# CS6903:Network Security
# Asg2:Openssl Tutorial

## PART A: Secure file transfer between Alice and Bob

Bob - Nisha M (CS19BTECH11012)
Alice - Vikas K (CS19BTECH11045)

The following steps were followed for the PART A:

1. Creating RSA(2048) key pairs for alice and bob . Names of the keys will be suffixed with our roll numbers respectively
    a. Creating a private key for Alice and encrypting it with a password.
    Command:

```
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt
rsa_keygen_bits:2048 -out Alice45_privatekey.pem
```

The output was as follows :

```
..................+++++
....+++++
Enter PEM pass phrase:alice@2
Verifying - Enter PEM pass phrase:alice@2
```

    b. Creating a private key for Bob and encrypting it with a password.
    Command :

```
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt
rsa_keygen_bits:2048 -out Bob12_privatekey.pem
```

Output :

```
.....+++++
.................................+++++
Enter PEM pass phrase:bob@2
Verifying - Enter PEM pass phrase:bob@2
```

c. The content of the private keys

i. Alice's private key :

Command :

```
$ cat Alice45_privatekey.pem
```

Output -

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIebp2BI1uDaQCAggA
MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBB0lV2BwLHbrC5vx8ciy98wBIIE
0M/YnojRyKZBet8adSgfQQoaqGYduRWdQYbq6CiJ7oZYt6etzLJeju5H19IzcT3f
1WFCHmCKEMpTySl4+PpbOKLVjqDgFgvpcBL/d5hoiOCdP7w7HdWjMU1zgorsdOi0
flrsSFWHudEmV57RmY9Nns2L2qNN4hOLFgzpEzQ7XJ1fd0+sl+q8f2F35MHI3JxS
BFgfJpEiFxMGHFS+4hl6kAFdYILbwxY4xC+N8aFiW1BbFPI7JiaEOq9Jmb3BytG0
/FxTOega34IxCpJsdrR3HP5cqVKw/nK05zROK8p8lsY/aIjr9nyZAIrnwP5ewU52
s+5NDyW8QEMWag4ImHnS+Am9dVU1Mbkm4/dgDx4Ce5718RKIsd5S5TwH2NQTGYI7
2zxnq1PZBhY6R98+dtAcROGO4W5kn7tFQ1+P4Z9oNJSOkaAdGDeMwi33Hg5GZ5CF
Nn1IOZUo3r1+EiuRlmbNItQll0lF5jTol+00XBinMLYpEmAaohCOeOYw4HFxS6cF
K4g7JQUsoqyvjEsSeKdoXLPP6qbPXmL1GKlCzsENGLIVEjeKile96jZ9B4dlOGfA
d6wPfQJCqq3TUk5JjgFD3uAjLA1KJASYVdsu3KWBQUsHR2ViEdpN+ooO09hYNEBS
tUiBrOe/ziIt+CnsrRfwhsm4ZOpaFP3XTbkk5JKLuouIHB1DfrRoLFYfe27tOz2a
rJiXkn/58uoSZG+zIXDLuXmDnKlheV9qVGj775OlwciOU9lKzFF5dzibd3xcUwei
6pfb3AKQM/qSHIPjyYnhbz+dAZXyDtWgJwO0VMbcfhx2JeVadqphpxmwzK0Dzr/W
y3CkfLfOQdV3JOyPEZ9Z/cWdleO8R2eJj0ln9sA0TaYs7SlqB4JMSjRKdMgov/RN
ViX2FBNt6sE3LImzaHxEl3M0fA5ISo+3Cnp+J6doprXOXg6SsFkcRJ8rebJB+zko
hOBBZLKnvPXAqKODze5DMgKFH+DGMD65oAqOO5pDOLGv2FW0JjDmk5oQLfRj6YOz
c8nhRRshRCvzyjNSJSiNFON45OiC/jga081FFJ0nZ/jxOjF58tLnmzqIhfbQY/ch
FWnBltyG+NNjLD2mNgdwkXYqQv3y0lPTAAUMSKJqCoQDeEbBCtyypR0ZIBWuz682
q8qJae15nq4HBTwzA+MFDN+4XSkWmcx/oCv5CycOuh7r+eJ8TYHVT/1lEdD/1RZB
cF4iKUdqIxHww3Jl/DHEiiRS1cgnPWCu0xLIWBee9Tn6eJV3f9LdF7HyQuAhQ/ED
ATrNl7WVYfcCRzV/oLDHdVpwGkYa4L6L0ZjM3K2Ky1DtCBrwKKZT85ESjw47MvTG
7UC4KG+Yd4/Uv/NztOpOjzDyFh8E/trFhyKHAYLvDiKGoEuNzdKnyPCe3SRFQukX
mfYboxkTKn6+xtAwhEO3iVsRcdPTFLa4LJrzIFJQZ48J/e/XtNfs3V4PJRVkpT8a
XzbOgVbiDYGkJSVtnqLV3Dhd7F9sG1EFmnZUVqNCeOMEzgUn2u1lHWE0A3RoRET9
GWo3jJmGAXElCfgLUGsGbkaGacHjc8ODTDSlhSzR1X4uQ9PbAtZ97N24f686ZI3Z
lnccKSmR2mMYXiX5ie8VwHE+4H5qcPj6YZxircY3Qh2O
-----END ENCRYPTED PRIVATE KEY-----
```

ii. Bob's private key:

Command :

