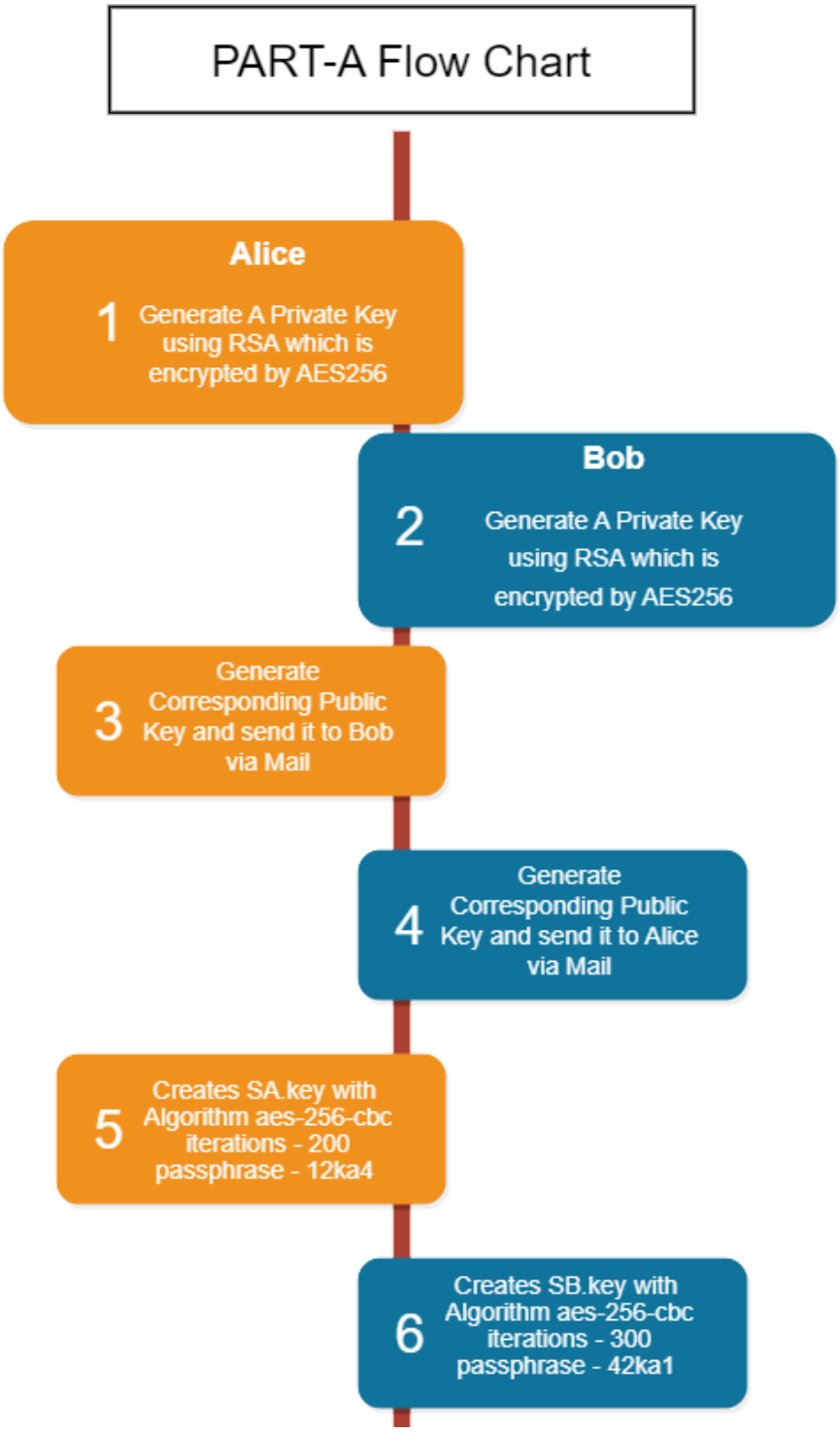
Assignment : OpenSSL Tutorial

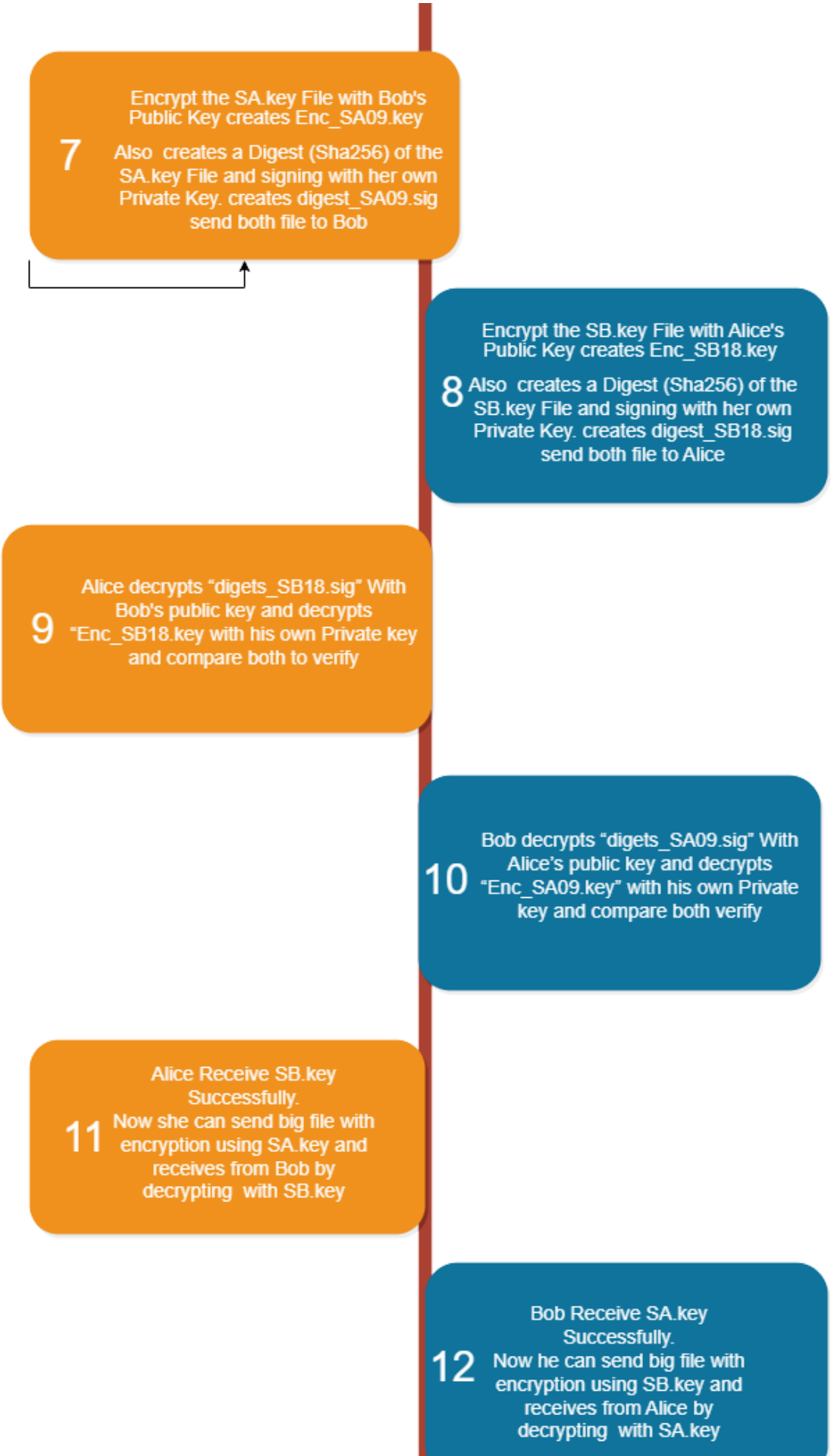
Name - Raj Popat
Signature - cs23mtech14009

