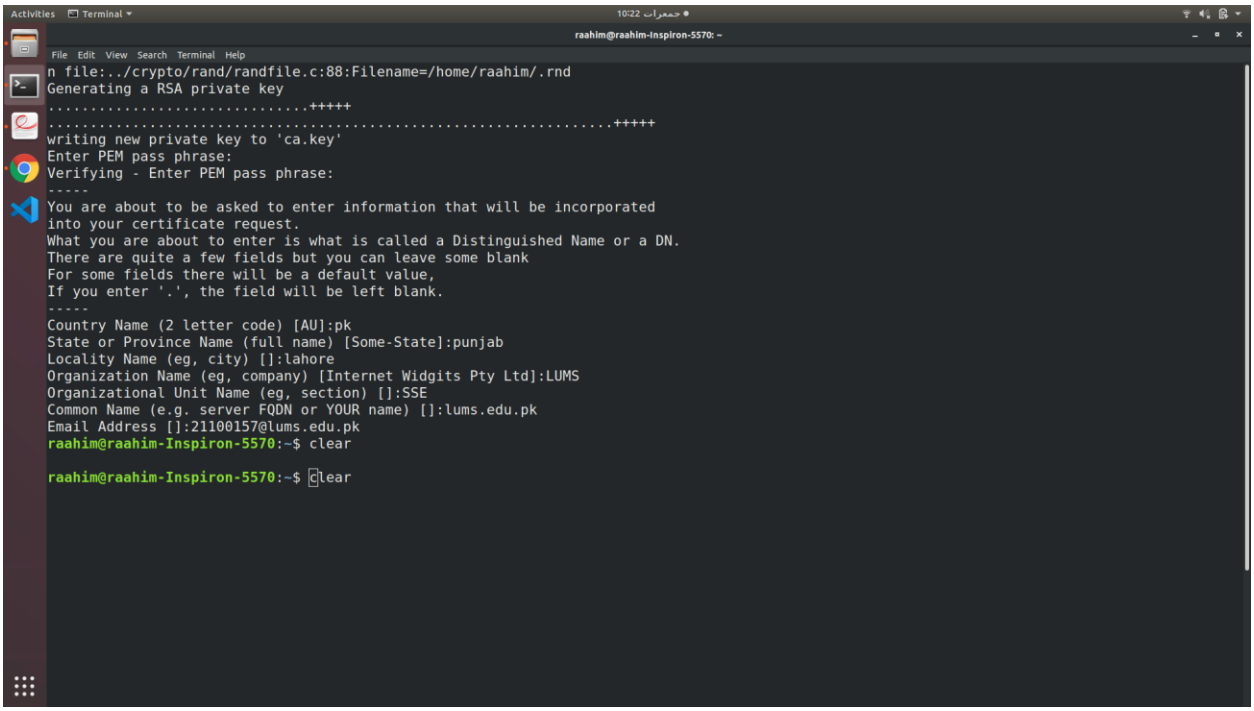
Cryptography Lab (Public-Key Cryptography and PKI)

Muhammad Raahim Khan

21100157

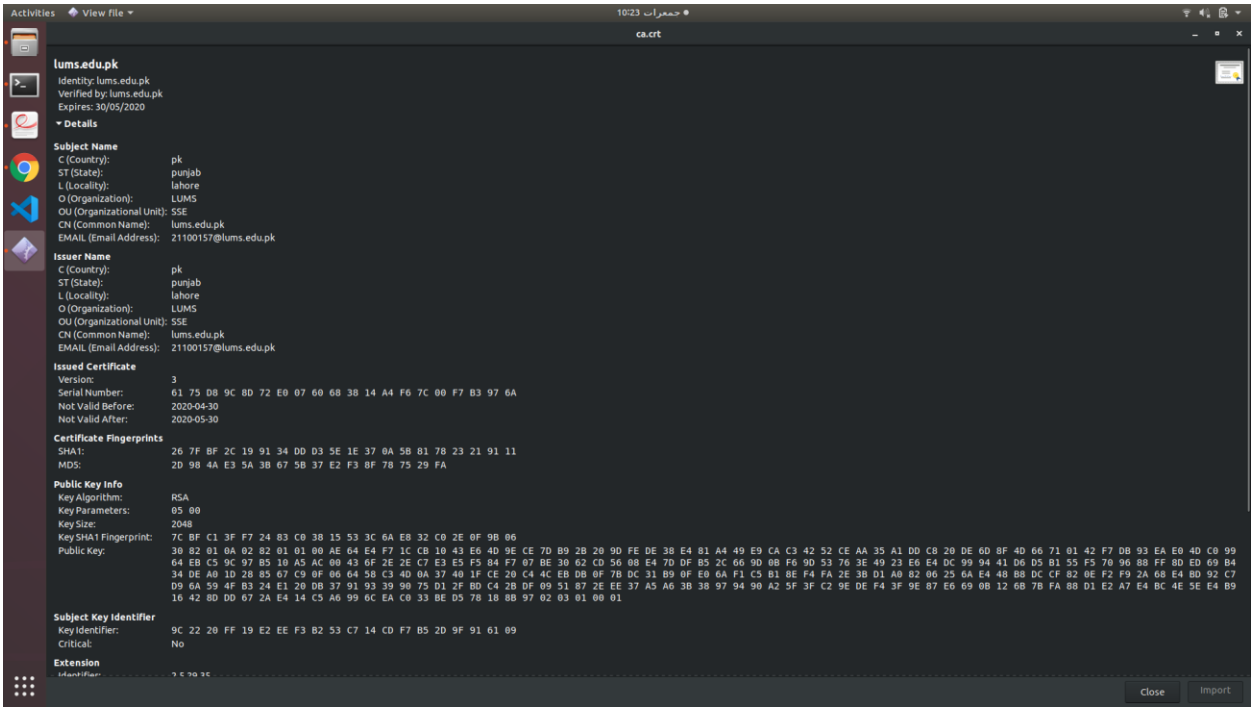
➢ Task 1:

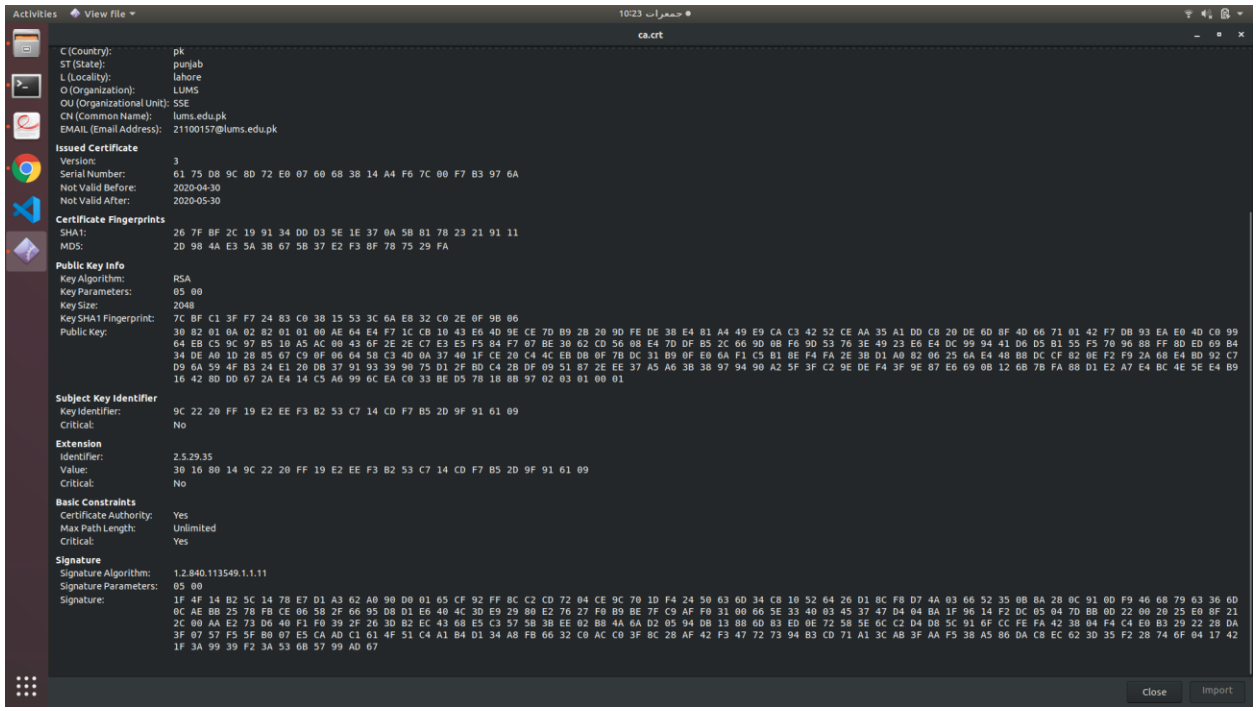



Activities
Terminal
raahim@raahim-Inspiron-5570: ~
File Edit View Search Terminal Help
n file:../crypto/rand/randfile.c:88:Filename=/home/raahim/.rnd
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

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.

