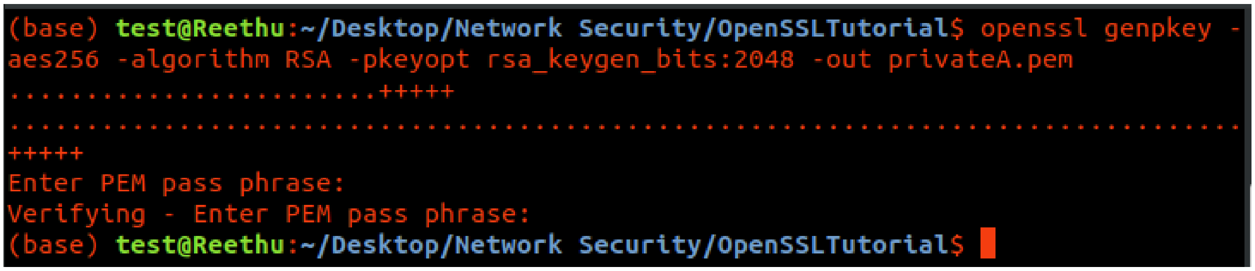
AKASH TADWAI, ES18BTECH11019
VINTA REETHU, ES18BTECH11028

Part A: Secure file transfer between Alice28 (Reethu) and Bob19 (Akash)

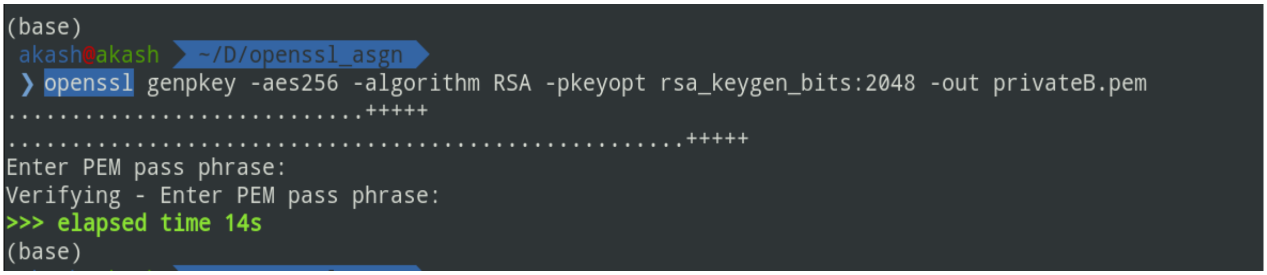
1. Create RSA (2048) key pairs for Alice28 and Bob19 and exchange public keys over email. Password protect your respective private keys

- Alice28 and Bob19 create password protected private keys using the following command.
  $ `openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out privateA.pem` (Alice28)



```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl genpkey -
aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out privateA.pem
........................+++++
................................................................................
+++++
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$
```

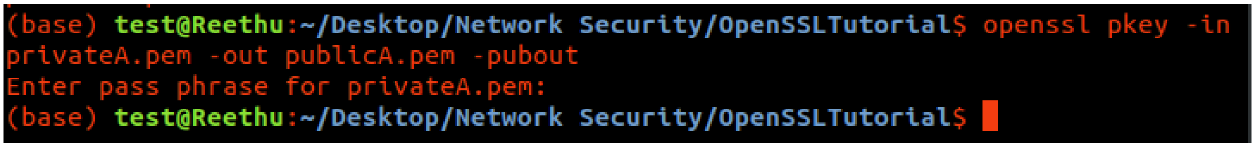
  $ `openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out privateB.pem` (Bob19)



```
(base)
 akash@akash  ~/D/openssl_asgn
 ❯ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out privateB.pem
............................+++++
.........................................................+++++
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
>>> elapsed time 14s
(base)
```

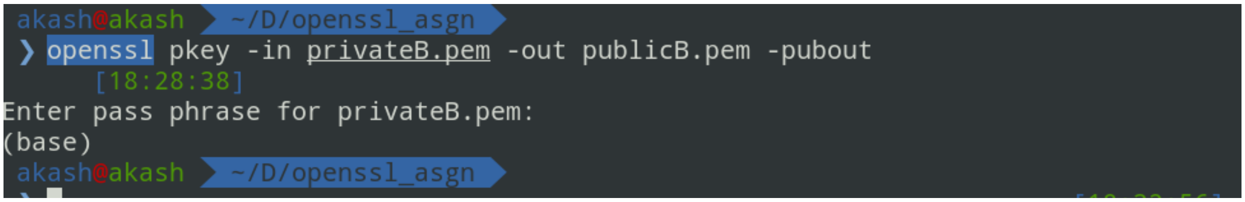
- Public keys can be extracted from the private keys by using the following command.

  $ `openssl pkey -in privateA.pem -out publicA.pem -pubout` (Alice28)



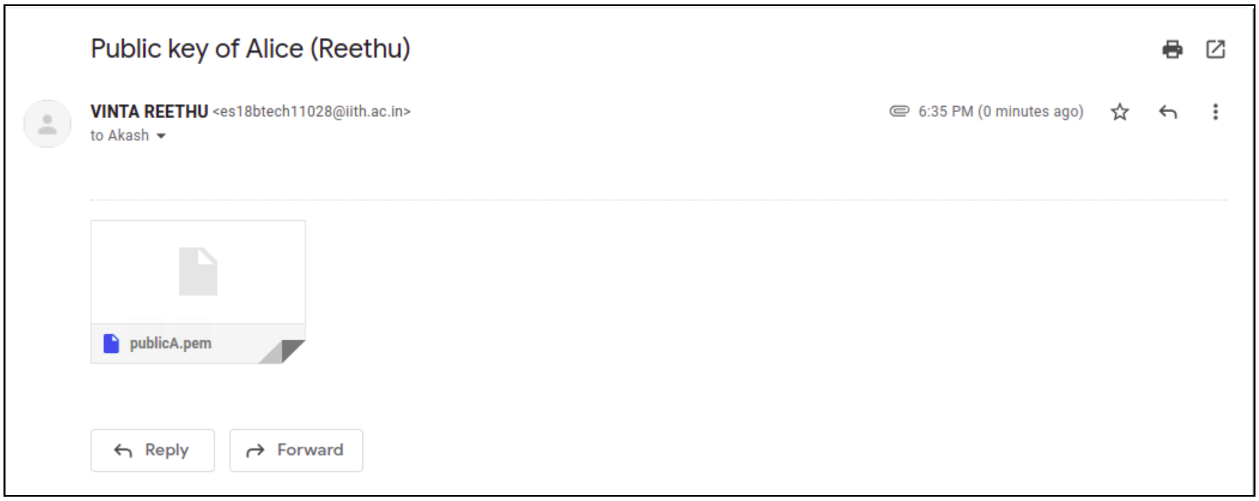
```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl pkey -in
privateA.pem -out publicA.pem -pubout
Enter pass phrase for privateA.pem:
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$
```

  $ `openssl pkey -in privateB.pem -out publicB.pem -pubout` (Bob19)



```
 akash@akash  ~/D/openssl_asgn 
 ❯ openssl pkey -in privateB.pem -out publicB.pem -pubout
      [18:28:38]
Enter pass phrase for privateB.pem:
(base)
 akash@akash  ~/D/openssl_asgn 
```

- Alice28 and Bob19 send Public keys over e-mail

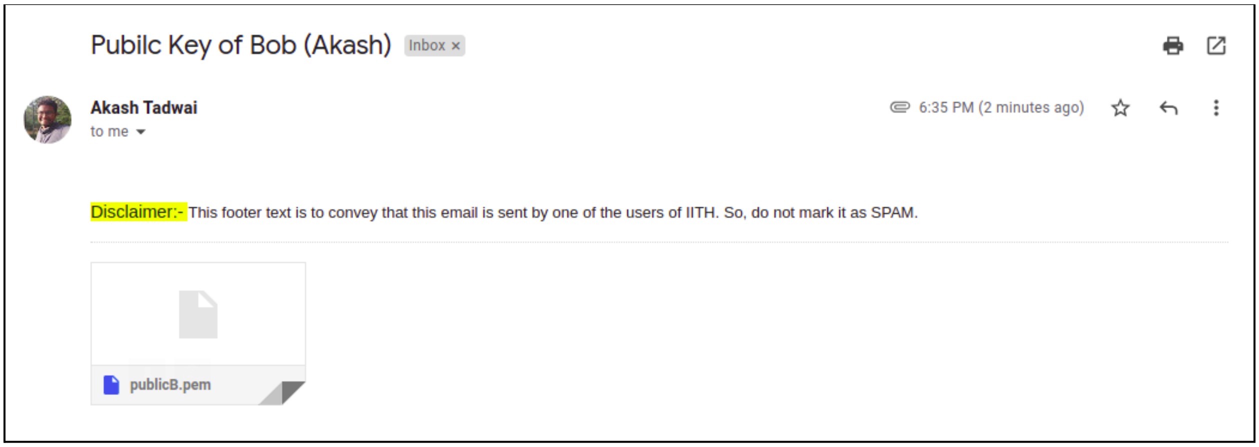


Public key of Alice (Reethu)

VINTA REETHU <es18btech11028@iith.ac.in>
to Akash

6:35 PM (0 minutes ago)

publicA.pem



Pubilc Key of Bob (Akash) Inbox ×

Akash Tadwai
to me

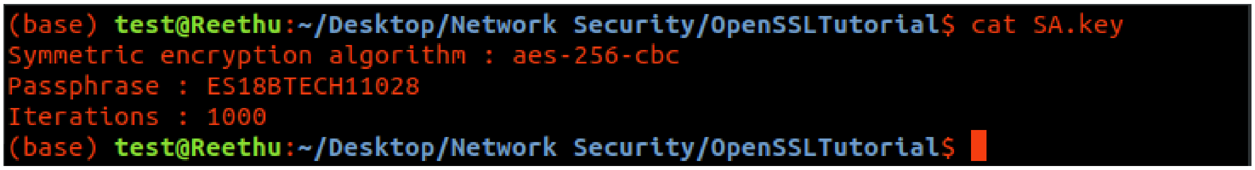
6:35 PM (2 minutes ago)

Disclaimer:- This footer text is to convey that this email is sent by one of the users of IITH. So, do not mark it as SPAM.