Name - Yash Shukla
Signature - cs23mtech14018

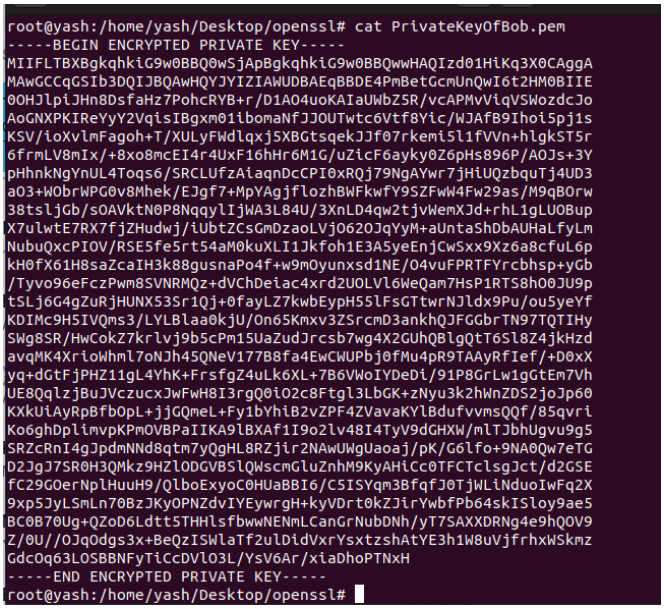
Quick Note : In this assignment, Raj acts as Alice , Yash as Bob , Roll No cs23mtech11026 - Bhargav as Charlie.

PART-A Flow Chart



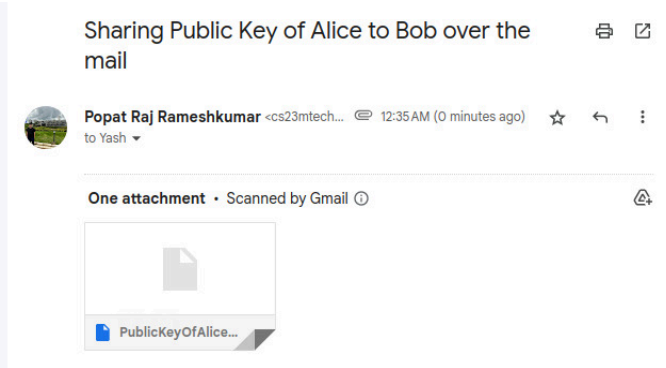

7
Encrypt the SA.key File with Bob's Public Key creates Enc_SA09.key
Also creates a Digest (Sha256) of the SA.key File and signing with her own Private Key. creates digest_SA09.sig send both file to Bob
8
Encrypt the SB.key File with Alice's Public Key creates Enc_SB18.key
Also creates a Digest (Sha256) of the SB.key File and signing with her own Private Key. creates digest_SB18.sig send both file to Alice
9
Alice decrypts "digets_SB18.sig" With Bob's public key and decrypts "Enc_SB18.key with his own Private key and compare both to verify
10
Bob decrypts "digets_SA09.sig" With Alice's public key and decrypts "Enc_SA09.key" with his own Private key and compare both verify
11
Alice Receive SB.key Successfully.
Now she can send big file with encryption using SA.key and receives from Bob by decrypting with SB.key
12
Bob Receive SA.key Successfully.
Now he can send big file with encryption using SB.key and receives from Alice by decrypting with SA.key

PART - A

Alice09	Bob18
1) & 2) - RSA Key Pair Generation and Exchange Creating Private key of Alice09 : we are using AES encryption to protect the private key file with a key size of 256 bits were PEM pass phrase is 1234	1) & 2) - RSA Key Pair Generation and Exchange Creating Private key of Bob18 : we are using AES encryption to protect the private key file with a key size of 256 bits were PEM pass phrase is 1234
 	
Viewing private key of alice09 :	Viewing Private key of Bob18 :
 	
Generating corresponding public key of Alice and viewing : here to use private key we have to enter passphrase 1234	Generating corresponding public key of Bob and viewing : here to use private key we have to enter passphrase 1234

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl pkey -in PrivateKeyOfAlice.pem
 -out PublicKeyOfAlice.pem -pubout
Enter pass phrase for PrivateKeyOfAlice.pem:
root@RajPopat:/home/raj/Desktop/NSAS-1# cat PublicKeyOfAlice.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwhgV4pszQ2b8lvzBX9wG
8iBfx88J1H6iHI11l3mp6ohmhZHbLqQdRb5W4Dony9HICq9iOPTIXGaUZMu0pcsw
ANMH7QKllWZNNVQpYMzBq8XJsPxHtVs5SigvREEO3pTHpc4fW122QxydUMP2n1Qb
Qvz6aQZQAT6O09WHpwGG0QByOt3YGOHy8CYDPEjCaTc36uiM5egoMztcSB5/D7nq
TTmsT/lbJolKraRmc1pJtGKnBrRIpFQrkG/sN8bO6H6feXdDzSWSDckZ0h4WvYlm
VKDZt0+b2r4BLb/nDJeAih9xy+Kx+E3kJq8uWyh6haiyFgOaSANIfKzL8YKwkMyt
HQIDAQAB
-----END PUBLIC KEY-----
```

Exchange of Public key over Email :



Creating symmetric key of alice and viewing :

We are creating SA.key which contain Algorithm, iterations and pass phrase