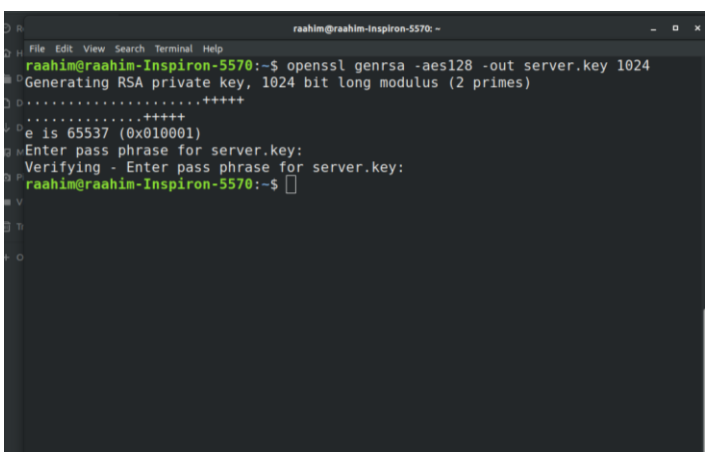
Country Name (2 letter code) [AU]:pk
State or Province Name (full name) [Some-State]:punjab
Locality Name (eg, city) []:lahore
Organization Name (eg, company) [Internet Widgits Pty Ltd]:LUMS
Organizational Unit Name (eg, section) []:SSE
Common Name (e.g. server FQDN or YOUR name) []:lums.edu.pk
Email Address []:21100157@lums.edu.pk
raahim@raahim-Inspiron-5570:~$ clear
raahim@raahim-Inspiron-5570:~$ clear


Activities
View file
ca.crt
lums.edu.pk
Identity: lums.edu.pk
Verified by: lums.edu.pk
Expires: 30/05/2020
Details
Subject Name
C (Country): pk
ST (State): punjab
L (Locality): lahore
O (Organization): LUMS
OU (Organizational Unit): SSE
CN (Common Name): lums.edu.pk
EMAIL (Email Address): 21100157@lums.edu.pk
Issuer Name
C (Country): pk
ST (State): punjab
L (Locality): lahore
O (Organization): LUMS
OU (Organizational Unit): SSE
CN (Common Name): lums.edu.pk
EMAIL (Email Address): 21100157@lums.edu.pk
Issued Certificate
Version: 3
Serial Number: 61 75 D8 9C 8D 72 E0 07 60 68 38 14 A4 F6 7C 00 F7 B3 97 6A
Not Valid Before: 2020-04-30
Not Valid After: 2020-05-30
Certificate Fingerprints
SHA1: 26 7F BF 2C 19 91 34 DD D3 5E 1E 37 0A 5B 81 78 23 21 91 11
MD5: 2D 98 4A E3 5A 3B 67 5B 37 E2 F3 8F 78 75 29 FA
Public Key Info
Key Algorithm: RSA
Key Parameters: 05 00
Key Size: 2048
Key SHA1 Fingerprint: 7C BF C1 3F F7 24 83 C0 3B 15 53 3C 6A E8 32 C0 2E 0F 9B 06
Public Key: 30 82 01 0A 02 82 01 01 00 AE 64 E4 F7 1C CB 10 43 E6 4D 9E CE 7D B9 2B 20 9D FE DE 38 E4 81 A4 49 E9 CA C3 42 52 CE AA 35 A1 DD C8 20 DE 6D 8F 4D 66 71 01 42 F7 DB 93 EA E0 4D C0 99
64 EB C5 9C 97 B5 10 A5 AC 00 43 6F 2E 2E C7 E3 E5 F5 84 F7 07 BE 30 62 CD 56 08 E4 7D DF B5 2C 66 9D 0B F6 9D 53 76 3E 49 23 E6 E4 DC 99 94 41 D6 D5 B1 55 F5 70 96 88 FF 8D ED 69 B4
34 DE A0 1D 28 85 67 C9 0F 06 64 58 C3 4D 0A 37 40 1F CE 20 C4 4C EB DB 0F 7B DC 31 B9 0F E0 6A F1 C5 B1 8E F4 FA 2E 3B D1 A0 82 06 25 6A E4 48 B8 DC CF 82 0E F2 F9 2A 68 E4 BD 92 C7
D9 6A 59 4F B3 24 E1 20 DB 37 91 93 39 90 75 D1 2F BD C4 2B DF 09 51 87 2E EE 37 A5 A6 3B 38 97 94 90 A2 5F 3F C2 9E DE F4 3F 9E 87 E6 69 0B 12 6B 7B FA 88 D1 E2 A7 E4 BC 4E 5E E4 B9
16 42 8D DD 67 2A E4 14 C5 A6 99 6C EA C0 33 BE D5 78 18 8B 97 02 03 01 00 01
Subject Key Identifier
Key Identifier: 9C 22 20 FF 19 E2 EE F3 B2 53 C7 14 CD F7 B5 2D 9F 91 61 09
Critical: No
Extension
Close
Import