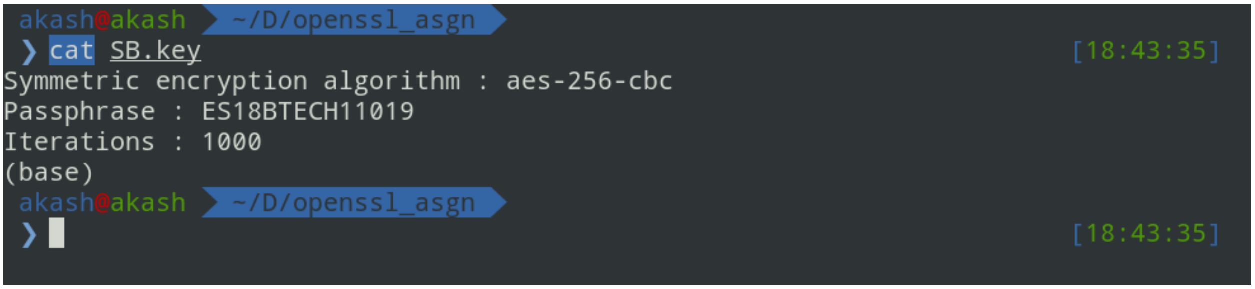
publicB.pem

2. Alice28 creates a text file named SA.key with this info <symmetric encryption algo, its parameters, and passphrase>. Bob also does the same thing (SB.key). These serve like keys for decrypting files exchanged in each way.

- SA.key, SB.key files are created by adding
  - Symmetric encryption Algorithm
  - Passphrase
  - Iterations
- SA.key, SB.key files look like



```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ cat SA.key
Symmetric encryption algorithm : aes-256-cbc
Passphrase : ES18BTECH11028
Iterations : 1000
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ 
```



```
 akash@akash  ~/D/openssl_asgn 
 ❯ cat SB.key                                                   [18:43:35]
Symmetric encryption algorithm : aes-256-cbc
Passphrase : ES18BTECH11019
Iterations : 1000
(base)
 akash@akash  ~/D/openssl_asgn 
 ❯                                                              [18:43:35]
```

3. Alice28 has to securely send SA.key to Bob. Devise a mechanism in such a way that only Bob19 can see that message and verify it indeed came from Alice28 without any tampering. Similarly, Bob19 has to securely send his SB.key to Alice28 and prove its authenticity and integrity.

- Both Alice28 and Bob19 exchange their respective symmetric keys in the following way,
  - **Alice28:**

    (encrypt SA.key with Bob's public key)

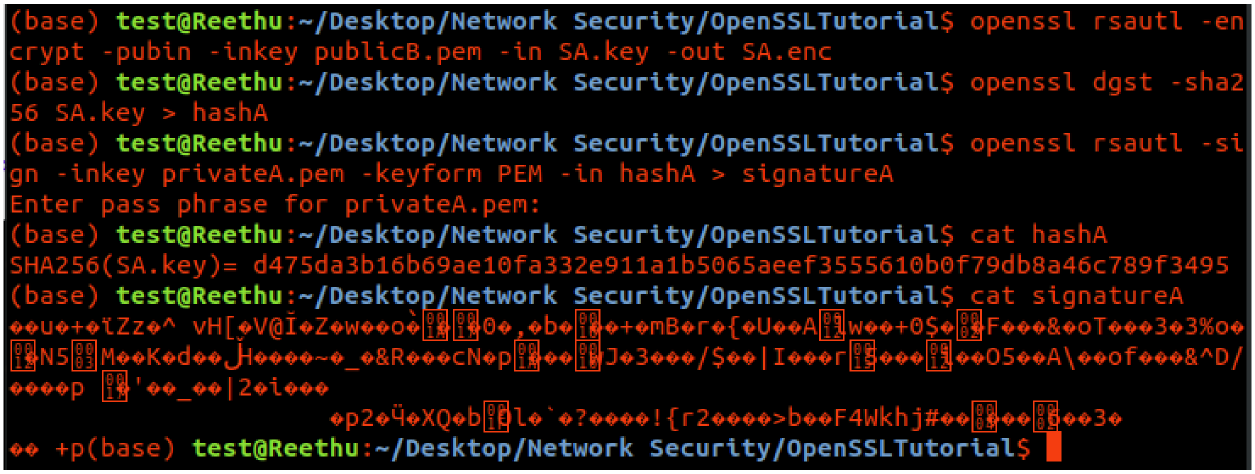
    ```
    $ openssl rsautl -encrypt -pubin -inkey publicB.pem -in SA.key -out SA.enc
    ```

    (generate digest of SA.key)

    ```
    $ openssl dgst -sha256 SA.key > hashA
    ```

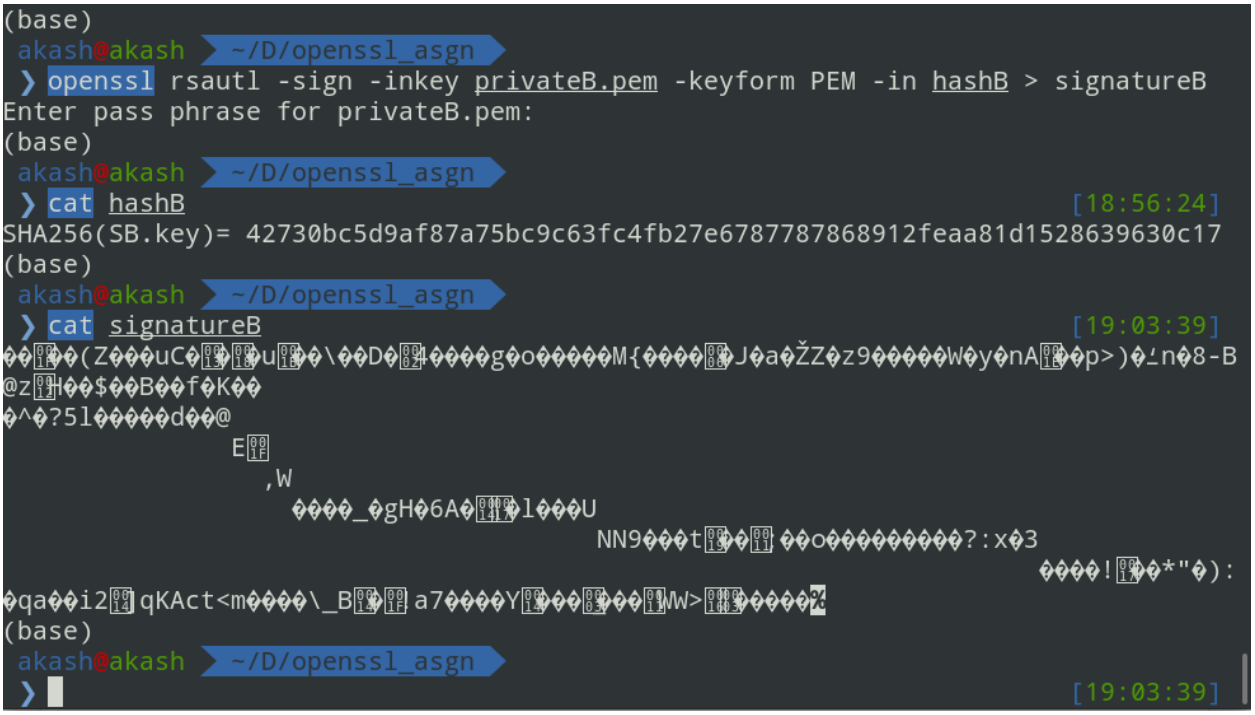
    (sign the digest with Alice's private key)

    ```
    $ openssl rsautl -sign -inkey privateA.pem -keyform PEM -in hashA > signatureA
    ```