```
$ cat Bob12_privatekey.pem
```

Output -

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIBwjScf9PrCgCAggA
MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBA9mamoOEtIQYTkahRxFHzVBIIE
0EquDf6qQP5nXFOmqEP4WCQRQaLpn0oe5Q7uBe4yhZD8KGS9XTdsyWxvTmS9d0FC
DCxCh5NYwgq6DI1U+QFxT/ehG1pw8u2+EJFvHNJbWDBCD4HlEs0VeznR/JqVfayn
UEKSHX6z+t0F36/B61JUDTbmGFG676ZWUWumzxeQ5ga2jqpzwr5X2smm+wKJmP5y
e94nnnmQZT5q1MDH751IE6SPEvV8PgYKoZi6onmVqfPH8HspRMg8QAPpPoJdyghQ
o9yLW8ZhQSTpTXbBFuoXnsiEekPgFzL3e8zUa5UIiFnlnq2o9ey9T4y+iZHPTiNw
lKf1pZ3GJWHeAHeVkvzi94Y2vWvEg3iiLMigTcdPqRrbe5PcDXMxewi7lqBeByTF
ZAiQvrrD5BJV4wr4d+0OOHm3LfT26H0BGkKd+MWSlt6aTqI8OHCyvgm3uBEcDnFL
NU2uR+/bfIN2RRhF1i4fv6g84salW+Hokj3hFF3g1JKQASIrEaHdaY5oZB/QXmwJ
AINrynbyrvkFh2P7c1brN20GhVA8tyO3O1sb/gvKCc1He7wjiECTv+8W64K0Yj8U
pp/6ua2jaquP+n5wxVsHUcrmeMlKKHjbc8UF/GWRFqS58EW+OePHwQMwLNci8seo
i0aslU2yH6HQOmTrd+PmGlxCpZNW6XFrRhraUte1kqnIMczvLl1SqkiyQgE03i0/
mUy7rn3j/QY1QHbmaT7qROQIeyBpcd5Hbc9LOaaum0+bLlGP6/ZC3R1F4A1TN0wm
K59CHYViyVKOeWzIepJFuBfaPZuOei5pa0YDjZPB09EKOwhHusEq3MNl0LcuR/GV
bEU1755kpQG1I5v8rc0JwAFM5Vq+YFZM6HFbQ0pFuYjShe3RTs0K0pu0SBCFsMis
xPI5cbUz0vecIe+i5TO9CCDskd8pQ+yeDCc9S/uFB/9xOf2Z4AjgNiWolzXZIMrY
LCHPS9x+xsv6IPTtU8EMd6idcB0iacuD0dMyqJkZ2wOgL6LCWMH+SUHNWAd54RtH
i5k8EaSrzA930kA05Eb7383BEpkFJk/vSn/5CXW9NBfyUAu9/xQ4Zmx4pgfAh7CZ
9OQMllF9z9oJZa3qznHd1V54GV6/Ov1gwLvTzu6UuZ0fSH0qxQQHLsyNrDsDIkW4
qa/kcotx+BfUHMJfeaTeAhUwGcacti1610v12boZTmN+8mQuOtn2LLby6NAhIlu/
WfACpz17lOZ2NV7StDpgT+gHyjLHc17/CVKjj4g2bTUkx8yC4WxIFY9UArZnCZDo
UACMTvw+y67I2Z10grP4XmKPyeZ/yjpJVw63ctOF1Oz80L8GtM1LyjCpZlGG/mC7
h+U3wKyvkOX7z9ICgHdViCqqGVe9Zgi8qQ2tEyKcs5O5o5SIoY7ouPnN1xsFHi/P
B3uo+1kJK7AnFaUEmbunn77yJW4T9FbXjuukRdA2YjKAfUjUhwhXlpWnpVsR1zFo
eJt+XAXBtgl/rj039F9lhyDhyRkAy9Q1wIb56CNlzBhVYCQ2PbqLkFPoD5MQPu3H
arXsXidoMyYzoimB+w8x7QlRL9zvhvYFsnsl5jZD4hGxcjTG+hGADcqre5+EER/U
aoR5PRix953dS6rugdIm2/DxC9lNQ1fyKiLZxZgRorIB
-----END ENCRYPTED PRIVATE KEY-----
```

d. Generation of public keys:

i. Generating the public key for Alice:

Command:

```
 $ openssl pkey -in Alice45_privatekey.pem -out
Alice45_publickey.pem -pubout
```

Output -

```
Enter pass phrase for Alice45_privatekey.pem: alice@2
```

ii. Generating the public key for Bob:

Command:

```
 $ openssl pkey -in Bob12_privatekey.pem -out
Bob12_publickey.pem -pubout
```

Output -

```
Enter pass phrase for Bob12_privatekey.pem: bob@2
```

e. The content of public keys :

i. Alice :

Command-

```
$ openssl pkey -in Alice45_publickey.pem -pubin -text
```

Output -

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArtKHU2MpIvnPoB9zMfPM
y4WO6Hbtuy4EJAySPiIqLffUjbsJ39d3obncucgztfg8wO2oJUOgmaO22TjhUWgM
mcd5nL06EMSddWpkQZD28+2N/X/pAkhYS6cnq7lZ1FXeF2jo4mm9WnsUHl42DJmR
4kxQE+zVdpDib64m+5DEziWgudtFschLD2mhSJsXF4GywfJEDk0fOqJR5g4ku6ch
bpzQc+58JlfEEghUg00vE+rOdbZNhkwEpm7u3oZSNzJ6UbEtA6hWlrGz0j21Yx++
4iFSbRTaTIQUgodM8r/AAdFMe6fjSASBwu+DMBF7vAqC4IxaqvMrTYiyfFvretow
```

```
tQIDAQAB
-----END PUBLIC KEY-----
RSA Public-Key: (2048 bit)
Modulus:
    00:ae:d2:87:53:63:29:22:f9:cf:a0:1f:73:31:f3:
    cc:cb:85:8e:e8:76:ed:bb:2e:04:24:0c:92:3e:22:
    2a:2d:f7:d4:8d:bb:09:df:d7:77:a1:b9:dc:b9:c8:
    33:b5:f8:3c:c0:ed:a8:25:43:a0:99:a3:b6:d9:38:
    e1:51:68:0c:99:c7:79:9c:bd:3a:10:c4:9d:75:6a:
    64:41:90:f6:f3:ed:8d:fd:7f:e9:02:48:58:4b:a7:
    27:ab:b9:59:d4:55:de:17:68:e8:e2:69:bd:5a:7b:
    14:1e:5e:36:0c:99:91:e2:4c:50:13:ec:d5:76:90:
    e2:6f:ae:26:fb:90:c4:ce:25:a0:b9:db:45:b1:c8:
    4b:0f:69:a1:48:9b:17:17:81:b2:c1:f2:44:0e:4d:
    1f:3a:a2:51:e6:0e:24:bb:a7:21:6e:9c:d0:73:ee:
    7c:26:57:c4:12:08:54:83:4d:2f:13:ea:ce:75:b6:
    4d:86:4c:04:a6:6e:ee:de:86:52:37:32:7a:51:b1:
    2d:03:a8:56:96:b1:b3:d2:3d:b5:63:1f:be:e2:21:
    52:6d:14:da:4c:84:14:82:87:4c:f2:bf:c0:01:d1:
    4c:7b:a7:e3:48:04:81:c2:ef:83:30:11:7b:bc:0a:
    82:e0:8c:5a:aa:f3:2b:4d:88:b2:7c:5b:eb:7a:da:
    30:b5
Exponent: 65537 (0x10001)
```

ii. Bob :

Command :