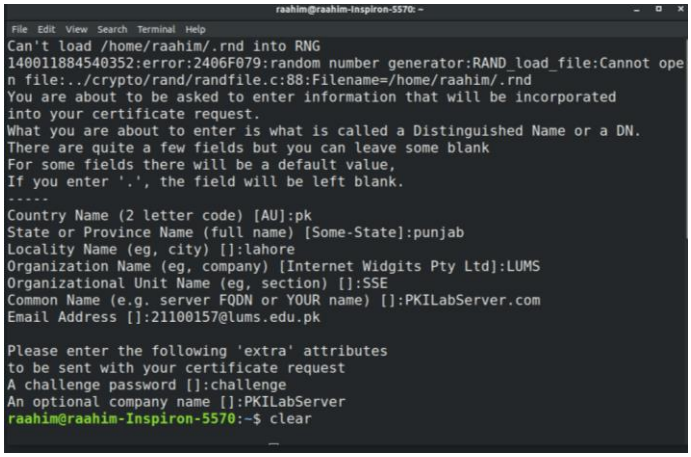

Activities
View file
10:23 جمعرات
ca.crt
C (Country): pk
ST (State): punjab
L (Locality): lahore
O (Organization): LUMS
OU (Organizational Unit): SSE
CN (Common Name): lums.edu.pk
EMAIL (Email Address): 21100157@lums.edu.pk
Issued Certificate
Version: 3
Serial Number: 61 75 D8 9C 8D 72 E0 07 60 68 38 14 A4 F6 7C 00 F7 B3 97 6A
Not Valid Before: 2020-04-30
Not Valid After: 2020-05-30
Certificate Fingerprints
SHA1: 26 7F BF 2C 19 91 34 DD D3 5E 1E 37 0A 5B 01 78 23 21 91 11
MD5: 2D 98 4A E3 5A 3B 67 5B 37 E2 F3 8F 78 75 29 FA
Public Key Info
Key Algorithm: RSA
Key Parameters: 05 00
Key Size: 2048
Key SHA1 Fingerprint: 7C BF C1 3F F7 24 83 C0 38 15 53 3C 6A E8 32 C0 2E 0F 9B 06
Public Key: 30 82 01 0A 02 82 01 01 00 AE 64 E4 F7 1C CB 10 43 E6 4D 9E CE 7D B9 2B 20 9D FE DE 3B E4 81 A4 49 E9 CA C3 42 52 CE AA 35 A1 DD C8 20 DE 6D 8F 4D 66 71 01 42 F7 DB 93 EA E0 4D C0 99
64 EB C5 9C 97 85 10 A5 AC 00 43 6F 2E C7 E3 E5 F5 84 F7 07 BE 30 62 CD 56 08 E4 7D DF B5 2C 66 9D 8B F6 9D 53 76 3E 49 23 E6 E4 DC 99 94 41 D6 D5 B1 55 F5 70 96 88 FF 8D ED 69 B4
34 DE A0 1D 28 85 67 C9 0F 06 64 58 C3 4D 0A 37 40 1F CE 20 C4 4C EB DB 0F 7B DC 31 B9 0F E0 6A F1 C5 B1 8E F4 FA 2E 3B D1 A0 82 06 25 6A E4 48 B8 DC CF 82 0E F2 F9 2A 68 E4 BD 92 C7
D9 6A 59 4F B3 24 E1 20 DB 37 91 93 39 90 75 D1 2F BD C4 2B DF 09 51 87 2E EE 37 A5 A6 3B 38 97 94 90 A2 5F 3F C2 9E DE F4 3F 9E 87 E6 69 0B 12 6B 7B FA 88 D1 E2 A7 E4 BC 4E 5E E4 B9
16 42 8D DD 67 2A E4 14 C5 A6 99 6C EA C0 33 BE D5 78 18 8B 97 02 03 01 00 01
Subject Key Identifier
Key Identifier: 9C 22 20 FF 19 E2 EE F3 B2 53 C7 14 CD F7 B5 2D 9F 91 61 09
Critical: No
Extension
Identifier: 2.5.29.35
Value: 30 16 80 14 9C 22 20 FF 19 E2 EE F3 B2 53 C7 14 CD F7 B5 2D 9F 91 61 09
Critical: No
Basic Constraints
Certificate Authority: Yes
Max Path Length: Unlimited
Critical: Yes
Signature
Signature Algorithm: 1.2.840.113549.1.1.11
Signature Parameters: 05 00
Signature: 1F 4F 14 B2 5C 14 78 E7 D1 A3 62 A0 90 D0 01 65 CF 92 FF 8C C2 CD 72 04 CE 9C 70 1D F4 24 50 63 6D 34 C8 10 52 64 26 D1 8C F8 D7 4A 03 66 52 35 0B 8A 28 0C 91 0D F9 46 68 79 63 36 6D
0C AE BB 25 78 FB CE 06 58 2F 66 95 D8 D1 E6 40 4C 3D E9 29 80 E2 76 27 F0 B9 BE 7F C9 AF F0 31 00 66 5E 33 40 03 45 37 47 D4 04 BA 1F 96 14 F2 DC 05 04 7D BB 0D 22 00 20 25 E0 8F 21
2C 00 AA E2 73 D6 40 F1 F0 39 2F 26 3D B2 EC 43 6B E5 C3 57 5B 3B EE 02 B8 4A 6A D2 05 94 DB 13 8B 6D B3 ED 0E 72 58 5E 6C C2 D4 D8 5C 91 6F CC FE FA 42 38 04 F4 C4 E0 B3 29 22 28 DA
3F 07 57 F5 5F B0 07 E5 CA AD C1 61 4F 51 C4 A1 B4 D1 34 A8 FB 66 32 C0 AC C0 3F 8C 28 AF 42 F3 47 72 73 94 B3 CD 71 A1 3C AB 3F AA F5 38 A5 86 DA C8 EC 62 3D 35 F2 28 74 6F 04 17 42
1F 3A 99 39 F2 3A 53 6B 57 99 AD 67
Close
Import

➢ Task 2:

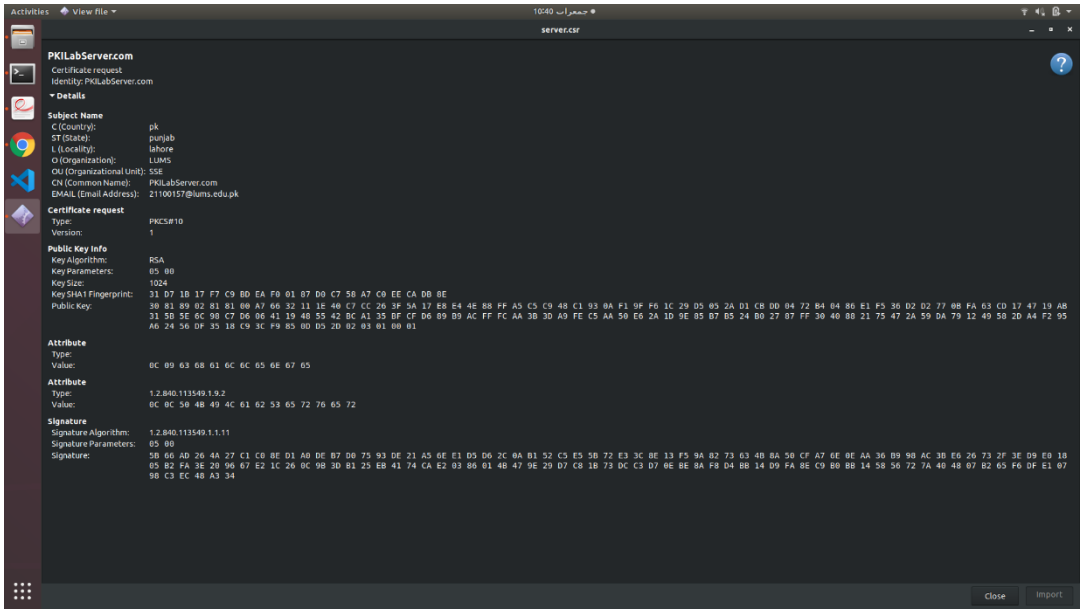
- **Step 1:**


raahim@raahim-inspiron-5570: ~
File Edit View Search Terminal Help
raahim@raahim-Inspiron-5570:~$ openssl genrsa -aes128 -out server.key 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
raahim@raahim-Inspiron-5570:~$

- **Step 2:**

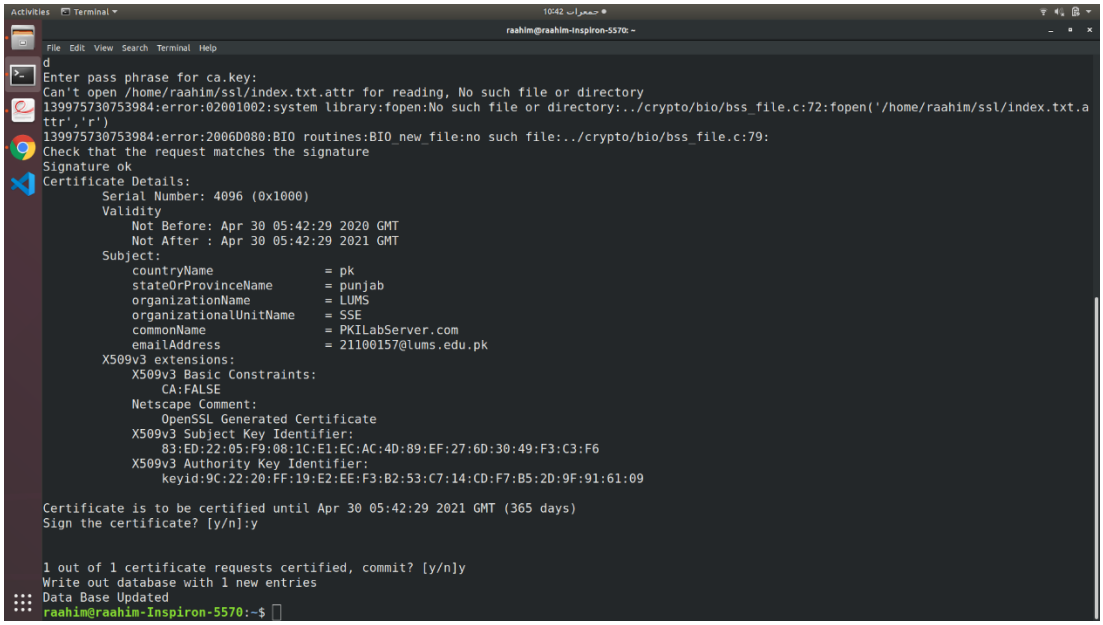


```
raahim@raahim-Inspiron-5570: ~
File  Edit  View  Search  Terminal  Help
Can't load /home/raahim/.rnd into RNG
140011884540352:error:2406F079:random number generator:RAND_load_file:Cannot ope
n file:../crypto/rand/randfile.c:88:Filename=/home/raahim/.rnd
You are about to be asked to enter information 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]:pk
State or Province Name (full name) [Some-State]:punjab
Locality Name (eg, city) []:lahore
Organization Name (eg, company) [Internet Widgits Pty Ltd]:LUMS
Organizational Unit Name (eg, section) []:SSE
Common Name (e.g. server FQDN or YOUR name) []:PKILabServer.com
Email Address []:21100157@lums.edu.pk

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:challenge
An optional company name []:PKILabServer
raahim@raahim-Inspiron-5570:~$ clear
```