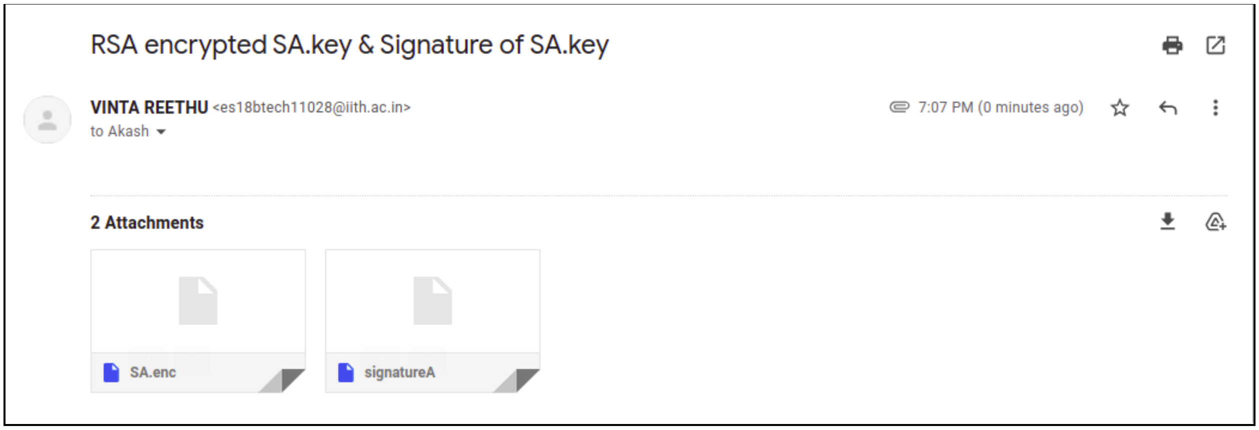
```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl rsautl -encrypt -pubin -inkey publicB.pem -in SA.key -out SA.enc
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl dgst -sha256 SA.key > hashA
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl rsautl -sign -inkey privateA.pem -keyform PEM -in hashA > signatureA
Enter pass phrase for privateA.pem:
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ cat hashA
SHA256(SA.key)= d475da3b16b69ae10fa332e911a1b5065aeef3555610b0f79db8a46c789f3495
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ cat signatureA
��u�+�ïZz�^ vH[�V@Ĭ�Z�w���ò���0�,�b���+�mB�r�{�U��A�w��+0$���F���&�oT���3�3%o�
��N5��M��K�d��ɈH����~�_�&R���cN�p�����J�3���/$��|I���r����������05��A\���of���&^D/
����p ��'��_��|2�i���
                    �p2�Ӵ�XQ�b��l�`�?����!{r2����>b��F4Wkhj#������3�
�� +p(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ 
```

- **Bob19:**
  (encrypt SB.key with Bob’s public key)
  $ `openssl rsautl -encrypt -pubin -inkey publicA.pem -in SB.key -out SB.enc`
  (generate digest of SB.key)
  $ `openssl dgst -sha256 SB.key > hashB`
  (sign the digest with Bob’’s private key)
  $ `openssl rsautl -sign -inkey privateB.pem -keyform PEM -in hashB > signatureB`



```
(base)
 akash@akash  ~/D/openssl_asgn 
 ❯ openssl rsautl -sign -inkey privateB.pem -keyform PEM -in hashB > signatureB
Enter pass phrase for privateB.pem:
(base)
 akash@akash  ~/D/openssl_asgn 
 ❯ cat hashB                                                              [18:56:24]
SHA256(SB.key)= 42730bc5d9af87a75bc9c63fc4fb27e6787787868912feaa81d1528639630c17
(base)
 akash@akash  ~/D/openssl_asgn 
 ❯ cat signatureB                                                         [19:03:39]
������(Z���uC�����u���\��D������g�o������M{���������J�a�ŽZ�z9�����W�y�nA����p>)�٤n�8-B
@z��H��$��B��f�K��
�^�?5l�����d���@
               E�
                 ,W
                   ����_�gH�6A����l���U
                                       NN9���t���� ��o����������?:x�3
                                                                     ����!���*"�):
�qa��i2� qKAct<m����\_B���� a7����Y�������������W>�������%
(base)
 akash@akash  ~/D/openssl_asgn 
 ❯                                                                        [19:03:39]
```

- Alice28 and Bob19 send encrypted RSA & it’s signature



RSA encrypted SA.key & Signature of SA.key

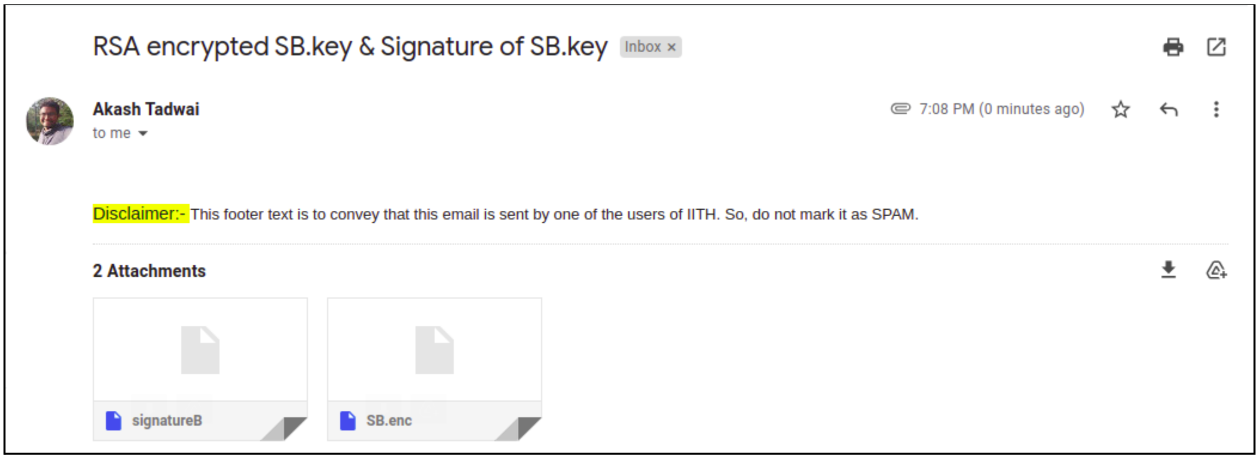
VINTA REETHU <es18btech11028@iith.ac.in>
to Akash

7:07 PM (0 minutes ago)

2 Attachments

SA.enc

signatureA



RSA encrypted SB.key & Signature of SB.key

Akash Tadwai
to me

7:08 PM (0 minutes ago)

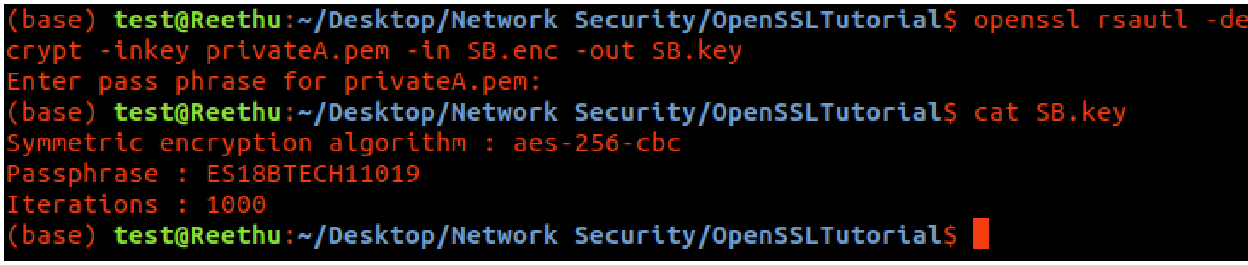
Disclaimer:- This footer text is to convey that this email is sent by one of the users of IITH. So, do not mark it as SPAM.

2 Attachments

signatureB

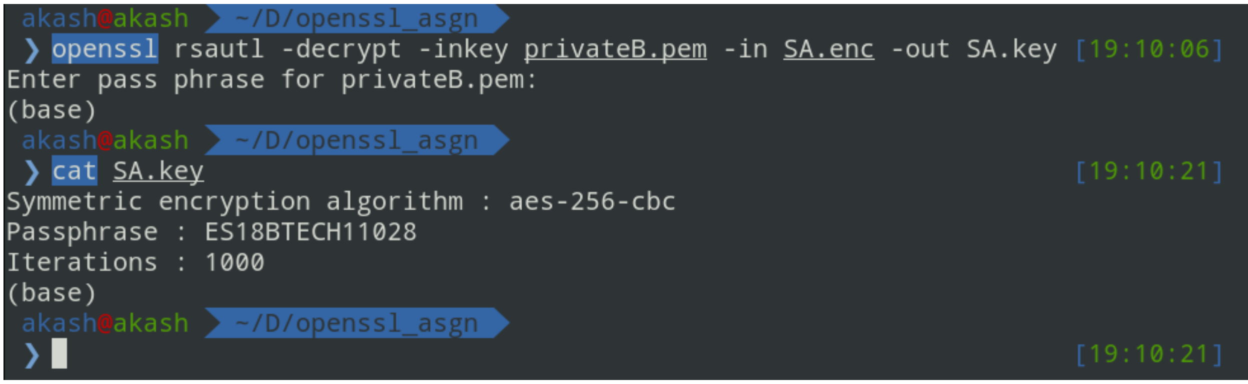
SB.enc

- Alice28 and Bob19 verify if the message came from the other one without any tampering.
  - Alice28 decrypts SB.enc as SB.key using its private key. Bob19 decrypts SA.enc as SA.key using its private key.
    $ `openssl rsautl -decrypt -inkey privateA.pem -in SB.enc -out SB.key`