```
$ openssl pkey -in Bob12_publickey.pem -pubin -text
```

Output -

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxC61uoprAp/7SlhCR1e2
dKN1WAGOuaJsN6GTofvvZRm2rRcFq/dPF9LPAv/nIBhumLgFi7Uclg/1Gmai9wrp
ckFdwhk/W/R98NgUZCZx8GTBqfU0XkV3s97VS2wjCzJizzD5rnFzYDvk7XfXmIu8
jfyHyywfdJ8RO0hkLh2rtF5lSuxD/Jw2RjPUgQnBFDHj1gyVrwP9devTzNqaTH/a
3jRqbEtnI3yCVVFf43K71r+nqMl7XCWzTbhGuUOSvchoMF+8PlfFgU12Jb4rDUfp
kSakaxapArfZARZgylmr/qoqJwowizx9wCoRzfmRL5DuOsr8K7NS2Ojj9Dd8FfuY
VQIDAQAB
-----END PUBLIC KEY-----
RSA Public-Key: (2048 bit)
Modulus:
    00:c4:2e:b5:ba:8a:6b:02:9f:fb:4a:58:42:47:57:
    b6:74:a3:75:58:01:8e:b9:a2:6c:37:a1:93:a1:fb:
    ef:65:19:b6:ad:17:05:ab:f7:4f:17:d2:cf:02:ff:
```

```
    e7:20:18:6e:98:b8:05:8b:b5:1c:96:0f:f5:1a:66:
    a2:f7:0a:e9:72:41:5d:c2:19:3f:5b:f4:7d:f0:d8:
    14:64:26:71:f0:64:c1:a9:f5:34:5e:45:77:b3:de:
    d5:4b:6c:23:0b:32:62:cf:30:f9:ae:71:73:60:3b:
    e4:ed:77:d7:98:8b:bc:8d:fc:87:cb:2c:1f:74:9f:
    11:3b:48:64:2e:1d:ab:b4:5e:65:4a:ec:43:fc:9c:
    36:46:33:d4:81:09:c1:14:31:e3:d6:0c:95:af:03:
    fd:75:eb:d3:cc:da:9a:4c:7f:da:de:34:6a:6c:4b:
    67:23:7c:82:55:51:5f:e3:72:bb:d6:bf:a7:a8:c9:
    7b:5c:25:b3:4d:b8:46:b9:43:92:bd:c8:68:30:5f:
    bc:3e:57:c5:81:4d:76:25:be:2b:0d:47:e9:91:26:
    a4:6b:16:a9:02:b7:d9:01:16:60:ca:59:ab:fe:aa:
    2a:27:0a:30:8b:3c:7d:c0:2a:11:cd:f9:91:2f:90:
    ee:3a:ca:fc:2b:b3:52:d8:e8:e3:f4:37:7c:15:fb:
    98:55
Exponent: 65537 (0x10001)
```

2. Exchange of public keys between Alice and Bob -

   The public key of Alice “Alice45_publickey.pem” was sent to Bob over email and Bob’s public key “Bob12_publickey.pem” was sent to Alice in the same manner.

3. Creating the .key files
   a. Alice creates SA45.key text file containing info <symmetric encryption algo, its parameters and passphrase>

   The content of the file :

   Command -

   ```
   $ cat SA45.key
   ```

   Output -

```
aes-256-cbc, 1000, alice@2
```

   b. Bob creates SB12.key text file containing info <symmetric encryption algo, its parameters and passphrase>

   The content of the file :

   Command -

   ```
   $ cat SB12.key
   ```

   Output -

```
aes-256-cbc, 1200, bob@2
```

4. Authenticity and integrity check
    a. Alice - Signing the files (generating signature for SA45.key file)

        Command :

```
$ openssl pkeyutl -sign -in SA45.key -out Alice45_sign.key
-inkey Alice45_privatekey.pem
```

        Output -

```
Enter pass phrase for Alice45_privatekey.pem: alice@2
```

    b. Bob - Signing the files (generating signature for SB12.key file)
        Command :

```
$ openssl pkeyutl -sign -in SB12.key -out Bob12_sign.key
-inkey Bob12_privatekey.pem
```

        Output -

```
Enter pass phrase for Bob12_privatekey.pem: bob@2
```

    c. Exchange of the .key files -
        Alice sends both “SA45.key” and “Alice45_sign.key” to Bob and Bob sends both “SB12.key” and “Bob12_sign.key” files to Alice.

5. Verifying each others’ integrity and authenticity with the help of the exchanged files
    a. Alice verifying Bob’s identity: Alice with the help of Bob’s public key and the SB12.key checks the identity of Bob that if it was indeed signed by Bob himself and it was not tampered by man in the middle. The below command description is as follows - Alice extracts the signed file with the help of Bob’s public key and checks if it matches with the SB12.key. If yes, then both authenticity and integrity is preserved.
        Command :

```
$ openssl pkeyutl -verify -sigfile Bob12_sign.key -in
SB12.key -inkey Bob12_publickey.pem -pubin
```

Output -

```
Signature Verified Successfully
```

b. Bob verifying Alice's identity: The same procedure as above is repeated by Bob as well.
Command :

```
$ openssl pkeyutl -verify -sigfile Alice45_sign.key -in
SA45.key -inkey Alice45_publickey.pem -pubin
```

Output -

```
Signature Verified Successfully
```

6. Encrypting large files, signing them and exchanging the same
   a. Alice encrypts a large file with SA45.key ,signing it .
   Command :

```
$ openssl enc -aes-256-cbc -e -iter 1000 -salt -in
alice_original_img.png -out alice_encrypt_img.png
```

Output -

```
enter aes-256-cbc encryption password:alice@2
Verifying - enter aes-256-cbc encryption password:alice@2
```

b. Bob encrypting a large file with SB12.key ,signing it and sending to Alice
Command :

```
$ openssl enc -aes-256-cbc -e -iter 1200 -salt -in
bob_original_img.png -out bob_encrypt_img.png
```

Output -

```
enter aes-256-cbc encryption password:bob@2
Verifying - enter aes-256-cbc encryption password:bob@2
```

c. Alice sends “alice_encrypt_img.png” to Bob and Bob sends “bob_encrypt_img.png” to Alice.