```
root@RajPopat:/home/raj/Desktop/NSAS-1# cat SA.key
Algorithm = aes-256-cbc
iterations = 200
pass phrase = 12ka4
```

```
root@yash:/home/yash/Desktop/openssl# openssl pkey -in PrivateKeyOfBob.
pem -out PublicKeyOfBob.pem -pubout
Enter pass phrase for PrivateKeyOfBob.pem:
root@yash:/home/yash/Desktop/openssl# cat PublicKeyOfBob.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmO9L/28FIzL6QECN1iyn
SEXenI0es44UrB0PBaRBxA8PyuyhRKS+dgg/PD6/6fj3VZ1vl7Whpl04gS6bx3W0
5GlyeekZdbyV/hclSsQLL3iWdDYZT8ieNtJ+Y0v4epM+fEHa4ZrUNlCBuIxaVrBo
jyAJhr5hxOvxhukD2CFADGWZSuExYCyvcmNqM7wbPh1vu3Llyqln+uXKjD2SDfFH
83R6Mhi9rm9lvkzn88IDIN8hRv4VS7vj6rLryYTC27ZWwEipdUKOzrdNxGy8QuEO
wnnXTA0KPc/oORDVRzDblsV3BWhOtFGQ0mIg5ziXQlpACDWldJ0o4BDEXw1lrsfr
RQIDAQAB
-----END PUBLIC KEY-----
root@yash:/home/yash/Desktop/openssl#
```

Exchange of Public key over Email :

Creating symmetric key of Bob and viewing :

We are creating SA.key which contain Algorithm, iterations and pass phrase

```
root@yash:/home/yash/Desktop/openssl# cat SB.key
Algorithm = aes-256-cbc
iteration = 300
pass phrase = 42ka1
```

3)
Mechanism for sending, verifying the message and authenticity and integrity :

Confidentiality :
Encrypt the SA.key File with Bob's Public Key

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl pkeyutl -encrypt -pubin -inkey .
./PublicKeyOfBob.pem -in SA.key -out Enc_SA09.key
root@RajPopat:/home/raj/Desktop/NSAS-1# ls
Enc_SA09.key  PrivateKeyOfAlice.pem  PublicKeyOfAlice.pem  SA.key
root@RajPopat:/home/raj/Desktop/NSAS-1# cat Enc_SA09.key
g◆&S◆Dc◆0◆◆◆◆,. oI_◆◆◆@~◆◆◆u◆B◆z◆◆@.◆~◆a◆7◆<◆m◆◆◆X| kn◆3◆◆NBb;◆◆◆?◆◆◆◆b◆gVQ◆h|
root@RajPopat:/home/raj/Desktop/NSAS-1#
```

Integrity and Authentication :

3)
Mechanism for sending, verifying the message and authenticity and integrity :

Confidentiality :
Encrypt the SB.key File with Alice's Public Key

```
root@yash:/home/yash/Desktop/openssl# openssl pkeyutl -encrypt -pubin -inkey ../PublicKe
yOfAlice.pem -in SB.key -out Enc_SB18.key
root@yash:/home/yash/Desktop/openssl# ls
Dec_SA09.key  Enc_SB18.key  PrivateKeyOfBob.pem  PublicKeyOfBob.pem  SB.key
root@yash:/home/yash/Desktop/openssl#
```

Integrity and Authentication :

Alice is creating a Digest (Sha256) of the SA.key File and signing with her own Private Key.

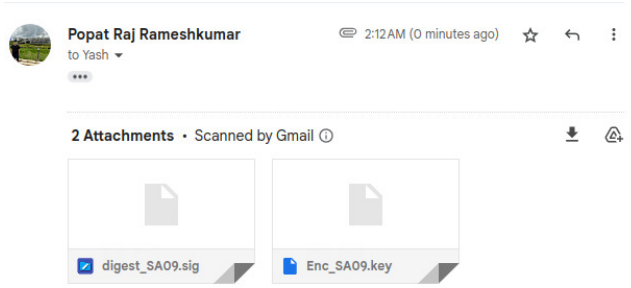
Alice will send both “Enc_SA09.key” and “digest_SA09.sig” to Bob.

where Bob decrypts “digest_SA09.sig” With Alice’s public key and decrypts “Enc_SA09.key” with his own Private key which provides Authenticity to Bob.

Then it will compare this digest with SA.key so it will provide the Integrity to Bob.

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl dgst -sha256 -sign PrivateKeyOfA
lice.pem -out digest_SA09.sig SA.key
Enter pass phrase for PrivateKeyOfAlice.pem:
root@RajPopat:/home/raj/Desktop/NSAS-1# ls
digest_SA09.sig  PrivateKeyOfAlice.pem  SA.key
Enc_SA09.key     PublicKeyOfAlice.pem
root@RajPopat:/home/raj/Desktop/NSAS-1# cat digest_SA09.sig
◆◆◆◆%}◆{◆k◆◆D ◆i),◆◆◆.eP3◆◆◆◆◆N◆◆m{◆b I◆◆,◆◆?◆z/◆ ◆◆◆W◆▶|◆y◆◆琥◆z{◆◆V   ◆^gk◆(/◆
◆◆\◆pt=u◆◆◆|◆1◆g◆]q◆W.◆◆?lk◆zN◆2+^◆;◆◆l"◆
       ◆E◆_◆◆◆U2◆◆!◆X◆◆◆◆騷◆*{N◆◆3
}◆G                                  &◆
cK);)◆jL◆◆$◆◆   ◆◆◆v$◆1◆2◆◆Ů◆◆:◆◆Sw$◆U◆>        Jd◆BZ◆◆◆◆s◆◆4◆4[illegible]h◆◆k5◆root@RajP
opat:/home/raj/Desktop/NSAS-1#
```

Sending Alice’s Encrypted File and Signed digest :



Decrypting file :

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl pkeyutl -decrypt -inkey PrivateK
eyOfAlice.pem -in ../Enc_SB18.key -out Dec_SB18.key
Enter pass phrase for PrivateKeyOfAlice.pem:
root@RajPopat:/home/raj/Desktop/NSAS-1# ls
Dec_SB18.key     Enc_SA09.key          PublicKeyOfAlice.pem
digest_SA09.sig  PrivateKeyOfAlice.pem  SA.key
```

```
root@RajPopat:/home/raj/Desktop/NSAS-1# cat Dec_SB18.key
Algorithm = aes-256-cbc
iteration = 300
pass phrase = 42ka1
root@RajPopat:/home/raj/Desktop/NSAS-1#
```

Verifying :