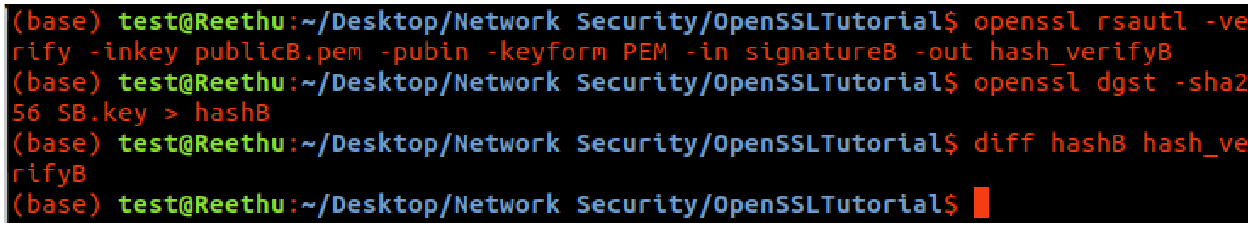
```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl rsautl -de
crypt -inkey privateA.pem -in SB.enc -out SB.key
Enter pass phrase for privateA.pem:
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ cat SB.key
Symmetric encryption algorithm : aes-256-cbc
Passphrase : ES18BTECH11019
Iterations : 1000
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ 
```

$ `openssl rsautl -decrypt -inkey privateB.pem -in SA.enc -out SA.key`



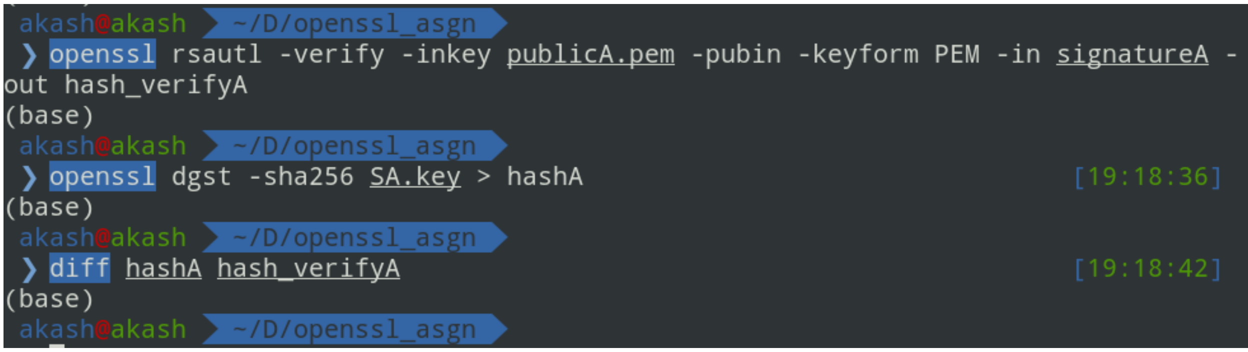
```
 akash@akash  ~/D/openssl_asgn
 ❯ openssl rsautl -decrypt -inkey privateB.pem -in SA.enc -out SA.key [19:10:06]
Enter pass phrase for privateB.pem:
(base)
 akash@akash  ~/D/openssl_asgn
 ❯ cat SA.key                                                          [19:10:21]
Symmetric encryption algorithm : aes-256-cbc
Passphrase : ES18BTECH11028
Iterations : 1000
(base)
 akash@akash  ~/D/openssl_asgn
 ❯                                                                     [19:10:21]
```

- Alice28 decrypts the signature sent by Bob19 with Bob's public key which generates the original SB.key (of Bob's) hash. Alice28 also generates her own hash from her now decrypted SB.key and compares both the hashes using diff command. Bob19 does the same.
  - **Alice28:**
    (decrypt Bob's signature using Bob's public key to get the original hash)
    $ `openssl rsautl -verify -inkey publicB.pem -pubin -keyform PEM -in signatureB -out hash_verifyB`
    (generate digest from the decrypted SB.key file sent by Bob)
    $ `openssl dgst -sha256 SB.key > hashB`
    (compare hashes using diff command)
    $ `diff hashB hash_verifyB`



```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl rsautl -ve
rify -inkey publicB.pem -pubin -keyform PEM -in signatureB -out hash_verifyB
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl dgst -sha2
56 SB.key > hashB
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ diff hashB hash_ve
rifyB
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ 
```

  - **Bob19:**
    (decrypt Alice's signature using Alice's public key to get the original hash)
    $ `openssl rsautl -verify -inkey publicA.pem -pubin -keyform PEM -in signatureA -out hash_verifyA`
    (generate digest from the decrypted SB.key file sent by Bob)
    $ `openssl dgst -sha256 SA.key > hashA`
    (compare hashes using diff command)
    $ `diff hashA hash_verifyA`



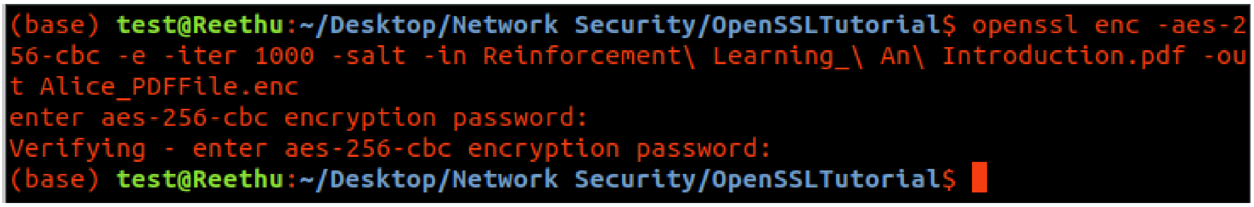
```
 akash@akash  ~/D/openssl_asgn
 ❯ openssl rsautl -verify -inkey publicA.pem -pubin -keyform PEM -in signatureA -
out hash_verifyA
(base)
 akash@akash  ~/D/openssl_asgn
 ❯ openssl dgst -sha256 SA.key > hashA                              [19:18:36]
(base)
 akash@akash  ~/D/openssl_asgn
 ❯ diff hashA hash_verifyA                                          [19:18:42]
(base)
 akash@akash  ~/D/openssl_asgn
```

- As the both hashes are the same we have proved the authenticity and integrity of the messages. Now Alice28 and Bob19 are ready to exchange any large files through their desired symmetric encryption.

4. Alice28 encrypts a large file (some PDF/Photo) with SA.key and sends it along with a signature to Bob19 so that he could decrypt it with the same SA.key and verify it indeed came from Alice28 without tampering. Similarly, Bob19 should send some large file securely to Alice28 without any tampering.

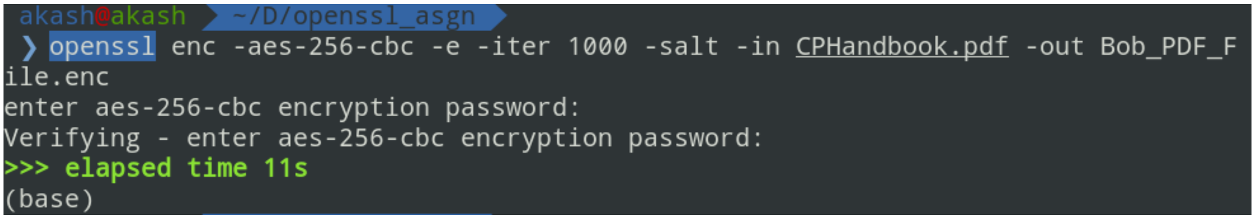
- Alice28 will send “Reinforcement Learning_ An Introduction.pdf”. Bob19 will send “CPHandbook.pdf”
- Alice28 encrypts her pdf file as Alice_PDFFile.enc using the details as mentioned in SA.key sent to Bob19. Alice28 encrypts her pdf file as Bob_PDFFile.enc using the details as mentioned in SB.key sent to Alice28.

```
$ openssl enc -aes-256-cbc -e -iter 1000 -salt -in Reinforcement\
Learning_\ An\ Introduction.pdf -out Alice_PDFFile.enc
```



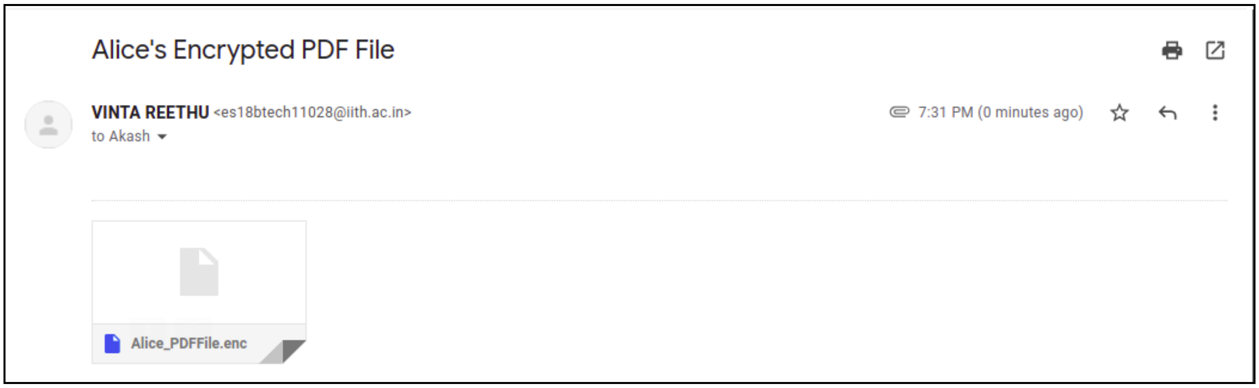
```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl enc -aes-2
56-cbc -e -iter 1000 -salt -in Reinforcement\ Learning_\ An\ Introduction.pdf -ou
t Alice_PDFFile.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ 
```

```
$ openssl enc -aes-256-cbc -e -iter 1000 -salt -in CPHandbook.pdf -out
Bob_PDF_File.enc
```



```
 akash@akash  ~/D/openssl_asgn
 ❯ openssl enc -aes-256-cbc -e -iter 1000 -salt -in CPHandbook.pdf -out Bob_PDF_F
ile.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
>>> elapsed time 11s
(base)
```

- Alice28 and Bob19 share encrypted pdf files over e-mail.