7. Decrypting the received files and checking authenticity
   a. Alice verifying Bob - Alice uses Bob’s SB12.key to decrypt the received file
   Command :

```
$ openssl enc -aes-256-cbc -d -iter 1200 -in
bob_encrypt_img.png -out bob_decrypt_img.png
```

Output -

```
enter aes-256-cbc decryption password:bob@2
```

   b. Bob verifying Alice - Bob uses Alice’s SA45.key to decrypt the received file
   Command :

```
$ openssl enc -aes-256-cbc -d -iter 1000 -in
alice_encrypt_img.png -out alice_decrypt_img.png
```

Output -

```
enter aes-256-cbc decryption password:alice@2
```

## PART B:

1. Generating CSR and viewing it before sending it to root CA.
   a. Bob creates a CSR with his private key
   Command :

```
$ openssl req -config /etc/ssl/openssl.cnf -new -key
Bob12_privatekey.pem -out bob12_browser.csr
```

Output :

```
Enter pass phrase for Bob12_privatekey.pem:
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]:IN
State or Province Name (full name) [Some-State]:Telangana
Locality Name (eg, city) []:Sangareddy
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IITH
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:Bob
Email Address []:bob@email.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:bob@2
An optional company name []:IITH
```

b. Viewing the CSR generated
Command :

```
$ openssl req -in bob12_browser.csr -text
```

Output :

```
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = IN, ST = Telangana, L = Sangareddy, O = IITH, CN =
Bob, emailAddress = bob@email.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
```

```
                    00:c4:2e:b5:ba:8a:6b:02:9f:fb:4a:58:42:47:57:
                    b6:74:a3:75:58:01:8e:b9:a2:6c:37:a1:93:a1:fb:
                    ef:65:19:b6:ad:17:05:ab:f7:4f:17:d2:cf:02:ff:
                    e7:20:18:6e:98:b8:05:8b:b5:1c:96:0f:f5:1a:66:
                    a2:f7:0a:e9:72:41:5d:c2:19:3f:5b:f4:7d:f0:d8:
                    14:64:26:71:f0:64:c1:a9:f5:34:5e:45:77:b3:de:
                    d5:4b:6c:23:0b:32:62:cf:30:f9:ae:71:73:60:3b:
                    e4:ed:77:d7:98:8b:bc:8d:fc:87:cb:2c:1f:74:9f:
                    11:3b:48:64:2e:1d:ab:b4:5e:65:4a:ec:43:fc:9c:
                    36:46:33:d4:81:09:c1:14:31:e3:d6:0c:95:af:03:
                    fd:75:eb:d3:cc:da:9a:4c:7f:da:de:34:6a:6c:4b:
                    67:23:7c:82:55:51:5f:e3:72:bb:d6:bf:a7:a8:c9:
                    7b:5c:25:b3:4d:b8:46:b9:43:92:bd:c8:68:30:5f:
                    bc:3e:57:c5:81:4d:76:25:be:2b:0d:47:e9:91:26:
                    a4:6b:16:a9:02:b7:d9:01:16:60:ca:59:ab:fe:aa:
                    2a:27:0a:30:8b:3c:7d:c0:2a:11:cd:f9:91:2f:90:
                    ee:3a:ca:fc:2b:b3:52:d8:e8:e3:f4:37:7c:15:fb:
                    98:55
                Exponent: 65537 (0x10001)
        Attributes:
            unstructuredName         :IITH
            challengePassword        :bob@2
        Requested Extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client
Authentication, Code Signing, E-mail Protection
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha256WithRSAEncryption
         34:5d:ce:b2:97:74:ee:d1:2a:e0:f8:10:7b:f9:95:a5:e4:93:
         a9:34:56:4d:e6:07:f4:e4:48:9d:fc:35:34:ac:41:27:57:0e:
         bc:14:03:9a:a8:57:fe:c2:6f:d5:59:e5:02:53:89:34:32:53:
         a8:98:1f:7a:3c:d1:ae:35:00:53:55:69:21:8a:20:a9:0e:9c:
         27:09:46:ad:c0:97:c4:a6:98:47:a9:e9:04:b1:ab:3c:76:97:
         df:a4:41:12:dd:55:75:66:62:e8:df:4e:99:54:7e:57:3d:ed:
         b9:b8:86:d6:fc:48:c2:7f:e7:74:45:4e:cd:f4:23:bc:53:04:
         5a:d8:2e:32:fc:92:22:20:18:92:29:c6:45:b1:90:ea:61:50:
         04:7a:d5:4f:f3:b0:22:75:32:f1:39:0b:a2:c3:db:9a:de:92:
         81:e9:65:55:20:1c:7a:70:fd:b9:bf:e5:98:13:66:7d:92:56:
         a9:1c:e9:3f:fc:3c:29:4b:89:61:7f:9c:42:17:5c:1f:d8:97:
         d1:d9:af:74:db:b5:de:f9:23:f4:af:27:7f:de:9d:b0:1e:f5:
```

```
        c6:13:14:de:39:b6:f9:85:8b:a3:01:ce:b9:76:6d:a7:7e:26:
        ee:69:e2:0e:37:91:7d:85:2e:0e:74:6b:69:ec:15:b4:b1:2f:
        24:61:c7:97
-----BEGIN CERTIFICATE REQUEST-----
MIIDPjCCAiYCAQAwcTELMAkGA1UEBhMCSU4xEjAQBgNVBAgMCVRlbGFuZ2FuYTET
MBEGA1UEBwwKU2FuZ2FyZWRkeTENMAsGA1UECgwESUlUSDEMMAoGA1UEAwwDQm9i
MRwwGgYJKoZIhvcNAQkBFg1ib2JAZW1haWwuY29tMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAxC61uoprAp/7SlhCR1e2dKN1WAGOuaJsN6GTofvvZRm2
rRcFq/dPF9LPAv/nIBhumLgFi7Uclg/1Gmai9wrpckFdwhk/W/R98NgUZCZx8GTB
qfU0XkV3s97VS2wjCzJizzD5rnFzYDvk7XfXmIu8jfyHyywfdJ8RO0hkLh2rtF5l
SuxD/Jw2RjPUgQnBFDHj1gyVrwP9devTzNqaTH/a3jRqbEtnI3yCVVFf43K71r+n
qMl7XCWzTbhGuUOSvchoMF+8PlfFgU12Jb4rDUfpkSakaxapArfZARZgylmr/qoq
Jwowizx9wCoRzfmRL5DuOsr8K7NS2Ojj9Dd8FfuYVQIDAQABoIGHMBMGCSqGSIb3
DQEJAjEGDARJSVRIMBQGCSqGSIb3DQEJBzEHDAVib2JAMjBaBgkqhkiG9w0BCQ4x
TTBLMDEGA1UdJQQqMCgGCCsGAQUFBwMBBggrBgEFBQcDAgYIKwYBBQUHAwMGCCsG
AQUFBwMEMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgXgMA0GCSqGSIb3DQEBCwUAA4IB
AQA0Xc6yl3Tu0Srg+BB7+ZWl5JOpNFZN5gf05Eid/DU0rEEnVw68FAOaqFf+wm/V
WeUCU4k0MlOomB96PNGuNQBTVWkhiiCpDpwnCUatwJfEpphHqekEsas8dpffpEES
3VV1ZmLo306ZVH5XPe25uIbW/EjCf+d0RU7N9CO8UwRa2C4y/JIiIBiSKcZFsZDq
YVAEetVP87AidTLxOQuiw9ua3pKB6WVVIBx6cP25v+WYE2Z9klapHOk//DwpS4lh
f5xCF1wf2JfR2a9027Xe+SP0ryd/3p2wHvXGExTeObb5hYujAc65dm2nfibuaeIO
N5F9hS4OdGtp7BW0sS8kYceX
-----END CERTIFICATE REQUEST-----
```