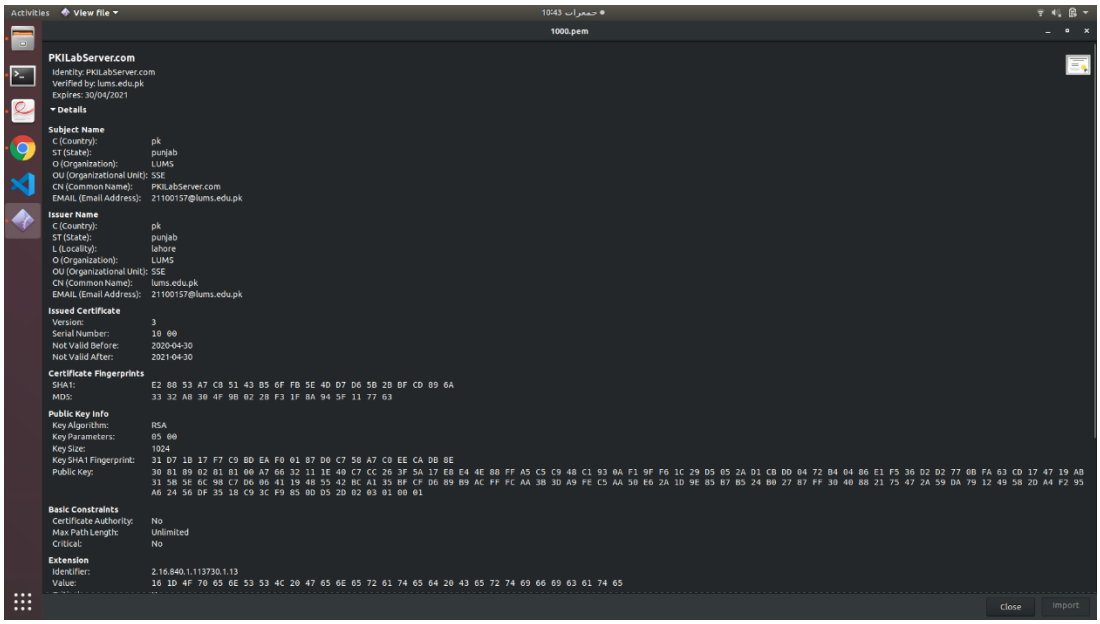


```
Activities   View file   جمعرات 10:40
server.csr

PKILabServer.com
Certificate request
Identity: PKILabServer.com
Details

Subject Name
C (Country):                pk
ST (State):                 punjab
L (Locality):               lahore
O (Organization):           LUMS
OU (Organizational Unit):   SSE
CN (Common Name):           PKILabServer.com
EMAIL (Email Address):      21100157@lums.edu.pk

Certificate request
Type:                       PKCS#10
Version:                    1

Public Key Info
Key Algorithm:              RSA
Key Parameters:             05 00
Key Size:                   1024
Key SHA1 Fingerprint:       31 D7 1B 17 F7 C9 BD EA F0 01 87 D0 C7 58 A7 C0 EE CA DB 8E
Public Key:                 30 81 89 02 81 81 00 A7 66 32 11 1E 40 C7 CC 26 3F 5A 17 E8 E4 4E 88 FF A5 C5 C9 48 C1 93 0A F1 9F F6 1C 29 D5 05 2A D1 CB DD 04 72 B4 04 86 E1 F5 36 D2 D2 77 0B FA 63 CD 17 47 19 AB
                            31 5B 5E 6C 98 C7 D6 06 41 19 40 55 42 BC A1 35 BF CF D6 89 B9 AC FF FC AA 3B 3D A9 FE C5 AA 50 E6 2A 1D 9E 85 B7 D5 24 B0 27 87 FF 30 40 88 21 75 47 2A 59 DA 79 12 49 58 2D A4 F2 95
                            A6 24 56 DF 35 18 C9 3C F9 85 8D D5 2D 02 03 01 00 01

Attribute
Type:
Value:                      0C 09 63 68 61 6C 6C 65 6E 67 65

Attribute
Type:                       1.2.840.113549.1.9.2
Value:                      0C 0C 50 4B 49 4C 61 62 53 65 72 76 65 72

Signature
Signature Algorithm:        1.2.840.113549.1.1.11
Signature Parameters:       05 00
Signature:                  5B 66 AD 26 4A 27 C1 C0 8E D1 A0 DE B7 D0 75 93 DE 21 A5 6E E1 D5 D6 2C 0A B1 52 C5 E5 5B 72 E3 3C 8E 13 F5 9A 82 73 63 4B 8A 50 CF A7 6E 0E AA 36 B9 98 AC 3B E6 26 73 2F 3E D9 E0 18
                            05 82 FA 3E 20 96 67 E2 1C 26 0C 9B 3D 81 25 EB 41 74 CA E2 03 86 01 4B 47 9E 29 D7 C8 1B 73 DC C3 D7 0E 8E 8A F8 D4 BB 14 D9 FA 8E C9 B0 BB 14 58 56 72 7A 40 48 07 B2 65 F6 DF E1 07
                            9B C3 EC 4B A3 34

Close   Import
```

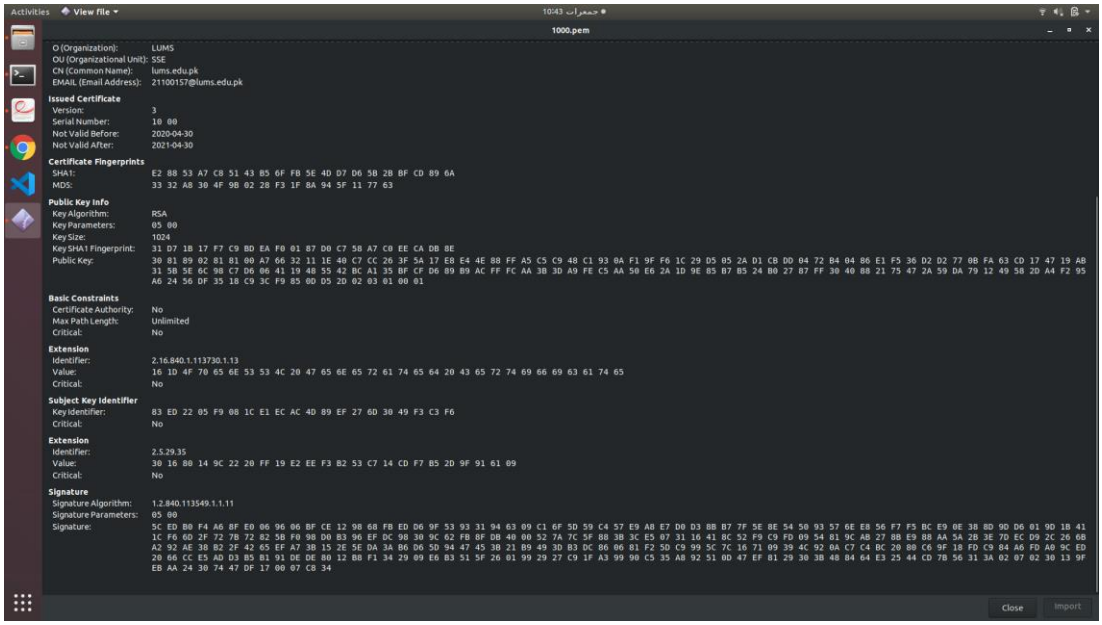
- **Step 3:**



```
d
Enter pass phrase for ca.key:
Can't open /home/raahim/ssl/index.txt.attr for reading, No such file or directory
139975730753984:error:02001002:system library:fopen:No such file or directory:../crypto/bio/bss_file.c:72:fopen('/home/raahim/ssl/index.txt.a
ttr','r')
139975730753984:error:2006D080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:79:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Apr 30 05:42:29 2020 GMT
            Not After : Apr 30 05:42:29 2021 GMT
        Subject:
            countryName               = pk
            stateOrProvinceName       = punjab
            organizationName          = LUMS
            organizationalUnitName    = SSE
            commonName                = PKILabServer.com
            emailAddress              = 21100157@lums.edu.pk
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                83:ED:22:05:F9:08:1C:E1:EC:AC:4D:89:EF:27:6D:30:49:F3:C3:F6
            X509v3 Authority Key Identifier:
                keyid:9C:22:20:FF:19:E2:EE:F3:B2:53:C7:14:CD:F7:B5:2D:9F:91:61:09

Certificate is to be certified until Apr 30 05:42:29 2021 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
raahim@raahim-Inspiron-5570:~$
```