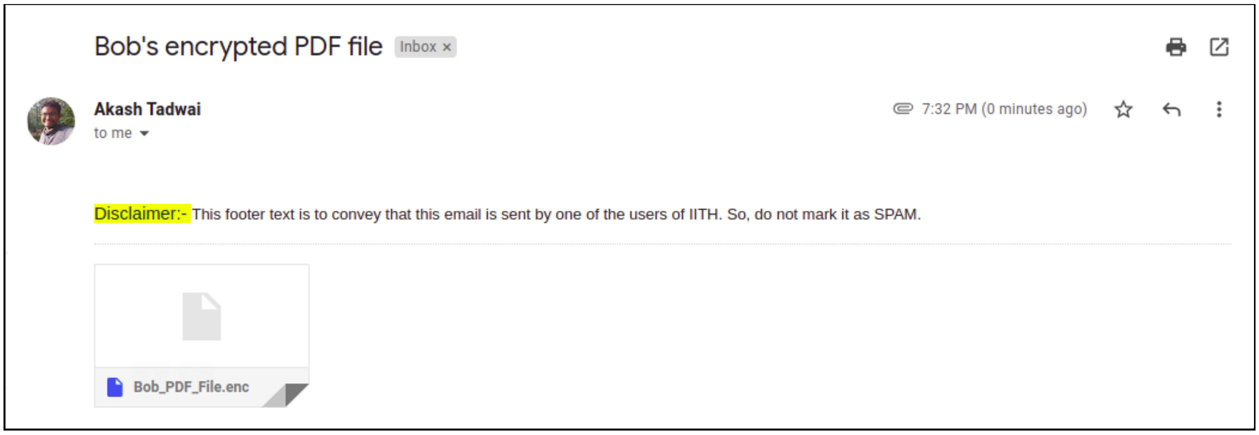


Alice's Encrypted PDF File

VINTA REETHU <es18btech11028@iith.ac.in>
to Akash

7:31 PM (0 minutes ago)

Alice_PDFFile.enc



Bob's encrypted PDF file Inbox ×

Akash Tadwai
to me

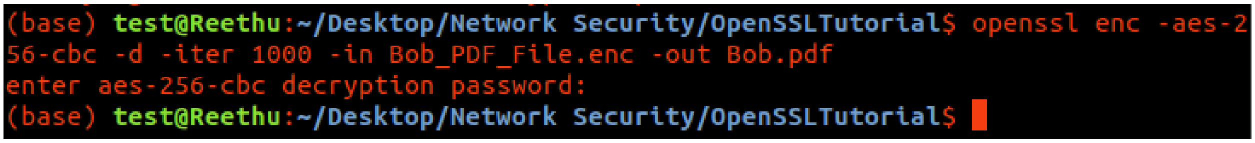
7:32 PM (0 minutes ago)

Disclaimer:- This footer text is to convey that this email is sent by one of the users of IITH. So, do not mark it as SPAM.

Bob_PDF_File.enc

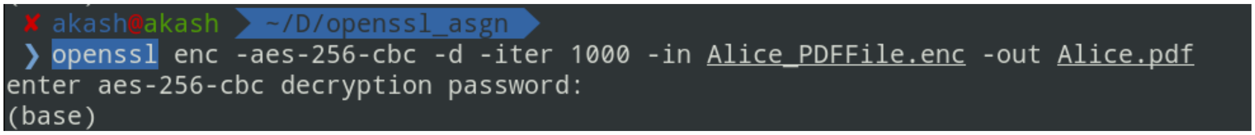
- Alice28 decrypts Bob_PDF_File.enc as Bob.pdf. Bob19 decrypts Alice_PDFFile.enc as Alice.pdf

```
$ openssl enc -aes-256-cbc -d -iter 1000 -in Bob_PDF_File.enc -out Bob.pdf
```



```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl enc -aes-256-cbc -d -iter 1000 -in Bob_PDF_File.enc -out Bob.pdf
enter aes-256-cbc decryption password:
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ 
```

```
$ openssl enc -aes-256-cbc -d -iter 1000 -in Alice_PDFFile.enc -out Alice.pdf
```



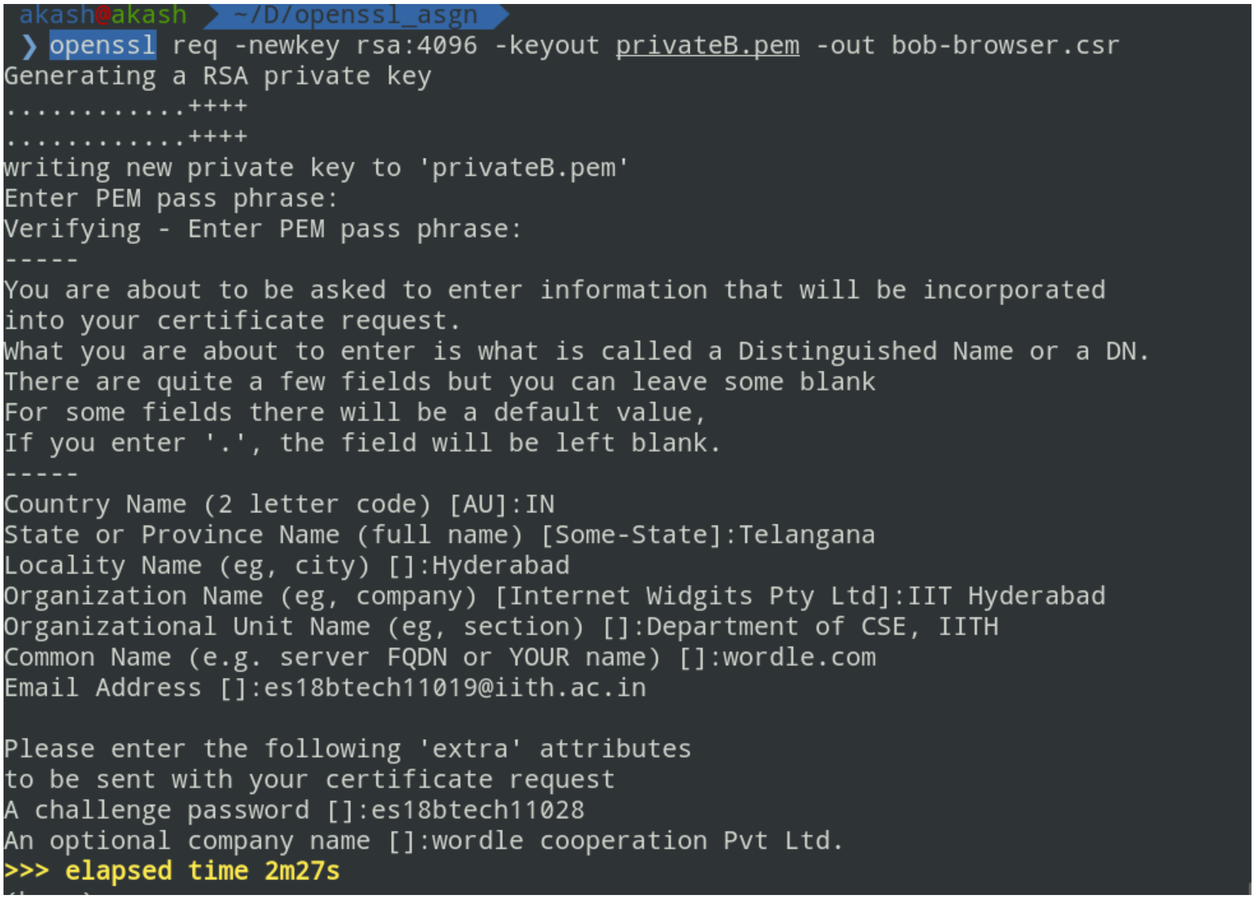
```
 ✘ akash@akash  ~/D/openssl_asgn 
 ❯ openssl enc -aes-256-cbc -d -iter 1000 -in Alice_PDFFile.enc -out Alice.pdf
enter aes-256-cbc decryption password:
(base)
```

- Alice28 and Bob19 have securely shared their respective PDF’s.