2. Bob sends his CSR to Charlie who acts as the root CA and requests for the end user certificate.
3. Root CA signs the request and sends the end user cert to Bob and along with this the Charlie also sends its self signed certificate to Alice.
    a. Viewing end user certificate of Bob
    Command:

```
$ openssl x509 -in bob-browser.crt -text
```

Output -

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6d:23:87:12:5a:29:1d:63:f0:07:ae:48:ed:eb:4d:51:d0:be:8b:72
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = IN, ST = Telangana, L = Sangareddy, O = IITH, OU = CSE,
```

```
CN = Root_CA, emailAddress = charlie@email.com
        Validity
            Not Before: Feb  6 06:57:10 2022 GMT
            Not After : Feb  4 06:57:10 2032 GMT
        Subject: C = IN, ST = Telangana, L = Sangareddy, O = IITH, CN =
Bob, emailAddress = bob@email.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:c4:2e:b5:ba:8a:6b:02:9f:fb:4a:58:42:47:57:
                    b6:74:a3:75:58:01:8e:b9:a2:6c:37:a1:93:a1:fb:
                    ef:65:19:b6:ad:17:05:ab:f7:4f:17:d2:cf:02:ff:
                    e7:20:18:6e:98:b8:05:8b:b5:1c:96:0f:f5:1a:66:
                    a2:f7:0a:e9:72:41:5d:c2:19:3f:5b:f4:7d:f0:d8:
                    14:64:26:71:f0:64:c1:a9:f5:34:5e:45:77:b3:de:
                    d5:4b:6c:23:0b:32:62:cf:30:f9:ae:71:73:60:3b:
                    e4:ed:77:d7:98:8b:bc:8d:fc:87:cb:2c:1f:74:9f:
                    11:3b:48:64:2e:1d:ab:b4:5e:65:4a:ec:43:fc:9c:
                    36:46:33:d4:81:09:c1:14:31:e3:d6:0c:95:af:03:
                    fd:75:eb:d3:cc:da:9a:4c:7f:da:de:34:6a:6c:4b:
                    67:23:7c:82:55:51:5f:e3:72:bb:d6:bf:a7:a8:c9:
                    7b:5c:25:b3:4d:b8:46:b9:43:92:bd:c8:68:30:5f:
                    bc:3e:57:c5:81:4d:76:25:be:2b:0d:47:e9:91:26:
                    a4:6b:16:a9:02:b7:d9:01:16:60:ca:59:ab:fe:aa:
                    2a:27:0a:30:8b:3c:7d:c0:2a:11:cd:f9:91:2f:90:
                    ee:3a:ca:fc:2b:b3:52:d8:e8:e3:f4:37:7c:15:fb:
                    98:55
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client
Authentication, Code Signing, E-mail Protection
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha256WithRSAEncryption
         53:1a:a2:27:44:f6:47:6a:19:bf:0f:74:4c:cc:6c:fc:be:4a:
         3f:da:98:ef:44:a7:a3:f6:e5:5c:95:19:f2:4b:13:cb:1d:91:
         32:37:86:22:53:3b:1b:03:26:4a:75:fb:29:de:1e:ed:9d:a9:
         45:2a:68:4b:78:e0:ee:a3:ef:4d:97:6e:a7:ee:fc:11:d4:65:
         ea:c6:bf:13:4e:90:64:72:db:d1:ba:92:e2:39:0f:2f:32:dc:
```