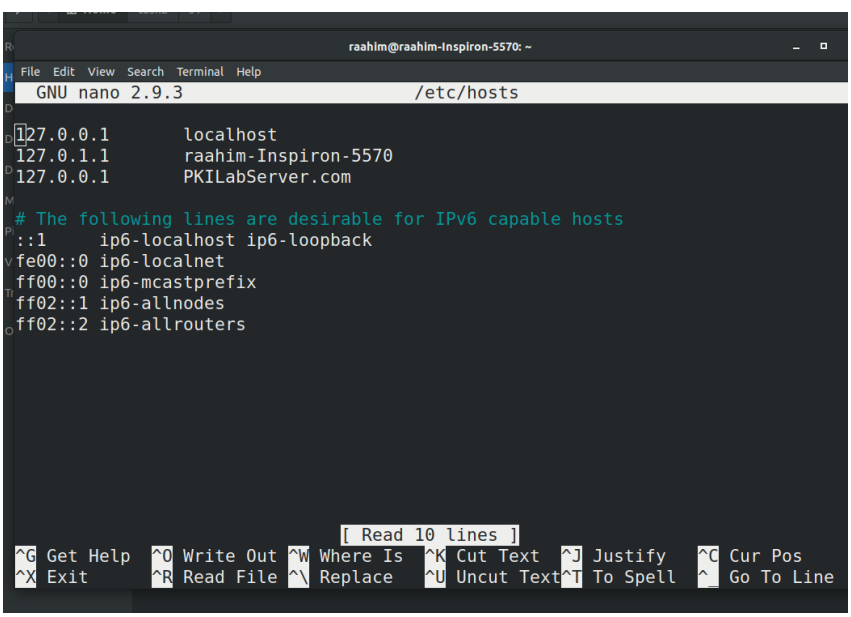
```
PKILabServer.com
Identity: PKILabServer.com
Verified by: lums.edu.pk
Expires: 30/04/2021
Details

Subject Name
  C (Country):                pk
  ST (State):                 punjab
  O (Organization):           LUMS
  OU (Organizational Unit):   SSE
  CN (Common Name):           PKILabServer.com
  EMAIL (Email Address):      21100157@lums.edu.pk
Issuer Name
  C (Country):                pk
  ST (State):                 punjab
  L (Locality):               lahore
  O (Organization):           LUMS
  OU (Organizational Unit):   SSE
  CN (Common Name):           lums.edu.pk
  EMAIL (Email Address):      21100157@lums.edu.pk
Issued Certificate
  Version:                    3
  Serial Number:              10 00
  Not Valid Before:           2020-04-30
  Not Valid After:            2021-04-30
Certificate Fingerprints
  SHA1:                       E2 88 53 A7 C8 51 43 B5 6F FB 5E 4D D7 D6 5B 2D BF CD 89 6A
  MD5:                        33 32 A8 30 4F 9B 02 28 F3 1F 8A 94 5F 11 77 63
Public Key Info
  Key Algorithm:              RSA
  Key Parameters:             05 00
  Key Size:                   1024
  Key SHA1 Fingerprint:       31 D7 1B 17 F7 C9 BD EA F0 01 87 D0 C7 58 A7 C0 EE CA DB 0E
  Public Key:                 30 81 89 02 81 81 00 A7 66 32 11 1E 40 C7 CC 26 3F 5A 17 E8 E4 4E 88 FF A5 C5 C9 48 C1 93 0A F1 9F F6 1C 29 D5 65 2A D1 CB DD 04 72 B4 04 86 E1 F5 36 D2 D2 77 0B FA 63 CD 17 47 19 AB
                              31 5B 5E 6C 98 C7 D6 06 41 19 48 55 42 BC A1 35 BF CF D6 B9 B9 AC FF FC AA 3B 3D A9 FE C5 AA 50 E6 2A 1D 9E 85 B7 B5 24 B0 27 87 FF 30 40 88 21 75 47 2A 59 DA 79 12 49 58 2D A4 F2 95
                              A6 24 56 DF 35 1B C9 3C F9 85 0D D5 2D 02 03 01 00 01

Basic Constraints
  Certificate Authority:      No
  Max Path Length:            Unlimited
  Critical:                   No
Extension
  Identifier:                 2.16.840.1.113730.1.13
  Value:                      16 1D 4F 70 65 6E 53 53 4C 20 47 65 6E 65 72 61 74 65 64 20 43 65 72 74 69 66 69 63 61 74 65
```

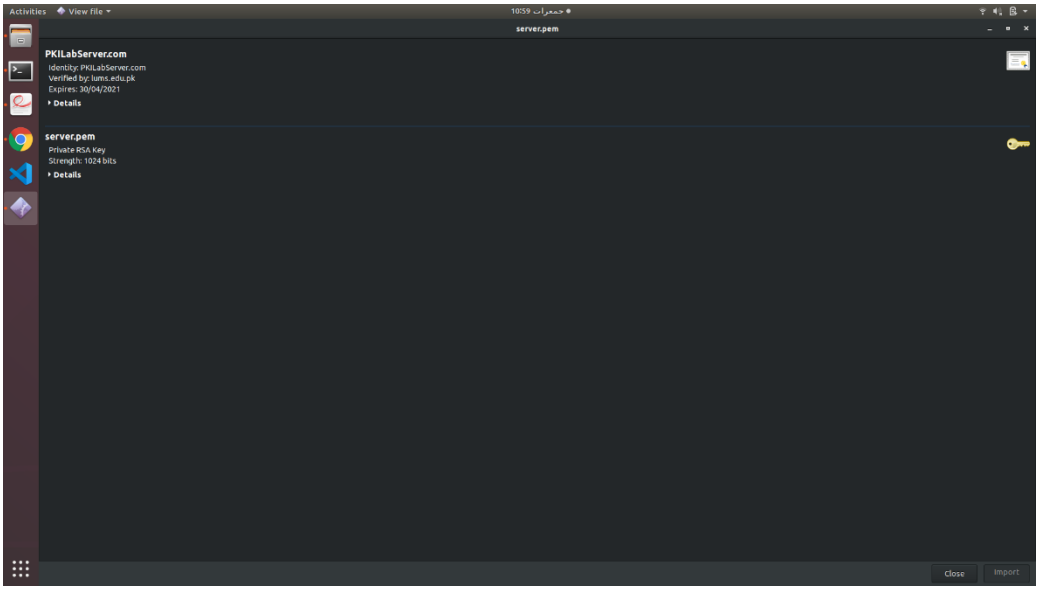


Note: 1000.pem was the file generated in the folder I specified in the openssl.conf file. Similar output was present in server.crt