Part B: Alice (Browser), Bob (web server) and Charlie (Root CA)

1. Charlie’s (one of the TAs of this course) certificate is generated by the TA.

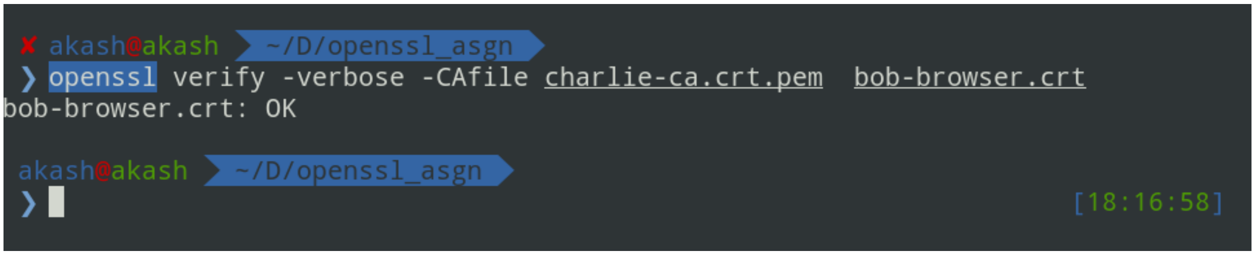
2. Bob generates CSR named bob-browser.csr and emails it to Charlie for providing end-user cert named bob-browser.crt
    - Bob generates CSR with the following command.
      $ `openssl req -newkey rsa:4096 -keyout privateB.pem -out bob-browser.csr`



```
 akash@akash  ~/D/openssl_asgn
❯ openssl req -newkey rsa:4096 -keyout privateB.pem -out bob-browser.csr
Generating a RSA private key
............++++
............++++
writing new private key to 'privateB.pem'
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]:IN
State or Province Name (full name) [Some-State]:Telangana
Locality Name (eg, city) []:Hyderabad
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IIT Hyderabad
Organizational Unit Name (eg, section) []:Department of CSE, IITH
Common Name (e.g. server FQDN or YOUR name) []:wordle.com
Email Address []:es18btech11019@iith.ac.in

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:es18btech11028
An optional company name []:wordle cooperation Pvt Ltd.
>>> elapsed time 2m27s
```

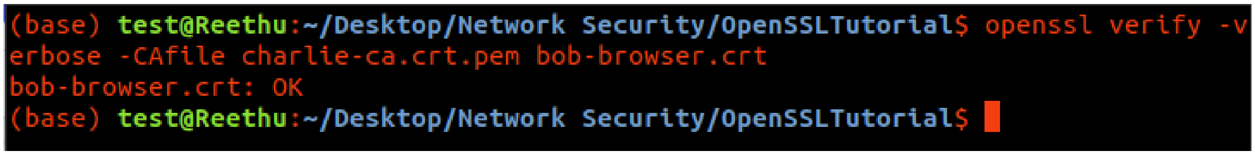
- Bob verifies bob-browser.crt is valid and is indeed signed by the root CA, Charlie.
  $ `openssl verify -verbose -CAfile charlie-ca.crt.pem bob-browser.crt`



```
 ✘ akash@akash  ~/D/openssl_asgn
❯ openssl verify -verbose -CAfile charlie-ca.crt.pem  bob-browser.crt
bob-browser.crt: OK

 akash@akash  ~/D/openssl_asgn
❯                                                                  [18:16:58]
```

3. Alice (Student A) gets charlie-ca.crt over email from Charlie and bob-browser.crt over email from Bob and verifies that Bob's certificate is valid and signed by the root CA, Charlie.
   $ `openssl verify -verbose -CAfile charlie-ca.crt.pem bob-browser.crt`



```
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$ openssl verify -v
erbose -CAfile charlie-ca.crt.pem bob-browser.crt
bob-browser.crt: OK
(base) test@Reethu:~/Desktop/Network Security/OpenSSLTutorial$
```

- **Bob's certificate:**
  - It is of type X.509 V3.
  - **Serial Number assigned:** 6D 23 87 12 5A 29 1D 63 F0 07 AE 48 ED EB 4D 51 D0 BE 8B 73
  - **Key Usages:** Digital signature, Key encipherment
  - **Crtitical:** No
  - **Certificate Authority:** No
  - **Max Path Length:** Unlimited

Deliverables:

- **openssl x509 -in <cert-name> -text**
  - **Charlie-ca.crt**

```
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+LdG7rS2cH8wDQYJKoZIhvcNA
QEL
BQAwgYcxCzAJBgNVBAYTAklOMRIwEAYDVQQIDAlUZWxhbmdhbmExEzARB
gNVBAcM
ClNhbmdhcmVkZHkxDTALBgNVBAoMBElJVEgxDDAKBgNVBAsMA0NTRTEQ
MA4GA1UE
AwwHUm9vdF9DQTEgMB4GCSqGSIb3DQEJARYRY2hhcmxpZUBlbWFpbC5jb
20wHhcN
MjIwMjAxMTkzMzU4WhcNNDIwMTI3MTkzMzU4WjCBhzELMAkGA1UEBhMCS
U4xEjAQ
BgNVBAgMCVRlbGFuZ2FuYTETMBEGA1UEBwwKU2FuZ2FyZWRkeTENMAs
GA1UECgwE
SUlUSDEMMAoGA1UECwwDQ1NFMRAwDgYDVQQDDAdSb290X0NBMSAwH
gYJKoZIhvcN
AQkBFhFjaGFybGllQGVtYWlsLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIP
ADCC
AgoCggIBAKi4Jjq0juVRZixwtFOtS+9zfrPtI7Wh1KaZFrRo+r7V6EtF8o5q7k7q
```

```
ewkLxPnBttMjiiL63XUosiC3BsAI2u47gFyH5Pmwo7pKIhdzRwW3O3hre2DUYOK
v
Dety0Qr/rNSui6AuNvIKDwof7IkGJx2aUWXq8m+2poC9nrc5IIpZHsdvBh7jcNHe
rbmY5y8DaU9xtCUadUv6FckgCERAGRrbnWPIuhIjpDV48P+AZu95sk8zGkDSTt
3f
PE+J3iEpF0I+Hb5XDFpHO2GpU5N8STFw5X6LA3O4F8kLB9B8Pt9HuEBRgzDf
WAbO
3iYnOE7nuBaQq17DOO/CuDEOSJaGZztZUDO4KMgcEDVRDBI5PT+X6IhskCG
W4y/T
CUxIUmj4zfAKG8IQc5V2wEHexAZKFKjkmsUnm2mcUhhaEOnrGgb1+osTIcUh0
Lct
WvTg06vhszZyYQyj7hjSZxrFUkdZbsvw+nMfz1fYDMFPrIo2VwnU3+eDsz2YIiCh
DCVjVOdtOEs3CCObG10oaKrGCXVHGZ/gTBGPBTpXc1nEmom7F5AXp4/ON
U1D4zEs
vxoTI/d7BMMb7G99DYSGkuzPraW0i1K6A7M3t+sInEEWZMCq8DVEhGEZy3bLj
gTg
8PgKEg+e693CUbrb2OnW5Maq0SmwRxNFY0gw440wpREX076Ni6/ZAgMBAA
GjUzBR
MB0GA1UdDgQWBBQIZtnihWuLSMEMmB8LVLgIhSX2bzAfBgNVHSMEGDAW
gBQIZtni
hWuLSMEMmB8LVLgIhSX2bzAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3
DQEBCwUA
A4ICAQBspY2iwzQp2N19wa8o8ABtdhqACsUCTb+izNY5gmQ9Sf+BgL6IbEb5X
BTg
XG4ZfRIO0hMqug8M5K5vcSD2I7DYr457m5YBR59PMIkses5pCjkB5JK5mGcCD
7UB
ZQu3z3iQwkjDWvEKz0WSh4tI0I1vTbmQ9kwj3KM8Yg0wMRvpid8UtHYTfb67eh
Db
dCZo1mp4+Ia7aSY4zNckS2iD/xf6ifKQG4mPx7tSI9gqcnISMIxwISWpxWZWIN1K
cwdqp9Nb9oiZxXvmcxTKkQsRQbJjZWFwubbPxIaQnIB1uHUpR0cT7A9Re8/6Q
dkQ
1FZyQuuL1jBu3wx3km4xCMGXZ1Pse4yGy8mMWeh71IHIPujbaIgdORb46ztCR
PfK
U0ZHsErvJvZ7kN+/KceOpxXsQW1To3PGCjbVW9GYUbkITRP3eZCF5uIQ26Ri
KaiX
/FMsOR1s05xi3RvP8gI9rQzr/NBfnOiByxwbboFILIHhg46X+Xgx82CS7fOYkbd3
pp65ZWfo4/eILS/MW768B7PInOzI7eUmQZI1XmQBCaEKYhRVw59rNT3IWXmK
r35m
AFaxW/PpxmsFMWz6Hncp005sJ7iRUyKk1buWsUvIwoIxhI6TbhcUqwp29dj+ND
7N
SVIRtjQLfoM6eO9IGJ++XQWzJT8E4qWNSxx7choImHxZAGHuOA==
-----END CERTIFICATE-----
```