```
        53:7a:c1:d0:04:65:87:de:9e:5f:49:09:3c:8a:35:6e:1b:65:
        85:1a:72:35:e9:cc:d0:22:03:f1:81:f0:96:fd:be:a1:7b:61:
        a7:ef:6c:d1:b2:74:21:9b:f2:bd:2b:db:e2:58:d3:0c:37:1e:
        92:91:51:bf:3f:b9:61:97:d1:9e:24:63:56:e6:1b:b1:62:93:
        48:73:e0:a6:61:cd:d2:26:01:bc:9c:58:00:2f:5a:31:ac:9c:
        47:fa:6d:e9:81:cc:e7:c6:2b:7e:fd:29:10:ef:fd:29:ca:c0:
        62:8a:c2:13:11:15:3a:01:05:55:00:51:60:55:9a:f2:0a:43:
        90:9e:a2:f7:20:bb:63:fd:00:34:2a:b4:3b:3c:3d:f0:ca:83:
        4c:b0:e2:f6:f5:3f:87:73:41:3a:65:22:e0:bf:5f:3e:2e:8d:
        60:ee:33:a2:6e:41:61:af:d9:63:ef:22:99:36:06:46:53:63:
        1e:2a:48:2f:ac:ad:44:a2:e8:98:a0:66:a3:fa:b6:f5:07:92:
        94:8c:7d:cd:ee:28:11:20:5b:4f:bd:1b:b3:78:b1:22:c7:fd:
        4c:01:ab:d9:f4:4d:fb:51:77:65:a1:8b:06:5a:ef:73:8c:b1:
        ed:39:5f:4d:a0:63:6e:e4:1a:6f:1c:27:fd:0e:cf:7e:dd:b4:
        1c:7c:27:36:e9:ff:fd:5f:41:e6:d4:9d:93:dc:48:5c:85:5e:
        23:73:c4:5e:3a:3a:60:0a:84:e0:3c:b8:57:d5:00:2a:d7:85:
        0d:b3:37:ca:ec:02:c8:8e:4c:85:b0:f5:80:ab:ff:6c:27:f1:
        b7:d7:2c:45:18:5e:42:18:fe:66:4a:89:e0:c5:bc:2a:12:ef:
        ce:ce:b1:ce:64:1e:61:b2:c8:d7:ab:27:6b:5d:ba:2f:bf:5e:
        96:01:bc:f7:5c:5d:77:06:e9:27:7a:1e:91:77:07:69:d6:78:
        33:97:b9:22:23:7b:e7:cd:60:63:22:31:65:86:59:40:71:e3:
        94:96:ea:8a:fb:32:f8:e7:e8:29:e6:63:be:d6:3c:f6:9d:24:
        01:63:1b:0c:05:e6:e5:9f:98:68:07:c7:83:ba:3f:73:5f:db:
        aa:b3:b5:cb:ab:99:d6:7b
-----BEGIN CERTIFICATE-----
MIIE1DCCArygAwIBAgIUbSOHElopHWPwB65I7etNUdC+i3IwDQYJKoZIhvcNAQEL
BQAwgYcxCzAJBgNVBAYTAklOMRIwEAYDVQQIDAlUZWxhbmdhbmExEzARBgNVBAcM
ClNhbmdhcmVkZHkxDTALBgNVBAoMBElJVEgxDDAKBgNVBAsMA0NTRTEQMA4GA1UE
AwwHUm9vdF9DQTEgMB4GCSqGSIb3DQEJARYRY2hhcmxpZUBlbWFpbC5jb20wHhcN
MjIwMjA2MDY1NzEwWhcNMzIwMjA0MDY1NzEwWjBxMQswCQYDVQQGEwJJTjESMBAG
A1UECAwJVGVsYW5nYW5hMRMwEQYDVQQHDApTYW5nYXJlZGR5MQ0wCwYDVQQKDARJ
SVRIMQwwCgYDVQQDDANCb2IxHDAaBgkqhkiG9w0BCQEWDWJvYkBlbWFpbC5jb20w
ggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDELrW6imsCn/tKWEJHV7Z0
o3VYAY65omw3oZOh++9lGbatFwWr908X0s8C/+cgGG6YuAWLtRyWD/UaZqL3Culy
QV3CGT9b9H3w2BRkJnHwZMGp9TReRXez3tVLbCMLMmLPMPmucXNgO+Ttd9eYi7yN
/IfLLB90nxE7SGQuHau0XmVK7EP8nDZGM9SBCcEUMePWDJWvA/1169PM2ppMf9re
NGpsS2cjfIJVUV/jcrvWv6eoyXtcJbNNuEa5Q5K9yGgwX7w+V8WBTXYlvisNR+mR
JqRrFqkCt9kBFmDKWav+qionCjCLPH3AKhHN+ZEvkO46yvwrs1LY6OP0N3wV+5hV
AgMBAAGjTTBLMDEGA1UdJQQqMCgGCCsGAQUFBwMBBggrBgEFBQcDAgYIKwYBBQUH
AwMGCCsGAQUFBwMEMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgXgMA0GCSqGSIb3DQEB
CwUAA4ICAQBTGqInRPZHahm/D3RMzGz8vko/2pjvRKej9uVclRnySxPLHZEyN4Yi
UzsbAyZKdfsp3h7tnalFKmhLeODuo+9Nl26n7vwR1GXqxr8TTpBkctvRupLiOQ8v
MtxTesHQBGWH3p5fSQk8ijVuG2WFGnI16czQIgPxgfCW/b6he2Gn72zRsnQhm/K9
```

```
K9viWNMMNx6SkVG/P7lhl9GeJGNW5huxYpNIc+CmYc3SJgG8nFgAL1oxrJxH+m3p
gcznxit+/SkQ7/0pysBiisITERU6AQVVAFFgVZryCkOQnqL3ILtj/QA0KrQ7PD3w
yoNMsOL29T+Hc0E6ZSLgv18+Lo1g7jOibkFhr9lj7yKZNgZGU2MeKkgvrK1EouiY
oGaj+rb1B5KUjH3N7igRIFtPvRuzeLEix/1MAavZ9E37UXdloYsGWu9zjLHtOV9N
oGNu5BpvHCf9Ds9+3bQcfCc26f/9X0Hm1J2T3EhchV4jc8ReOjpgCoTgPLhX1QAq
14UNszfK7ALIjkyFsPWAq/9sJ/G31yxFGF5CGP5mSongxbwqEu/OzrHOZB5hssjX
qydrXbovv16WAbz3XF13Bukneh6Rdwdp1ngzl7kiI3vnzWBjIjFlhllAceOUluqK
+zL45+gp5mO+1jz2nSQBYxsMBebln5hoB8eDuj9zX9uqs7XLq5nWew==
-----END CERTIFICATE-----
```

b. Viewing self signed certificate of charlie(Root ca)
Command :