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl dgst -sha256 -verify ../PublicKe
yOfBob.pem -signature ../digest_SB18.sig Dec_SB18.key
Verified OK
root@RajPopat:/home/raj/Desktop/NSAS-1#
```

Bob is creating a Digest(Sha256) of the SB.key File and signing with his own Private Key.

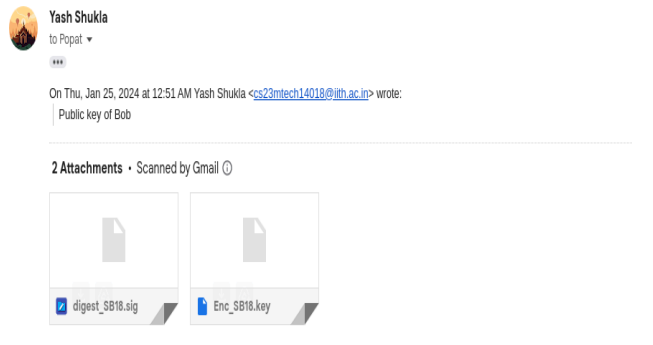
Bob will send both “Enc_Sb18.key” and “digest_Sb18.sig” to Alice.

where Alice decrypts “digest_SB18.sig” With Bob’s Public key and decrypts “Enc_SB18.key” with her own Private key which provides Authenticity to Alice.

Then it will compare this digest with SB.key so it will provide the Integrity to Bob.

```
root@yash:/home/yash/Desktop/openssl# openssl dgst -sha256 -sign PrivateKeyOfBob.pem -ou
t digest_SB18.sig SB.key
Enter pass phrase for PrivateKeyOfBob.pem:
root@yash:/home/yash/Desktop/openssl# ls
Dec_SA09.key     Enc_SB18.key         PublicKeyOfBob.pem
digest_SB18.sig  PrivateKeyOfBob.pem  SB.key
root@yash:/home/yash/Desktop/openssl# cat digest_SB18.sig
.zC◆◆◆X◆GG◆>◆3◆◆t◆◆◆◆'f◆i?◆X◆t=p◆z}◆A"k-
yq◆@/`à◆◆◆◆◆ o]◆◆       ◆◆U◆;|%?◆:u(◆◆o\◆◆S◆psL ◆◆◆L◆'◆Z(6◆LtX%J◆}b◆=$◆vY◆GD◆◆!Q*Q◆;◆◆e◆
xI◆◆,Qt-◆!f◆m3root@yash:/home/yash/Desktop/openssl#
```

Sending Bob’s Encrypted File and Signed digest :



Decrypting file :

```
root@yash:/home/yash/Desktop/openssl# openssl pkeyutl -decrypt -inkey PrivateKeyOfBob.pe
m -in ../Enc_SA09.key -out Dec_SA09.key
Enter pass phrase for PrivateKeyOfBob.pem:
root@yash:/home/yash/Desktop/openssl# ls
Dec_SA09.key  PrivateKeyOfBob.pem  PublicKeyOfBob.pem  SB.key
```

```
root@yash:/home/yash/Desktop/openssl# cat Dec_SA09.key
Algorithm = aes-256-cbc
iteration = 200
pass phrase = 12ka4

root@yash:/home/yash/Desktop/openssl#
```

Verifying :

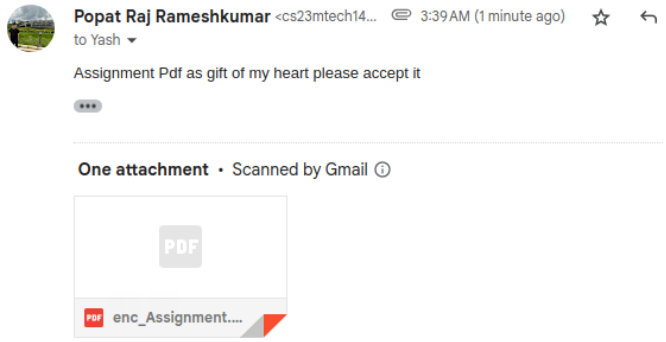
```
root@yash:/home/yash/Desktop/openssl# openssl dgst -sha256 -verify ../PublicKeyOfBob.p
em -signature ../digest_SA09.sig Dec_SA09.key
Verified OK
```

Verified OK!

Verified OK!

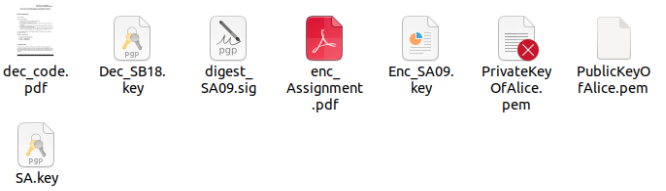
4) Alice Encrypts a PDF with given inputs in SA.key

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl enc -aes-256-cbc -e -iter 200 -s
alt -in ../Assignment.pdf -out enc_Assignment.pdf
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
root@RajPopat:/home/raj/Desktop/NSAS-1# ls
Dec_SB18.key     enc_Assignment.pdf  PrivateKeyOfAlice.pem  SA.key
digest_SA09.sig  Enc_SA09.key        PublicKeyOfAlice.pem
root@RajPopat:/home/raj/Desktop/NSAS-1#
```



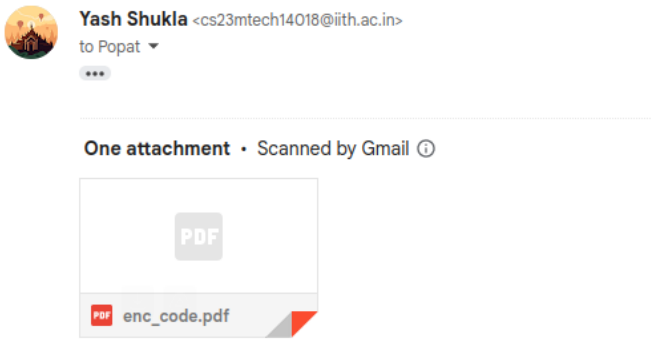
Now Alice got encrypted file from Bob which she will decrypt using inputs in the file SB.key

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl enc -aes-256-cbc -d -iter 300 -i
n ../enc_code.pdf -out dec_code.pdf
enter AES-256-CBC decryption password:
root@RajPopat:/home/raj/Desktop/NSAS-1# ls
dec_code.pdf  digest_SA09.sig     Enc_SA09.key          PublicKeyOfAlice.pem
Dec_SB18.key  enc_Assignment.pdf  PrivateKeyOfAlice.pem  SA.key
root@RajPopat:/home/raj/Desktop/NSAS-1#
```



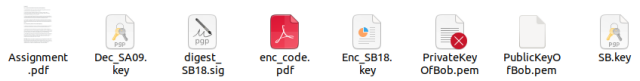
4) Bob Encrypts a PDF with given inputs in SB.key