- **bob-browser.crt**
  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:73
        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 06:58:26 2022 GMT
            Not After : Feb  4 06:58:26 2032 GMT
        Subject: C = IN, ST = Telangana, L = Hyderabad, O = IIT Hyderabad, OU =
"Department of CSE, IIT Hyderabad", CN = wordle.com, emailAddress =
es18btech11019@iith.ac.in
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:bc:5f:75:74:24:62:c8:48:dd:56:45:f3:32:fd:
                    5b:c4:07:9b:67:9b:94:2b:ab:4e:3a:07:17:17:c1:
                    8c:25:98:d2:30:0a:58:74:36:13:dc:ca:04:cb:96:
                    7c:a0:d5:25:6c:b4:50:01:8c:65:d3:f9:62:78:9f:
                    55:e7:b5:45:7f:f2:6e:fc:7f:41:24:53:97:9c:57:
                    30:32:d7:b1:c2:bd:de:c5:af:c3:d1:80:37:1d:11:
                    a7:f1:94:73:34:79:39:6b:8c:96:96:61:03:f2:14:
                    2a:97:82:18:a2:e3:f8:b9:fd:1d:d2:a0:c1:5d:d4:
                    61:69:4c:be:19:b0:aa:82:69:87:d8:ec:76:12:cb:
                    19:b1:c5:1c:56:66:ed:05:a3:09:fc:20:01:77:bc:
                    c8:31:10:6e:70:a2:96:ae:ca:26:9a:b8:8a:2e:45:
                    85:29:15:ef:73:2a:32:59:c1:55:4f:5a:96:75:47:
                    c6:9c:a0:cd:7f:96:2b:4a:a5:ac:8e:9d:c4:8c:fb:
                    57:49:44:d8:35:97:c0:5e:81:7e:07:64:a8:df:0b:
                    fb:e1:0f:be:ec:4c:8c:08:f4:4b:c5:d8:6f:e5:8f:
                    5b:ad:c0:10:30:ec:32:55:4a:c5:1b:e6:57:15:0c:
                    b8:a9:bd:9b:63:32:81:9a:e1:87:a2:d9:02:6a:a8:
                    43:5e:84:2d:1d:de:e4:0a:59:76:b7:1b:1a:a7:4a:
                    d1:e6:7e:5c:4c:ef:d6:c7:cf:6e:78:17:d1:84:24:
                    4c:5e:fe:18:50:1f:da:ca:38:29:de:f7:e4:66:52:
                    fe:86:68:7d:48:27:e3:7b:64:01:e6:b9:38:ae:5d:
                    ee:5b:8c:0f:82:c0:ac:d1:b8:d5:8d:e8:65:f1:3e:
                    74:06:cb:10:b5:e3:ca:e1:d0:69:47:ed:a9:5b:80:
                    b9:19:85:21:8c:27:58:bc:04:2d:52:8c:d3:83:ea:
                    8b:82:79:31:7a:1f:63:a3:8c:26:ba:83:cf:26:1f:
                    8b:34:38:fa:24:6e:4f:95:ec:af:8b:bf:02:54:3c:
                    f8:57:3d:b9:12:02:6e:6d:67:f3:d5:38:f9:00:73:
```

```
                    b5:f2:03:73:de:14:ee:0b:a3:7c:e2:ac:51:46:0b:
                    7d:b5:fb:4d:ea:12:0b:1a:44:09:38:b4:4d:ea:48:
                    e0:29:32:6f:51:58:22:34:50:1f:a9:22:90:44:7b:
                    91:53:4a:46:24:ad:5b:a4:a3:20:e1:5b:ae:3d:6f:
                    01:3c:e1:cb:b1:1a:a0:38:f3:3d:90:4f:71:0d:76:
                    97:f3:0a:34:dc:af:37:df:45:d6:c4:f4:1e:6f:db:
                    aa:14:74:4c:1b:f0:56:6a:64:c8:4d:95:a0:df:05:
                    40:fc:df
                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
         28:c9:83:6b:82:ce:c1:42:12:dd:29:8a:41:a8:e1:b8:3a:bb:
         d9:0d:bd:d2:95:a1:a9:87:23:ae:e5:1b:61:48:8e:cd:91:77:
         bf:bc:15:25:0a:ab:f9:2e:95:fc:49:d7:2e:a2:3e:19:8f:21:
         74:10:83:2a:3f:11:e0:fe:93:79:3d:da:00:f8:f1:a6:af:91:
         b7:44:b4:4f:66:d5:db:5b:a7:22:0f:f2:92:ca:3f:bf:af:65:
         ab:cc:4b:1f:e4:73:6c:74:e7:05:d5:28:86:e8:69:de:cd:bc:
         10:d0:fa:7a:61:8b:26:0e:dc:98:c1:38:49:2a:2d:3c:36:39:
         05:bd:24:3e:c4:94:e1:dc:73:2c:e1:8e:f6:ee:43:0a:ed:cb:
         9c:b5:f4:06:db:06:1b:09:e1:d6:e8:7f:0c:58:df:69:7a:a5:
         6d:00:96:ee:bf:5c:34:36:7e:6d:78:26:96:f5:9e:c8:b6:d1:
         2f:4f:14:b6:2c:cb:30:bd:45:ae:cd:87:af:14:29:07:7e:96:
         a8:6f:50:46:ad:1c:1e:07:33:f8:87:c5:7f:a0:a8:7a:aa:16:
         93:bb:2e:2e:5f:fe:fd:b2:e5:54:18:6c:7e:2b:38:a7:db:9c:
         86:43:ac:3d:99:ac:78:f0:65:83:f3:b3:f1:09:03:d6:0b:a8:
         6f:62:2a:2d:b4:e9:9c:63:74:e0:aa:5e:d1:1a:38:7f:bd:63:
         20:fa:6a:c7:12:a5:7e:4a:ba:f5:32:13:8c:a5:3b:18:3c:ff:
         87:ea:2a:aa:d1:9a:87:1b:e9:cb:c2:30:46:5b:58:1e:c7:1c:
         15:7e:cc:d4:44:f6:01:14:78:e4:e8:34:35:24:70:1a:2b:65:
         bc:a9:88:a8:34:7d:02:2d:9a:29:45:e0:2d:c6:a9:f9:ab:c4:
         ae:3b:2b:4e:c2:88:8a:45:c8:0b:6c:3e:4c:67:9b:77:cf:6b:
         f7:8f:a9:3e:bc:2b:c0:56:74:97:a0:77:d0:55:b1:40:49:44:
         c2:1e:d9:10:bb:9c:9e:86:82:b7:2d:d8:04:27:1a:da:36:8d:
         51:5b:b7:52:08:79:20:c1:0d:ce:3a:72:d5:09:60:73:36:5c:
         49:0d:13:1f:a0:2c:ca:e6:b3:ff:88:ec:c4:47:1b:b3:80:70:
         d6:78:87:55:7a:c7:27:dd:a9:7b:37:aa:d9:45:48:a1:5a:b5:
         bd:59:e1:24:12:43:d7:cf:59:ba:a1:fe:31:a0:68:24:c2:50:
```

```
        f4:a6:6c:18:47:56:7a:91:f4:1c:f8:39:d7:03:9e:e2:02:8f:
        75:59:ba:bf:88:b5:d5:41:ef:53:88:ca:d6:7e:96:fd:2e:43:
        cb:8b:8e:89:d2:36:87:bf
```

-----BEGIN CERTIFICATE-----
MIIGGzCCBAOgAwIBAgIUbSOHEIopHWPwB65I7etNUdC+i3MwDQYJKoZIhvcN
AQEL
BQAwgYcxCzAJBgNVBAYTAklOMRIwEAYDVQQIDAlUZWxhbmdhbmExEzARB
gNVBAcM
ClNhbmdhcmVkZHkxDTALBgNVBAoMBElJVEgxDDAKBgNVBAsMA0NTRTEQ
MA4GA1UE
AwwHUm9vdF9DQTEgMB4GCSqGSIb3DQEJARYRY2hhcmxpZUBlbWFpbC5jb
20wHhcN
MjIwMjA2MDY1ODI2WhcNMzIwMjA0MDY1ODI2WjCBtzELMAkGA1UEBhMCSU
4xEjAQ
BgNVBAgMCVRlbGFuZ2FuYTESMBAGA1UEBwwJSHlkZXJhYmFkMRYwFAYD
VQQKDA1J
SVQgSHlkZXJhYmFkMSkwJwYDVQQLDCBEZXBhcnRtZW50IG9mIENTRSwgS
UlUIEh5
ZGVyYWJhZDETMBEGA1UEAwwKd29yZGxlLmNvbTEoMCYGCSqGSIb3DQEJ
ARYZZXMx
OGJ0ZWNoMTEwMTlAaWl0aC5hYy5pbjCCAiIwDQYJKoZIhvcNAQEBBQADggI
PADCC
AgoCggIBALxfdXQkYshI3VZF8zL9W8QHm2ebICurTjoHFxfBjCWY0jAKWHQ2E
9zK
BMuWfKDVJWy0UAGMZdP5YnifVee1RX/ybvx/QSRTl5xXMDLXscK93sWvw9G
ANx0R
p/GUczR5OWuMIpZhA/IUKpeCGKLj+Ln9HdKgwV3UYWIMvhmwqoJph9jsdhLLG
bHF
HFZm7QWjCfwgAXe8yDEQbnCiIq7KJpq4ii5FhSkV73MqMInBVU9aInVHxpygzX
+W
K0qIrI6dxIz7V0IE2DWXwF6BfgdkqN8L++EPvuxMjAj0S8XYb+WPW63AEDDsMI
VK
xRvmVxUMuKm9m2MygZrhh6LZAmqoQ16ELR3e5ApZdrcbGqdK0eZ+XEzv1sf
PbngX
0YQkTF7+GFAf2so4Kd735GZS/oZofUgn43tkAea5OK5d7IuMD4LArNG41Y3oZf
E+
dAbLELXjyuHQaUftqVuAuRmFIYwnWLwELVKM04Pqi4J5MXofY6OMJrqDzyYfiz
Q4
+iRuT5Xsr4u/AIQ8+Fc9uRICbm1n89U4+QBztfIDc94U7gujfOKsUUYLfbX7TeoS
CxpECTi0TepI4Ckyb1FYIjRQH6kikER7kVNKRiStW6SjIOFbrj1vATzhy7EaoDjz
PZBPcQ12I/MKNNyvN99F1sT0Hm/bqhR0TBvwVmpkyE2VoN8FQPzfAgMBAAGj
TTBL
MDEGA1UdJQQqMCgGCCsGAQUFBwMBBggrBgEFBQcDAgYIKwYBBQUHAw
MGCCsGAQUF