```
$ openssl x509 -in charlie-ca.crt.pem -text
```

Output -

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            53:6d:83:32:51:2f:bb:3e:4a:36:ae:5b:f8:b7:46:ee:b4:b6:70:7f
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = IN, ST = Telangana, L = Sangareddy, O = IITH, OU = CSE,
CN = Root_CA, emailAddress = charlie@email.com
        Validity
            Not Before: Feb  1 19:33:58 2022 GMT
            Not After : Jan 27 19:33:58 2042 GMT
        Subject: C = IN, ST = Telangana, L = Sangareddy, O = IITH, OU =
CSE, CN = Root_CA, emailAddress = charlie@email.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:a8:b8:26:3a:b4:8e:e5:51:66:2c:70:b4:53:ad:
                    4b:ef:73:7e:b3:ed:23:b5:a1:d4:a6:99:16:b4:68:
                         fa:be:d5:e8:4b:45:f2:8e:6a:ee:4e:ea:7b:09:0b:
                    c4:f9:c1:b6:d3:23:8a:22:fa:dd:75:28:b2:20:b7:
                    06:c0:08:da:ee:3b:80:5c:87:e4:f9:b0:a3:ba:4a:
                    96:17:73:47:05:b7:3b:78:6b:7b:60:d4:60:e2:af:
                    0d:eb:72:d1:0a:ff:ac:d4:ae:8b:a0:2e:36:f2:0a:
```

```
                    0f:0a:1f:ec:89:06:27:1d:9a:51:65:ea:f2:6f:b6:
                    a6:80:bd:9e:b7:39:94:8a:59:1e:c7:6f:06:1e:e3:
                    70:d1:de:ad:b9:98:e7:2f:03:69:4f:71:b4:25:1a:
                    75:4b:fa:15:c9:20:08:44:40:19:1a:db:9d:63:e5:
                    ba:12:23:a4:35:78:f0:ff:80:66:ef:79:b2:4f:33:
                    1a:40:d2:4e:dd:df:3c:4f:89:de:21:29:17:49:7e:
                    1d:be:57:0c:5a:47:3b:61:a9:53:93:7c:49:31:70:
                    e5:7e:8b:03:73:b8:17:c9:0b:07:d0:7c:3e:df:47:
                    b8:40:51:83:30:df:58:06:ce:de:26:27:38:4e:e7:
                    b8:16:90:ab:5e:c3:38:ef:c2:b8:31:0e:48:96:86:
                    67:3b:59:50:33:b8:28:c8:1c:10:35:51:0c:12:39:
                    3d:3f:97:ea:58:6c:90:21:96:e3:2f:d3:09:4c:65:
                    52:68:f8:cd:f0:0a:1b:c2:10:73:95:76:c0:41:de:
                    c4:06:4a:14:a8:e4:9a:c5:27:9b:69:9c:52:18:5a:
                    10:e9:eb:1a:06:f5:fa:8b:13:95:c5:21:d0:b7:2d:
                    5a:f4:e0:d3:ab:e1:b3:36:72:61:0c:a3:ee:18:d2:
                    67:1a:c5:52:47:59:6e:cb:f0:fa:73:1f:cf:57:d8:
                    0c:c1:4f:ae:5a:36:57:09:d4:df:e7:83:b3:3d:98:
                    22:20:a1:0c:25:63:54:e7:6d:38:4b:37:08:23:9b:
                    1b:5d:28:68:aa:c6:09:75:47:19:9f:e0:4c:11:8f:
                    05:3a:57:73:59:c4:9a:89:bb:17:90:17:a7:8f:ce:
                    35:4d:43:e3:31:2c:bf:1a:13:97:f7:7b:04:c3:1b:
                    ec:6f:7d:0d:84:86:92:ec:cf:ad:a5:b4:8b:52:ba:
                    03:b3:37:b7:eb:08:9c:41:16:64:c0:aa:f0:35:44:
                    84:61:19:cb:76:cb:8e:04:e0:f0:f8:0a:12:0f:9e:
                         eb:dd:c2:51:ba:db:d8:e9:d6:e4:c6:aa:d1:29:b0:
                    47:13:45:63:48:30:e3:8d:30:a5:11:17:d3:be:8d:
                    8b:af:d9
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                08:66:D9:E2:85:6B:8B:48:C1:0C:98:1F:0B:54:B8:25:85:25:F6:6F
            X509v3 Authority Key Identifier:

keyid:08:66:D9:E2:85:6B:8B:48:C1:0C:98:1F:0B:54:B8:25:85:25:F6:6F

            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
         6c:a5:8d:a2:c3:34:29:d8:dd:7d:c1:af:28:f0:00:6d:76:1a:
         80:0a:c5:02:4d:bf:a2:cc:d6:39:82:64:3d:49:ff:81:80:be:
         88:6c:46:f9:5c:14:e0:5c:6e:19:7d:19:4e:d2:13:2a:ba:0f:
```

```
        0c:e4:ae:6f:71:20:f6:23:b0:d8:af:8e:7b:9b:96:01:47:9f:
        4f:32:59:2c:7a:ce:69:0a:39:01:e4:92:b9:98:67:02:0f:b5:
        01:65:0b:b7:cf:78:90:c2:48:c3:5a:f1:0a:cf:45:92:87:8b:
        48:d0:8d:6f:4d:b9:90:f6:4c:23:dc:a3:3c:62:0d:30:31:1b:
        e9:89:df:14:b4:76:13:7d:be:bb:7a:10:db:74:26:68:d6:6a:
        78:fa:56:bb:69:26:38:cc:d7:24:4b:68:83:ff:17:fa:89:f2:
        90:1b:89:8f:c7:bb:52:97:d8:2a:72:79:52:30:8c:70:21:25:
        a9:c5:66:56:94:dd:4a:73:07:6a:a7:d3:5b:f6:88:99:c5:7b:
        e6:73:14:ca:91:0b:11:41:b2:63:65:61:70:b9:b6:cf:c4:86:
        90:9c:80:75:b8:75:29:47:47:13:ec:0f:51:7b:cf:fa:41:d9:
        10:d4:56:72:42:eb:8b:d6:30:6e:df:0c:77:92:6e:31:08:c1:
        97:67:53:ec:7b:8c:86:cb:c9:8c:59:e8:7b:d4:81:e5:3e:e8:
        db:6a:58:1d:39:16:f8:eb:3b:42:44:f7:ca:53:46:47:b0:4a:
        ef:26:f6:7b:90:df:bf:29:c7:8e:a7:15:ec:41:6d:53:a3:73:
        c6:0a:36:d5:5b:d1:98:51:b9:08:4d:13:f7:79:90:85:e6:e2:
        10:db:a4:62:29:a8:97:fc:53:2c:39:1d:6c:d3:9c:62:dd:1b:
        cf:f2:02:3d:ad:0c:eb:fc:d0:5f:9c:e8:81:cb:1c:1b:6e:81:
        65:2c:81:e1:83:8e:97:f9:78:31:f3:60:92:ed:f3:98:91:b7:
        77:a6:9e:b9:65:67:e8:e3:f7:a5:2d:2f:cc:5b:be:bc:07:b3:
        e5:9c:ec:e5:ed:e5:26:41:99:75:5e:64:01:09:a1:0a:62:14:
        55:c3:9f:6b:35:3d:c8:59:79:8a:af:7e:66:00:56:b1:5b:f3:
        e9:c6:6b:05:31:6c:fa:1e:77:29:d3:4e:6c:27:b8:91:53:22:
        a4:d5:bb:96:b1:4b:e5:c2:89:71:86:5e:93:6e:17:14:ab:0a:
        76:f5:d8:fe:34:3e:cd:49:59:51:b6:34:0b:7e:83:3a:78:ef:
        48:18:9f:be:5d:05:b3:25:3f:04:e2:a5:8d:4b:1c:7b:72:1a:
        08:98:7c:59:00:61:ee:38
-----BEGIN CERTIFICATE-----
MIIF8TCCA9mgAwIBAgIUU22DMlEvuz5KNq5b+LdG7rS2cH8wDQYJKoZIhvcNAQEL
BQAwgYcxCzAJBgNVBAYTAklOMRIwEAYDVQQIDAlUZWxhbmdhbmExEzARBgNVBAcM
ClNhbmdhcmVkZHkxDTALBgNVBAoMBElJVEgxDDAKBgNVBAsMA0NTRTEQMA4GA1UE
AwwHUm9vdF9DQTEgMB4GCSqGSIb3DQEJARYRY2hhcmxpZUBlbWFpbC5jb20wHhcN
MjIwMjAxMTkzMzU4WhcNNDIwMTI3MTkzMzU4WjCBhzELMAkGA1UEBhMCSU4xEjAQ
BgNVBAgMCVRlbGFuZ2FuYTETMBEGA1UEBwwKU2FuZ2FyZWRkeTENMAsGA1UECgwE
SUlUSDEMMAoGA1UECwwDQ1NFMRAwDgYDVQQDDAdSb290X0NBMSAwHgYJKoZIhvcN
AQkBFhFjaGFybGllQGVtYWlsLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCC
AgoCggIBAKi4Jjq0juVRZixwtFOtS+9zfrPtI7Wh1KaZFrRo+r7V6EtF8o5q7k7q
ewkLxPnBttMjiiL63XUosiC3BsAI2u47gFyH5Pmwo7pKlhdzRwW3O3hre2DUYOKv
Dety0Qr/rNSui6AuNvIKDwof7IkGJx2aUWXq8m+2poC9nrc5lIpZHsdvBh7jcNHe
rbmY5y8DaU9xtCUadUv6FckgCERAGRrbnWPluhIjpDV48P+AZu95sk8zGkDSTt3f
PE+J3iEpF0l+Hb5XDFpHO2GpU5N8STFw5X6LA3O4F8kLB9B8Pt9HuEBRgzDfWAbO
3iYnOE7nuBaQq17DOO/CuDEOSJaGZztZUDO4KMgcEDVRDBI5PT+X6lhskCGW4y/T
CUxlUmj4zfAKG8IQc5V2wEHexAZKFKjkmsUnm2mcUhhaEOnrGgb1+osTlcUh0Lct
```

```
WvTg06vhszZyYQyj7hjSZxrFUkdZbsvw+nMfz1fYDMFPrlo2VwnU3+eDsz2YIiCh
DCVjVOdtOEs3CCObG10oaKrGCXVHGZ/gTBGPBTpXc1nEmom7F5AXp4/ONU1D4zEs
vxoTl/d7BMMb7G99DYSGkuzPraW0i1K6A7M3t+sInEEWZMCq8DVEhGEZy3bLjgTg
8PgKEg+e693CUbrb2OnW5Maq0SmwRxNFY0gw440wpREX076Ni6/ZAgMBAAGjUzBR
MB0GA1UdDgQWBBQIZtnihWuLSMEMmB8LVLglhSX2bzAfBgNVHSMEGDAWgBQIZtni
hWuLSMEMmB8LVLglhSX2bzAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUA
A4ICAQBspY2iwzQp2N19wa8o8ABtdhqACsUCTb+izNY5gmQ9Sf+BgL6IbEb5XBTg
XG4ZfRlO0hMqug8M5K5vcSD2I7DYr457m5YBR59PMlkses5pCjkB5JK5mGcCD7UB
ZQu3z3iQwkjDWvEKz0WSh4tI0I1vTbmQ9kwj3KM8Yg0wMRvpid8UtHYTfb67ehDb
dCZo1mp4+la7aSY4zNckS2iD/xf6ifKQG4mPx7tSl9gqcnlSMIxwISWpxWZWlN1K
cwdqp9Nb9oiZxXvmcxTKkQsRQbJjZWFwubbPxIaQnIB1uHUpR0cT7A9Re8/6QdkQ
1FZyQuuL1jBu3wx3km4xCMGXZ1Pse4yGy8mMWeh71IHlPujbalgdORb46ztCRPfK
U0ZHsErvJvZ7kN+/KceOpxXsQW1To3PGCjbVW9GYUbkITRP3eZCF5uIQ26RiKaiX
/FMsOR1s05xi3RvP8gI9rQzr/NBfnOiByxwbboFlLIHhg46X+Xgx82CS7fOYkbd3
pp65ZWfo4/elLS/MW768B7PlnOzl7eUmQZl1XmQBCaEKYhRVw59rNT3IWXmKr35m
AFaxW/PpxmsFMWz6Hncp005sJ7iRUyKk1buWsUvlwolxhl6TbhcUqwp29dj+ND7N
SVlRtjQLfoM6eO9IGJ++XQWzJT8E4qWNSxx7choImHxZAGHuOA==
-----END CERTIFICATE-----
```

4. Bob sends his end user cert received from Charlie to Alice.
5. This is the final step where Alice verifies whether the certificate of Bob is valid or not.
   Command :

```
 $ openssl verify -verbose -CAfile charlie-ca.crt.pem
bob-browser.crt
```

Output :

```
bob-browser.crt: OK
```