```
root@yash:/home/yash/Desktop/openssl# openssl enc -aes-256-cbc -e -iter 300 -salt -in ..
/code.pdf -out enc_code.pdf
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
root@yash:/home/yash/Desktop/openssl# ls
Dec_SA09.key     enc_code.pdf  PrivateKeyOfBob.pem  SB.key
digest_SB18.sig  Enc_SB18.key  PublicKeyOfBob.pem
root@yash:/home/yash/Desktop/openssl#
```



Now Bob got encrypted file from Alice which he will decrypt using inputs in the file SA.key

```
root@yash:/home/yash/Desktop/openssl# openssl enc -aes-256-cbc -d -iter 200 -in ../enc_A
ssignment.pdf -out Assignment.pdf
enter AES-256-CBC decryption password:
root@yash:/home/yash/Desktop/openssl# ls
Assignment.pdf  digest_SB18.sig  Enc_SB18.key         PublicKeyOfBob.pem
Dec_SA09.key    enc_code.pdf     PrivateKeyOfBob.pem  SB.key
root@yash:/home/yash/Desktop/openssl#
```

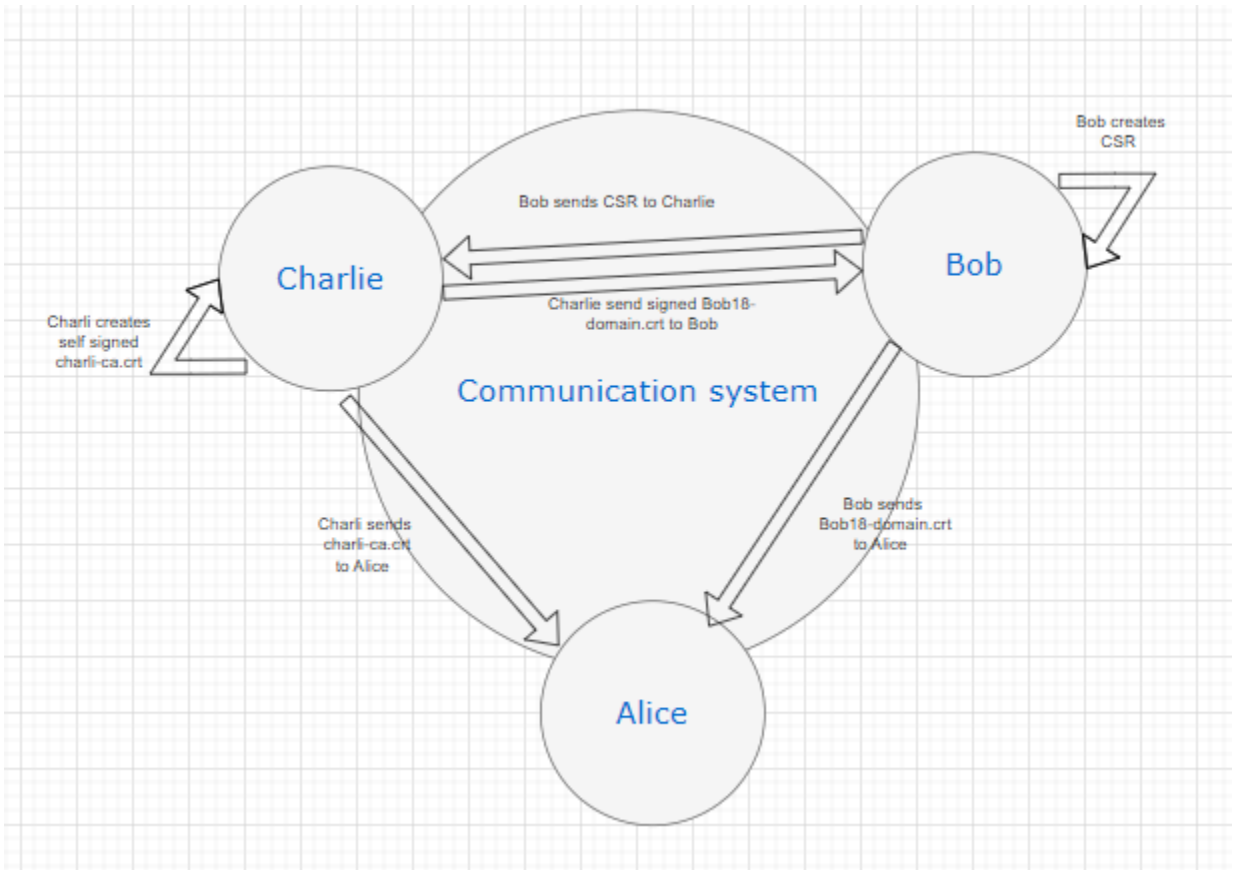


PART - B :

In Part B of the assignment,

We created a secure communication system with three participants :
Bob, the web server,
Alice, the client,
Charlie, who serves as the root Certificate Authority.

Flow Diagram



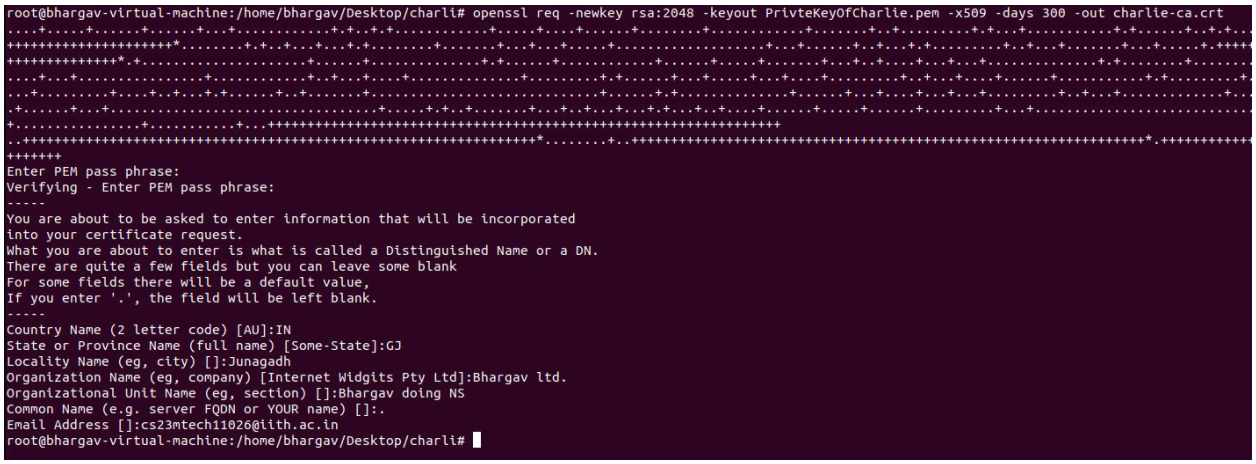
Charlie became the root CA at the beginning of the process by using RSA (a key of 2048 bits) to create a **self-signed X.509 certificate** which is valid for 300 days.

Now Bob creates a **Certificate Signing Request** (CSR).
Bob, the web server, sends CSR to Charlie and Charlie verifies the Bob CSR(details) and signs with its root CA private key. Charlie sends Bob18-domain.crt to Bob.

This allowed Bob to receive an **end-user certificate**. Alice used the root CA certificate (charlie-ca.crt) to **validate** this certificate, bob-domain.crt.

In the public key infrastructure, the verification procedure guaranteed Bob's certificate's **integrity and authenticity**.

Step1 : Charlie : Charlie Generates a Self-signed Certificate