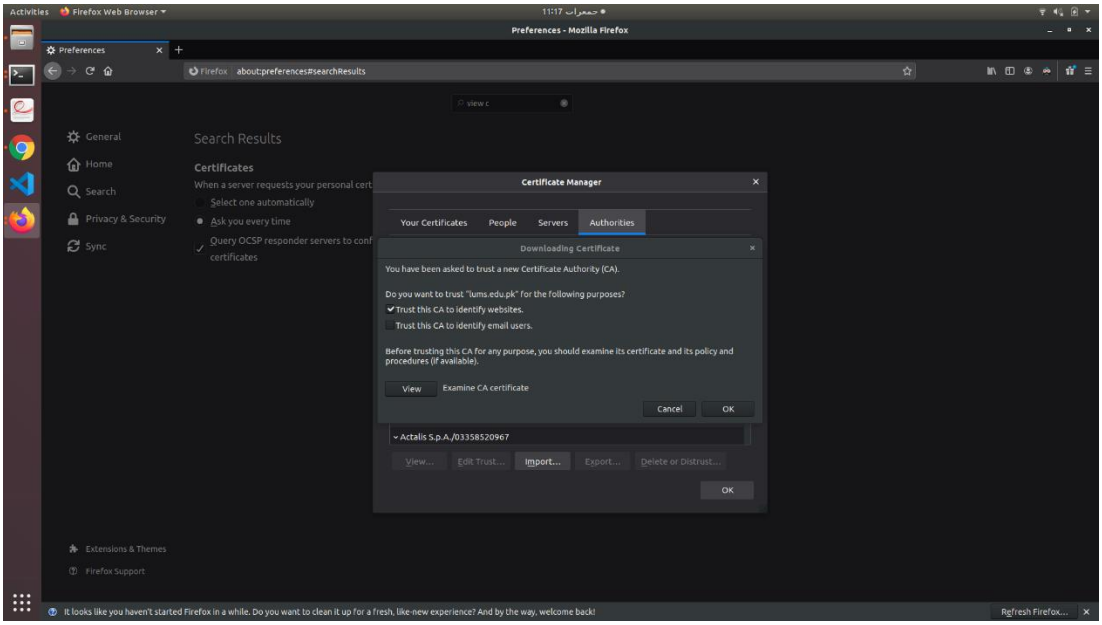
---

➢ Task 3:

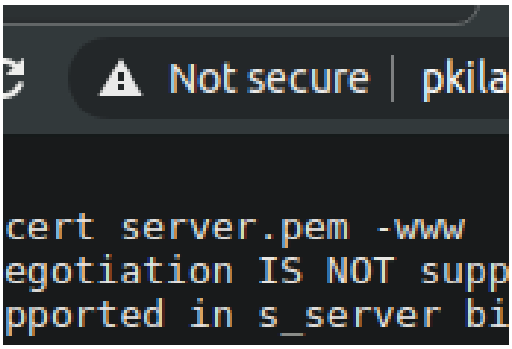



Activities
View file
10:59 جمعرات
server.pem
PKILabServer.com
Identity: PKILabServer.com
Verified by: lums.edu.pk
Expires: 30/04/2021
Details
server.pem
Private RSA Key
Strength: 1024 bits
Details
Close
Import

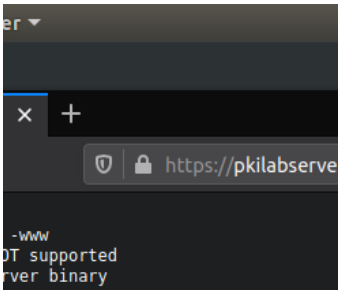

This Connection is Untrusted
You have asked Firefox to connect securely to pkilabserver.com:4433, but we can't confirm that your connection is secure.
Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.
What Should I Do?
If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.
Get me out of here!
Technical Details
I Understand the Risks



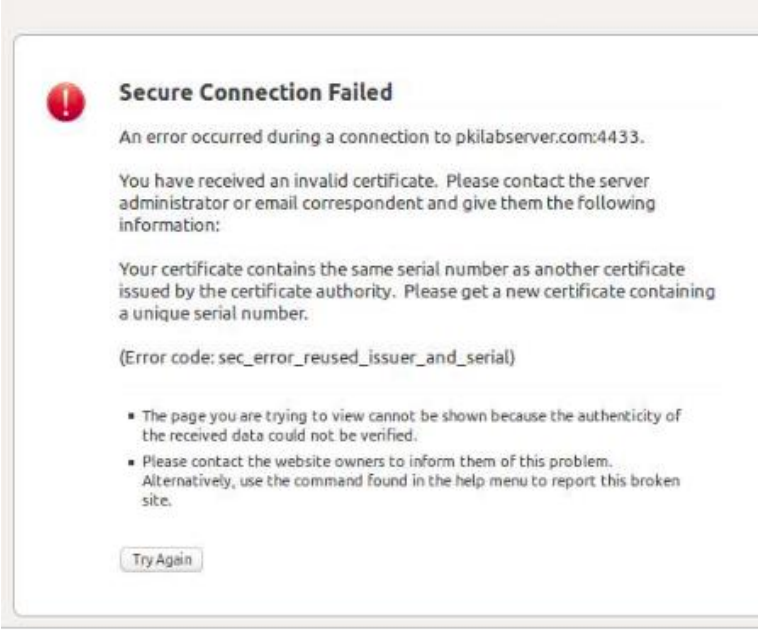
Before importing the certificate into the browser, the certificate is invalid so the browser gives a warning message concerning security issues. There is also a “Not secure” sign in the browser.



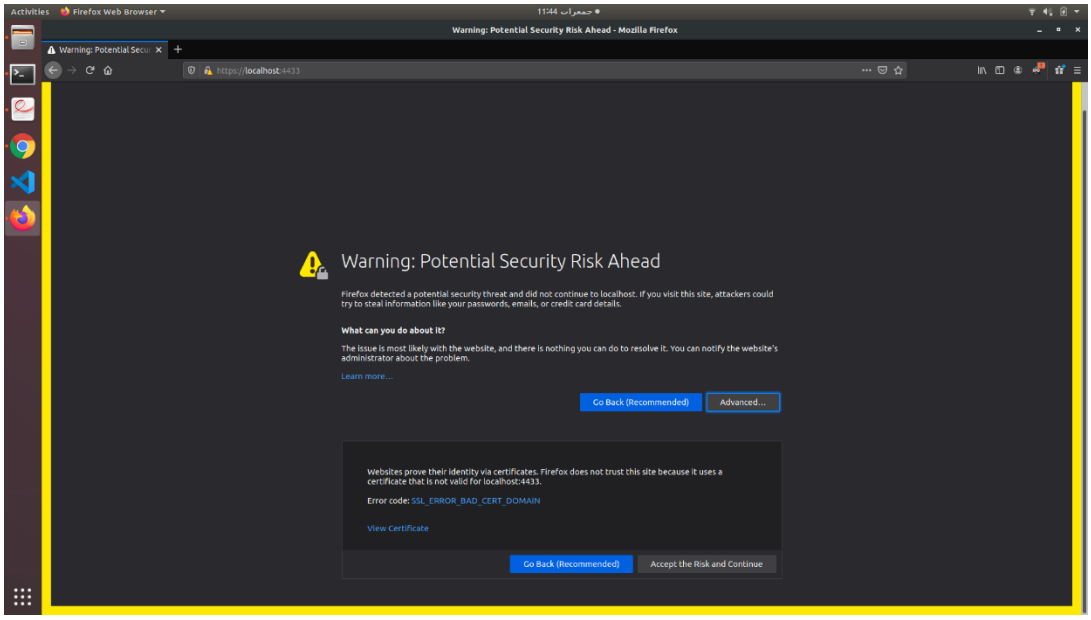
After loading ca.crt into Firefox, the warning message is disappeared. Instead of a “Not secure” sign in the browser, I now saw a lock sign next to address area. This indicates that the certificate signed for PKILabServer.com has been accepted as a valid certificate by my browser. Hence, now all certificates signed by this certificate will be trusted.



1) After modifying a single byte of server.pem, I restarted the server and reloaded the provided URL in the homework pdf file. I observed that the site would not open and it gave an error. Reason is that the certificate was modified illegally and hence, the signature did not match.