```
root@bhargav-virtual-machine:/home/bhargav/Desktop/charli# cat charlie-ca.crt
-----BEGIN CERTIFICATE-----
MIID9TCCAt2gAwIBAgIUfs1SihlatysGAnvI2O0c3WewAvwwDQYJKoZIhvcNAQEL
BQAwgYkxCzAJBgNVBAYTAklOMQswCQYDVQQIDAJHSjERMA8GA1UEBwwISnVuYWdh
ZGgxFTATBgNVBAoMDEJoYXJnYXYgbHRkLjEZMBcGA1UECwwQQmhhcmdhdiBkb2lu
ZyBOUzEoMCYGCSqGSIb3DQEJARYZY3MyM210ZWNoMTEwMjZAaWl0aC5hYy5pbjAe
Fw0yNDAxMjUwODQ1NTJaFw0yNDExMjAwODQ1NTJaMIGJMQswCQYDVQQGEwJJTjEL
MAkGA1UECAwCR0oxETAPBgNVBAcMCEp1bmFnYWRoMRUwEwYDVQQKDAxCaGFyZ2F2
IGx0ZC4xGTAXBgNVBAsMEEJoYXJnYXYgZG9pbmcgTlMxKDAmBgkqhkiG9w0BCQEW
GWNzMjNtdGVjaDExMDI2QGlpdGguYWMuaW4wggEiMA0GCSqGSIb3DQEBAQUAA4IB
DwAwggEKAoIBAQCR/GZgblIB1C82tzsGuVuLFCZBST4uSlYbYmQTVcuzasIXXoxs
fJ0scWYz0iCPvL/0p0fpp0NHd5ADNUHnFg7imo1c+tOoVoulke9v8TnR631wFx1Z
LQ85t2kP7y/ClaF0rBO0eR0mGWJu/JhLv36znQZapCcys6w/dmsI5QgvY7x9+8jd
hfpljTl31842bhLBaKQ9QkFhxzg9tvsJ9ko3K47xaH+epnVhIldvHaHmcIbzBjtq
IphGwLGpptf2UAW79klagPd2c9cfvaitMIOOnyXCqb9qHKz8Bt5Lm8kgDKsq+uTR
EDiuFJtJ98yL5c/dkQeL1oKLZzhT3qdSsTTbAgMBAAGjUzBRMB0GA1UdDgQWBBQV
NFKn9veUZ5MW0VJE2RSSeTIFKzAfBgNVHSMEGDAWgBQVWFKn9veUZ5MW0VJE2RSS
eTIFKzAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBVyIfbNzke
3/H7/Brzr61oj7OjVTDkRhG7L63U5LCxADgKoUPRlZUPvWICtaQuRxDDwwmgZPhF
qxL5Cgq/6XmWXh9p3f14TQ9KjN/UN20Ezon+x6hS57SmJWk4AWaTjPlfjx7lUm3i
2rio0MpmnLkTkf78i4gBQ9Pus0U5Mtd2zokoq3MbACZe5YvsOsGZV8HNnI0mCrLA
H39ig6GAoy5JwupvcuVtIqDWuxOrBqXCAGHbUHciTLGJ08FNwO72frpg64zPKYfF
6e2kFqrqF8+vYIl98XCWUK59lWO692AFJIucRs8a20s8o0jZzoReApUZSNtWI0oB
7mQZnDRYLlrw
-----END CERTIFICATE-----
```

Step 2: Bob : Bob Generates a CSR.

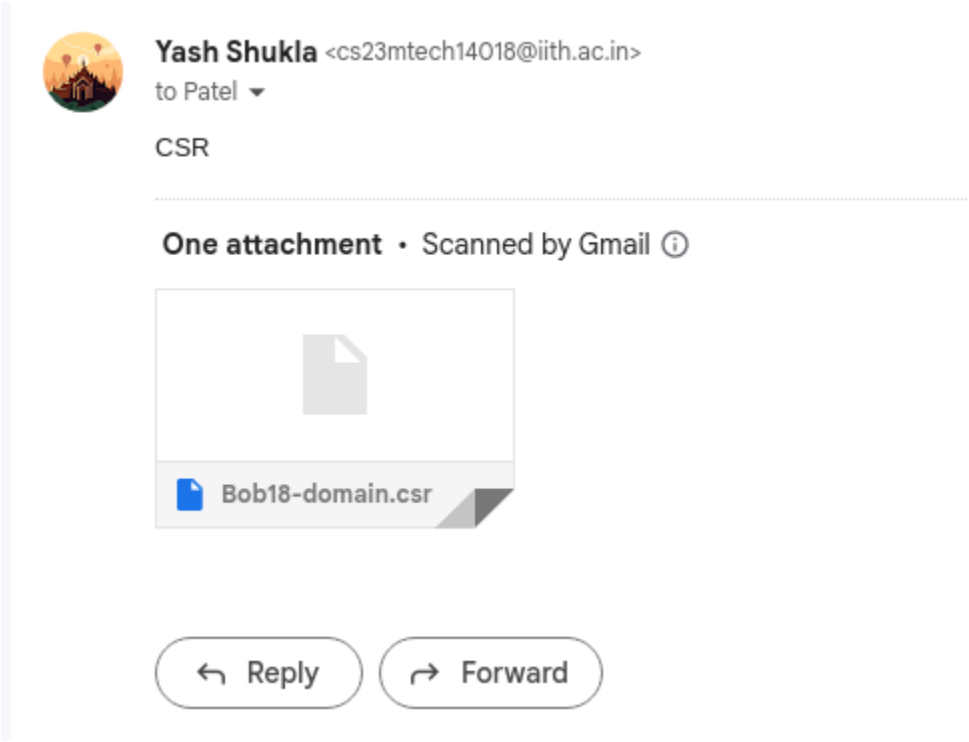
```
root@yash:/home/yash/Desktop/openssl# openssl req -key PrivateKeyOfBob.pem -new -passin pass:1234 -out Bob18-domain.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
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]:IN
State or Province Name (full name) [Some-State]:GJ
Locality Name (eg, city) []:ABD
Organization Name (eg, company) [Internet Widgits Pty Ltd]:yash ltd
Organizational Unit Name (eg, section) []:bob doing NS
Common Name (e.g. server FQDN or YOUR name) []:bob
Email Address []:cs23mtech14018@iith.ac.in

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234
An optional company name []:.
```

Step 3 : Viewing CSR :

```
root@yash:/home/yash/Desktop/openssl# cat Bob18-domain.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIC5TCCAc0CAQAwgYoxCzAJBgNVBAYTAklOMQswCQYDVQQIDAJHSjEMMAoGA1UE
BwwDQUJEMREwDwYDVQQKDAh5YXNoIGx0ZDEVMBMGA1UECwwMYm9iIGRvaW5nIE5T
MQwwCgYDVQQDDANib2IxKDAmBgkqhkiG9w0BCQEWGWNzMjNtdGVjaDE0MDE4QGlp
dGguYWMuaW4wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCY70v/bwUj
MvpAQI3WLKdIRd6cjR6zjhSsHQ8FpEHEDw/K7KFEpL52CD88Pr/p+PdVnW+XtaGm
XTiBLpvHdbTkaXJ56Rl1vJX+FyVKxAsveJZ0NhlPyJ420n5jS/h6kz58QdrhmtQ2
UIG4jFpWsGiPIAmGvmHE6/GG6QPYIUAMZZlK4TFgLK9yY2ozvBs+HW+7cuXKqWf6
5cqMPZIN8UfzdHoyGL2ub2W+TOfzwgMg3yFG/hVLu+PqsuvJhMLbtlbASKl1Qo7O
t03EbLxC4Q7CeddMDQo9z+g5ENVHMNuWxXcFaE60UZDSYiDnOJdCWkAINaV0nSjg
EMRfDWWux+tFAgMBAAGgFTATBgkqhkiG9w0BCQcxBgwEMTIzNDANBgkqhkiG9w0B
AQsFAAOCAQEAgeUOgFRiWMFpB1iUh3PU66XTvWX3MDifAeUTjK8b02VsSIf8TlFJ
lh64NpiY7VaT6YOY+T3oSv0U8deLM6DeEwWyFUnkDiML5nU/dUXPdBQ5a+79AvC0
FocLcVV3xC/UQK15489o1AKk3zHWWllIjPD3HZiity3/7tL8yFq0SknRDJg/8DnH
FGDpe1BlYrPbuMpCn1wsUxtVhkvM/lSTtpH6opDVba7+pTCT4XRuxjjBChsmOnIG
nGTBbIYj05hc3inAqvp60xnnCvAoFXLbzUVUM1FuW57l7xKjfbW7A3Qwsw14IARF
0uZNGJyiBucq+zeigsGJYuHjF7hBeZg5DA==
-----END CERTIFICATE REQUEST-----
```

Step 4 : Sending CSR to Charlie :



Step 5 : Charlie :Signing Certificate for Bob18-domain.csr

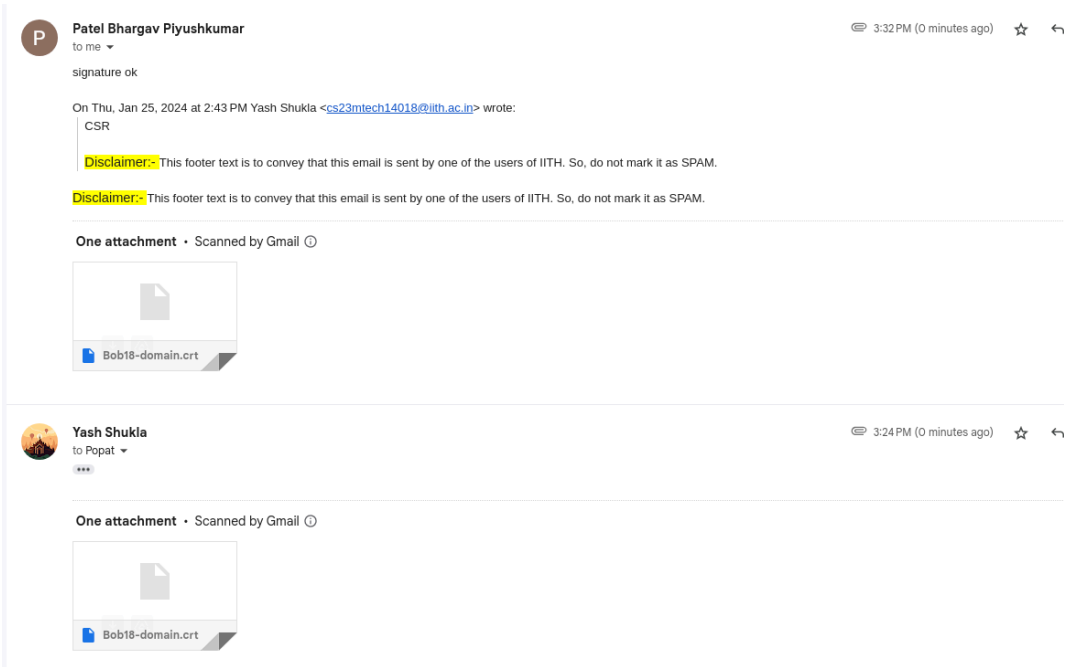
```
root@bhargav-virtual-machine:/home/bhargav/Desktop/charli# openssl x509 -req -sha256 -days 100 -in ../Bob18-domain.csr -CAkey PrivteKeyOfCharlie.pem -passin pass:1234 -CA charlie-ca.crt -out Bob18-domain.
crt -CAcreateserial
Certificate request self-signature ok
subject=C = IN, ST = GJ, L = ABD, O = yash ltd, OU = bob doing NS, CN = bob, emailAddress = cs23mtech14018@iith.ac.in
```

Self Signature OK.

Step 6 : Viewing Certificate :

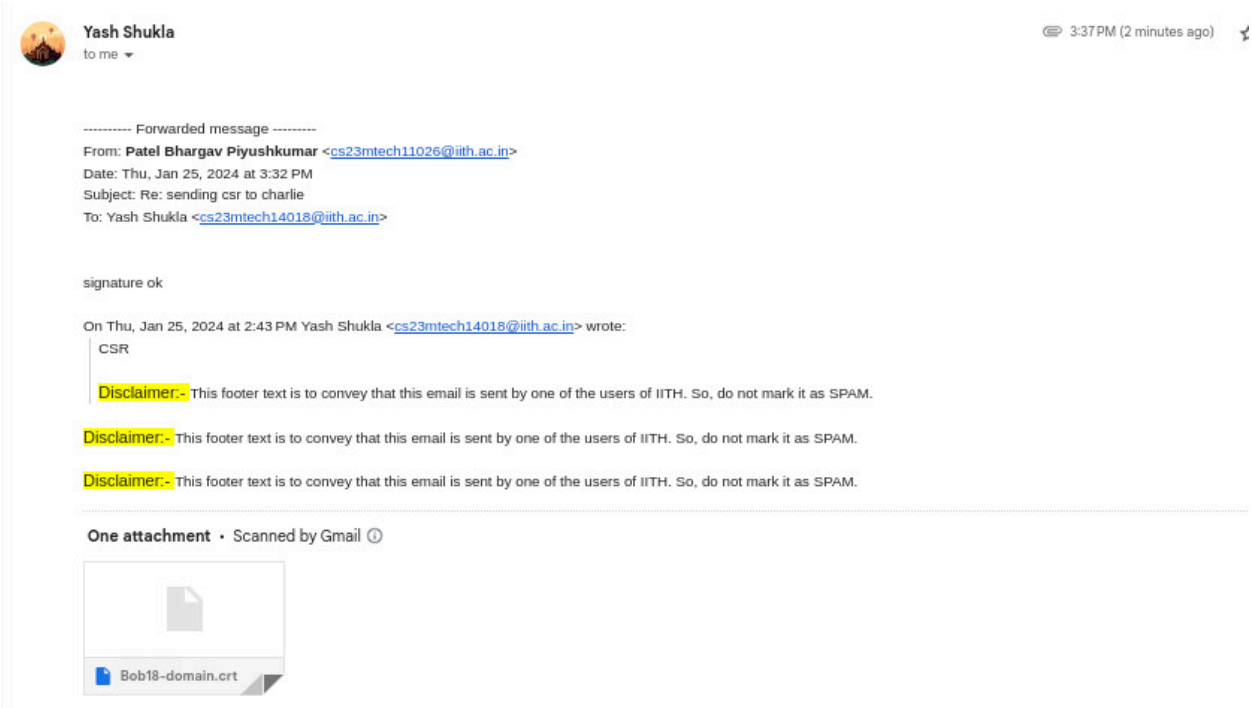
```
root@bhargav-virtual-machine:/home/bhargav/Desktop/charli# ls