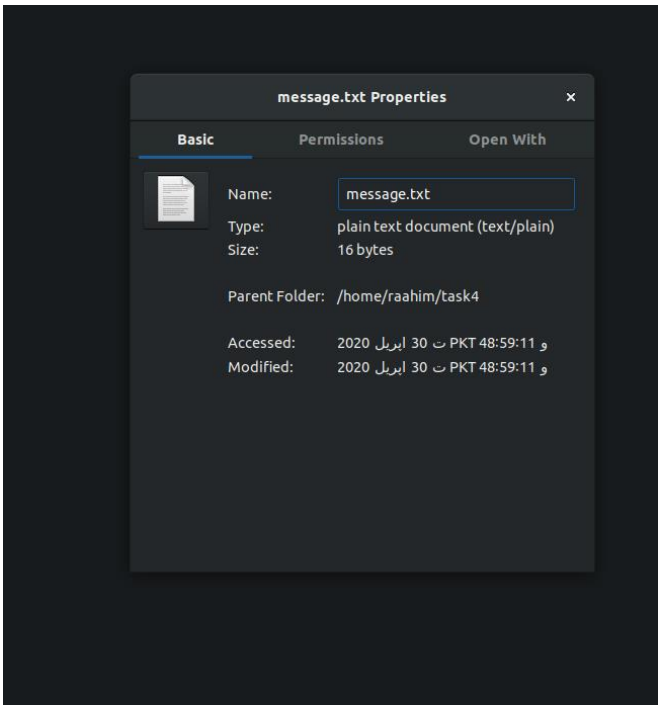


2) After redirecting my browser to https://localhost:4433, I saw a warning message about an untrusted certificate. This could be due to the reason that certificate is only valid for the domain PKILabServer.com and hence, if another domain is used to access the site, the certificate will be considered to be untrusted.



➢ **Task 4:**

First I created a file named “message.txt”. File size was of 16 bytes.



Then I generated a 1024-bit RSA public/private key pair using the command :

"**openssl genrsa -aes128 -out task4.key 1024**"

For encrypting the message using the public key, I first extracted the public key from the public/private key pair I generated. I used the following command for this purpose:

"**openssl rsa -pubout -in task4.key -out pub_key.key**"

Then I encrypted "message.txt" using the above extracted public key using the following command:

"**openssl rsautl -encrypt -in message.txt -out messageenc.txt -inkey pub_key.key -pubin**"

Finally, I decrypted "messageenc.txt" using the private key. For this purpose, I used the following command:

"**openssl rsautl -decrypt -in messageenc.txt -inkey task4.key**"

For encrypting "message.txt" using a 128-bit AES key, I used the following command:

"**openssl aes-128-cbc -in message.txt -out messageenc.enc**"

The above operations were too fast to measure. Hence, I used the "**time**" command. Both have almost similar time of around 0.009 seconds. So, in order to affectively differentiate between the two, I wrote a bash script in order to execute each of the above stated commands a 1000 number of times and then taking the average.

For RSA, I used the following bash script:

helperscript.sh

C: > Users > gs > Desktop > lab6 > task4 > helperscript.sh

```
#!/bin/bash
x=1
while [ $x -le 1000 ]
do
  openssl rsautl -encrypt -in message.txt -inkey pub_key.key -pubin -out messageenc.txt
  x=$(( $x + 1 ))
done
```

Output was:

```
raahim@raahim-Inspiron-5570: ~/task4
File  Edit  View  Search  Terminal  Help
raahim@raahim-Inspiron-5570:~/task4$ time ./helperscript.sh

real    0m4.223s
user    0m2.887s
sys     0m1.402s
raahim@raahim-Inspiron-5570:~/task4$
```

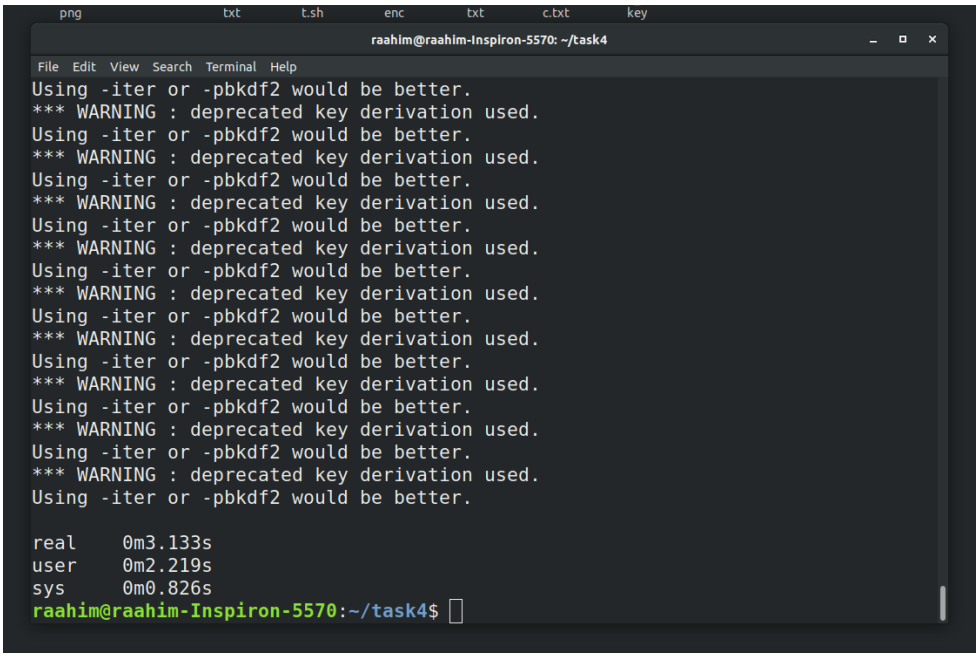
Similarly, for AES the same bash script was modified and used:

helperscript.sh

C: > Users > gs > Desktop > lab6 > task4 > helperscript.sh

```
#!/bin/bash
x=1
while [ $x -le 1000 ]
do
    openssl aes-128-cbc -in message.txt -out message.enc -pass pass:raahim12
    x=$(( $x + 1 ))
done
```

Output was:



```
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

real    0m3.133s
user    0m2.219s
sys     0m0.826s
raahim@raahim-Inspiron-5570:~/task4$ 
```

Command used to run the above scripts were:

“**time ./helperscript.sh**”

Even after executing both operation a large number of times and taking the average there is not much difference between the speeds of RSA and AES. We should use a large file size in order to affectively differentiate between the speeds of the two algorithms.

Bench-marking results for RSA were:

```
raahim@raahim-Inspiron-5570: ~/task4
File  Edit  View  Search  Terminal  Help
Doing 7680 bits private rsa's for 10s: 297 7680 bits private RSA's in 10.01s
Doing 7680 bits public rsa's for 10s: 52302 7680 bits public RSA's in 10.00s
Doing 15360 bits private rsa's for 10s: 59 15360 bits private RSA's in 10.02s
Doing 15360 bits public rsa's for 10s: 13408 15360 bits public RSA's in 10.00s
OpenSSL 1.1.1  11 Sep 2018
built on: Tue Nov 12 16:58:35 2019 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -
O2 -fdebug-prefix-map=/build/openssl-kxN_24/openssl-1.1.1=. -fstack-protector-st
rong -Wformat -Werror=format-security -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSS
L_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN
_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK16
00_ASM -DRC4_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DGHASH_ASM -DECP_N
ISTZ256_ASM -DX25519_ASM -DPADLOCK_ASM -DPOLY1305_ASM -DNDEBUG -Wdate-time -D_FO
RTIFY_SOURCE=2
                  sign    verify    sign/s verify/s
rsa  512 bits 0.000040s 0.000002s  25296.5 455872.7
rsa 1024 bits 0.000083s 0.000005s  12058.2 192877.4
rsa 2048 bits 0.000546s 0.000016s   1831.2  61616.2
rsa 3072 bits 0.001650s 0.000033s    606.0  29995.0
rsa 4096 bits 0.003674s 0.000058s    272.2  17104.5
rsa 7680 bits 0.033704s 0.000191s     29.7   5230.2
rsa 15360 bits 0.169831s 0.000746s      5.9   1340.8
raahim@raahim-Inspiron-5570:~/task4$
```