Bob18-domain.crt  charlie-ca.crt  PrivteKeyOfCharlie.pem
root@bhargav-virtual-machine:/home/bhargav/Desktop/charli# cat Bob18-domain.crt
-----BEGIN CERTIFICATE-----
MIIDnDCCAoQCFB47tN8mOocwr4VvCfvPpvHX0XWhMA0GCSqGSIb3DQEBCwUAMIGJ
MQswCQYDVQQGEwJJTjELMAkGA1UECAwCR0oxETAPBgNVBAcMCEp1bmFnYWRoMRUw
EwYDVQQKDAxCaGFyZ2F2IGx0ZC4xGTAXBgNVBAsMEEJoYXJnYXYgZG9pbmcgTlMx
KDAmBgkqhkiG9w0BCQEWGWNzMjNtdGVjaDExMDI2QGlpdGguYWMuaW4wHhcNMjQw
MTI1MDkyNTU3WhcNMjQwNTA0MDkyNTU3WjCBijELMAkGA1UEBhMCSU4xCzAJBgNV
BAgMAkdKMQwwCgYDVQQHDANBQkQxETAPBgNVBAoMCHlhc2ggbHRkMRUwEwYDVQQL
DAxib2IgZG9pbmcgTlMxDDAKBgNVBAMMA2JvYjEoMCYGCSqGSIb3DQEJARYZY3My
M210ZWNoMTQwMThAaWl0aC5hYy5pbjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAJjvS/9vBSMy+kBAjdYsp0hF3pyNHrOOFKwdDwWkQcQPD8rsoUSkvnYI
Pzw+v+n491Wdb5e1oaZdOIEum8d1tORpcnnpGXW8lf4XJUrECy94lnQ2GU/InjbS
fmNL+HqTPnxB2uGa1DZQgbiMWlawaI8gCYa+YcTr8YbpA9ghQAxlmUrhMWAsr3Jj
ajO8Gz4db7ty5cqpZ/rlyow9kg3xR/N0ejIYva5vZb5M5/PCAyDfIUb+FUu74+qy
68mEwtu2VsBIqXVCjs63TcRsvELhDsJ510wNCj3P6DkQ1Ucw25bFdwVoTrRRkNJi
IOc4l0JaQAg1pXSdKOAQxF8NZa7H60UCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEA
foJGe7ruOHfSf6dQxmL4PcZETGgaj3ug+CbrPfI9zGacmkUn4XRMRTHsaubJmrJi
hvrtHVXbYUxKKG185L+2QXN6+bpNxnXJKGEz+j0zkuGXxv3NG7SsIhx5T0Bnhpz3
B2VyVNnw4B2dc4jsvPpnJAe0qkmln1hSkQetfpoLAzDK/D2q5Ktbyw79N6rsTfPA
4Tcgirb6Pkjf8QbKdyHLY+Qv+Gh0mR8eYGpbqNFFbdH8CvJMxhicn2Ug8x8qLmLz
60fnlNfEAqPt8wUTUvyuILYSmkKE2ORkArwOZziwIo32OcY9drCmcZoSXuUqENBt
8boau1UxiS0qxhQbrs7Z3g==
-----END CERTIFICATE-----
```

Step 7 : Sending Bob18-domain.crt Certificate to Bob :



And also sending Charlie Self Sign Certificate to Alice :

Also here Bob sending Bob18-domain.crt to Alice



Step 8 : Alice verifying whether Bob's certificate is valid or not:

Alice got both Bob18-domain.crt and charlie-ca.crt.

```
root@RajPopat:/home/raj/Desktop/NSAS-1# openssl verify -verbose -CAfile ../charl
ie-ca.crt ../Bob18-domain.crt
../Bob18-domain.crt: OK
```

Thank You