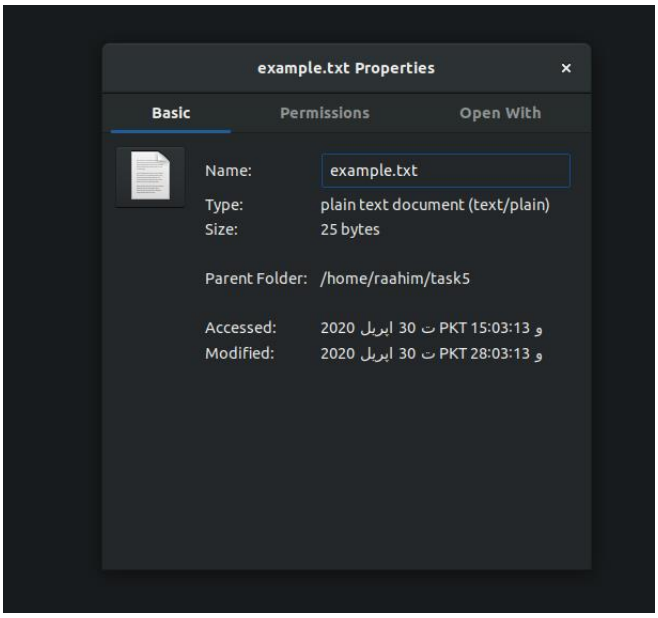
Bench-marking results for AES were:

```
raahim@raahim-Inspiron-5570: ~/task4
File  Edit  View  Search  Terminal  Help
Doing aes-256 cbc for 3s on 1024 size blocks: 358885 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 8192 size blocks: 45011 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 16384 size blocks: 22578 aes-256 cbc's in 3.00s
OpenSSL 1.1.1  11 Sep 2018
built on: Tue Nov 12 16:58:35 2019 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -
O2 -fdebug-prefix-map=/build/openssl-kxN_24/openssl-1.1.1=. -fstack-protector-st
rong -Wformat -Werror=format-security -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSS
L_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN
_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK16
00_ASM -DRC4_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DGHASH_ASM -DECP_N
ISTZ256_ASM -DX25519_ASM -DPADLOCK_ASM -DPOLY1305_ASM -DNDEBUG -Wdate-time -D_FO
RTIFY_SOURCE=2
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  1
6384 bytes
aes-128 cbc     146631.65k   162844.05k   168269.57k   167302.49k   164566.36k
 165942.61k
aes-192 cbc     121472.55k   134595.39k   127405.14k   138523.65k   139834.71k
 139329.54k
aes-256 cbc     105671.33k   114308.74k   121729.19k   122499.41k   122910.04k
 123305.98k
raahim@raahim-Inspiron-5570:~/task4$
```

My observations are almost similar to those from the outputs of the speed command. There was a slight difference but then again all this depends on the size of file as well as other factors such as number of bits used (in the case of RSA or the block size (in the case of AES).

- **Task 5:**

For this task, I first generated a file “example.txt” of size 25 bytes.



For preparing RSA public/private key pair I used the command:

“**openssl genrsa -aes128 -out task5.key 1024”**

Before signing, I first generated the public key from the public/private key pair using the following command:

“**openssl rsa -pubout -in task5.key -out pub_key.key**”

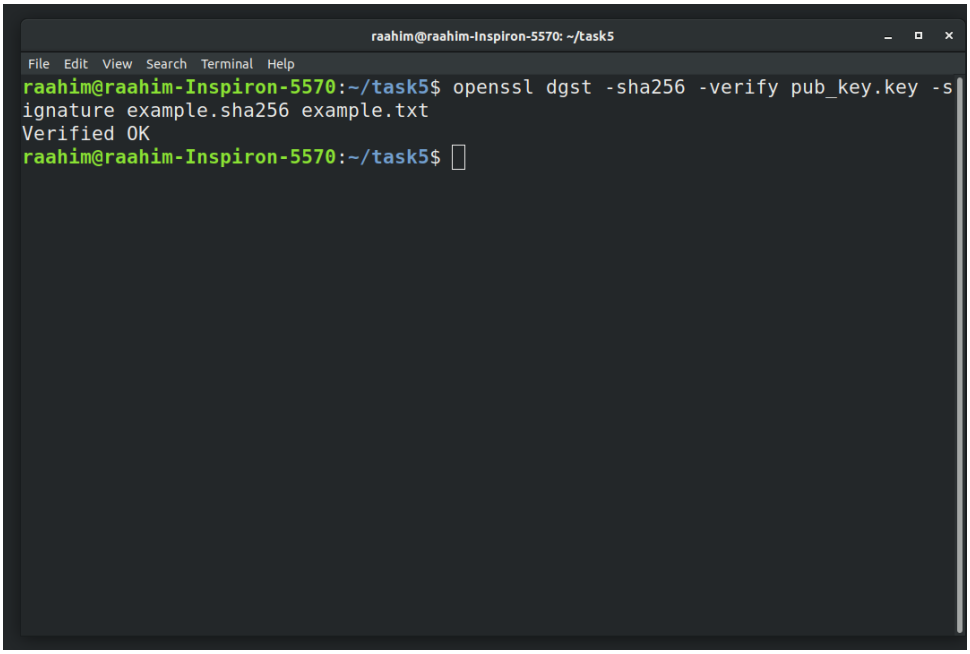
For signing:

“**openssl dgst -sha256 -sign task5.key -out example.sha256 example.txt**”

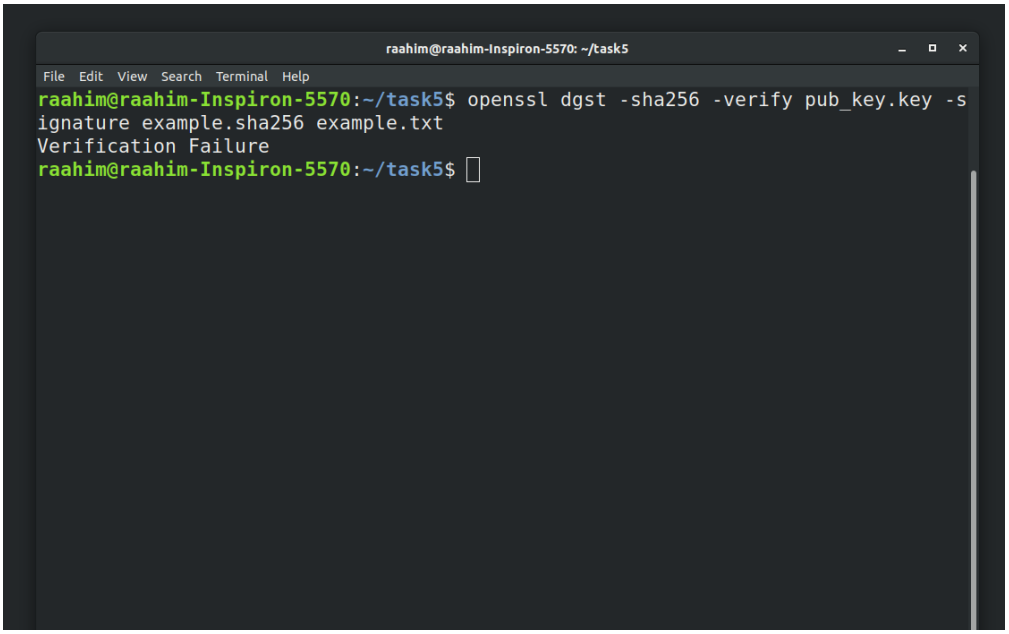
For verifying:

“**openssl dgst -sha256 -verify pub_key.key -signature example.sha256 example.txt**”

And the output was:



After modifying “example.txt” and verifying the digital signature again using the same command used above I got the following output:



Verification failed upon modifying “example.txt” because digest file became corrupted or we can say illegal after the modification. This happened because hash functions cause a noticeable or significant change in the output if a little modification is made in the original file.

Digital signatures are useful because they permit us to verify the integrity of a file in a secure way. Hash function provides the integrity and signature process provides the identification. Moreover, they increase the transparency of online transactions and develop trust between customers and vendors.