```
BwMEMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgXgMA0GCSqGSIb3DQEBC
wUAA4ICAQAo
yYNrgs7BQhLdKYpBqOG4OrvZDb3SlaGphyOu5RthSI7NkXe/vBUICqv5LpX8Sd
cu
oj4ZjyF0EIMqPxHg/pN5PdoA+PGmr5G3RLRPZtXbW6ciD/KSyj+/r2WrzEsf5HNs
dOcF1SiG6GnezbwQ0Pp6YYsmDtyYwThJKi08NjkFvSQ+xJTh3HMs4Y727kMK
7cuc
tfQG2wYbCeHW6H8MWN9peqVtAJbuv1w0Nn5teCaW9Z7IttEvTxS2LMswvUWu
zYev
FCkHfpaob1BGrRweBzP4h8V/oKh6qhaTuy4uX/79suVUGGx+Kzin25yGQ6w9m
ax4
8GWD87PxCQPWC6hvYiottOmcY3TgqI7RGjh/vWMg+mrHEqV+Srr1MhOMpTsY
PP+H
6iqq0ZqHG+nLwjBGW1gexxwVfszURPYBFHjk6DQ1JHAaK2W8qYioNH0CLZop
ReAt
xqn5q8SuOytOwoiKRcgLbD5MZ5t3z2v3j6k+vCvAVnSXoHfQVbFASUTCHtkQu5
ye
hoK3LdgEJxraNo1RW7dSCHkgwQ3OOnLVCWBzNlxJDRMfoCzK5rP/iOzERxuz
gHDW
eIdVescn3aI7N6rZRUihWrW9WeEkEkPXz1m6of4xoGgkwlD0pmwYR1Z6kfQc+D
nX
A57iAo91Wbq/iLXVQe9TiMrWfpb9LkPLi46J0jaHvw==
-----END CERTIFICATE-----
```

- **openssl req -in <csr-name> -text**
    - **Bob (bob-browser.csr):**

```
Certificate Request:
  Data:
    Version: 1 (0x0)
    Subject: C = IN, ST = Telangana, L = Hyderabad, O = IIT Hyderabad, OU =
"Department of CSE, IITH", CN = wordle.com, emailAddress =
es18btech11019@iith.ac.in
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
        RSA Public-Key: (4096 bit)
        Modulus:
          00:cd:7f:fc:40:3f:01:c9:62:84:c0:4b:f4:b2:1f:
          b4:71:93:7f:67:48:2a:e9:60:5a:51:70:c8:52:2f:
          1e:4b:d7:7d:d6:6c:96:da:89:a1:b2:b4:fe:94:ca:
          14:13:0e:86:10:8c:05:3b:25:58:35:5c:2a:92:69:
          36:49:03:40:47:59:fc:ec:b8:82:ba:a6:35:3e:b8:
          22:bc:f1:6a:3a:2c:b5:77:33:b0:6d:49:e7:27:5d:
          48:86:85:6f:e6:d2:07:3d:4f:e8:80:91:6e:6e:9f:
          63:6f:94:16:0f:32:ea:4f:b4:74:ff:cc:29:4f:bf:
                64:a2:0b:d5:44:d3:a4:71:7d:48:30:b5:7f:9d:ca:
                fe:b5:58:f4:56:18:4e:98:bf:b1:28:f2:c3:a2:1c:
                a8:ee:1a:13:21:ba:dc:af:f3:31:54:a9:51:99:12:
                d5:17:a4:67:c0:c2:c0:00:63:c8:b7:3c:16:d7:cc:
                65:ab:87:b5:f7:64:ce:4b:79:b0:83:4d:90:5a:8b:
                43:f2:98:b0:0a:f9:92:f3:89:9b:32:69:f1:81:0c:
                29:5b:70:f0:a1:e4:a3:9b:a2:a4:c0:ae:0f:74:78:
                37:0d:7b:56:13:eb:fc:bd:18:55:a9:fc:82:7e:ae:
                ab:b7:55:4a:f9:74:e6:f7:4d:14:2f:bc:ab:a7:fd:
                2c:3a:6e:93:b1:ce:e2:9e:78:5f:a4:a7:8b:2a:e7:
                2d:46:4c:12:c6:24:c5:39:35:b1:c6:29:c8:83:5c:
                58:60:96:41:9e:9b:60:a8:52:26:0c:4c:ba:e1:79:
                92:56:66:23:ed:d2:21:e0:1b:a9:06:62:f2:23:b5:
                0a:2d:c8:14:08:cb:14:69:ba:b0:06:75:81:17:4e:
                43:d4:67:cc:14:b5:8b:85:12:a7:2d:40:81:e7:c7:
                2b:70:45:f5:3a:22:07:98:76:73:2c:54:2e:b1:bc:
                99:3e:9c:1d:35:55:32:d1:b0:08:0e:8a:55:2e:69:
                2f:ee:76:2d:23:ea:bd:89:08:25:91:04:62:27:5e:
                4b:7e:d3:bf:5f:5e:9d:e3:2a:30:6e:47:da:71:9a:
                2d:62:38:27:fc:52:de:45:5a:ed:3c:7f:d4:8b:41:
                f4:71:35:c9:d7:e0:f8:0c:a3:99:78:fa:80:9a:50:
                a0:b9:5d:46:0d:14:21:dc:0a:33:2c:f1:e8:26:c7:
                e1:e4:be:4c:d4:14:97:ff:f2:22:ff:36:f2:e6:77:
                8f:83:cb:e8:96:84:46:76:70:a5:d3:69:84:db:f3:
                45:17:bb:7e:26:61:fa:2c:30:2d:d3:3b:41:5d:06:
                79:8a:fe:fa:6d:6c:ba:4d:0b:4a:eb:f7:91:fb:2b:
                f7:9c:f3
            Exponent: 65537 (0x10001)
    Attributes:
        challengePassword        :es18btech11028
        unstructuredName         :wordle cooperation Pvt Ltd.
Signature Algorithm: sha256WithRSAEncryption
    51:7c:a8:78:a8:24:3f:d6:7d:9d:24:0c:81:47:4b:9c:17:00:
    95:fa:91:d3:b9:e7:54:2e:40:68:2e:94:93:c6:93:8b:e0:8f:
    e3:db:ca:4d:a3:2e:d8:9b:c2:21:92:ec:89:3f:7b:8f:21:62:
    ce:fa:cb:1a:73:b4:e3:03:1e:c4:72:bc:1b:7e:cf:9f:4d:70:
    c6:1f:5b:2b:f8:92:12:c5:78:90:e0:df:00:94:53:96:e1:8c:
    27:cb:73:76:96:18:a7:9d:e3:fa:50:17:a0:97:1f:01:67:76:
    f1:41:1c:e3:e5:20:6d:d9:a2:ea:d8:d3:7b:f0:d0:9b:f6:a6:
    9b:88:ce:8f:cd:dd:9c:41:21:15:55:63:74:14:6e:96:0c:a8:
    f2:00:21:63:24:92:18:b1:17:1c:58:9a:51:05:5e:03:d8:0f:
    b5:4f:f5:59:fc:89:7f:83:fd:ec:84:16:2c:5c:65:7c:e4:6b:
    e8:ab:d6:dd:03:85:6e:9b:e7:61:69:fb:eb:66:e2:b2:eb:1c:
    c2:1b:2c:a6:ff:90:40:85:a7:11:e8:e9:5f:6c:03:aa:2a:af:
        de:a3:5e:a9:98:8e:69:87:5c:ad:01:74:cf:40:7f:f9:a4:78:
        d5:64:be:b5:89:11:e0:6b:d6:6f:15:ed:e7:20:91:12:93:44:
        5a:bf:f1:ce:62:59:1a:ef:87:14:2b:cb:97:6f:d5:f8:41:4f:
        fb:08:75:62:c3:ea:d5:5e:c7:b8:65:df:3c:a7:aa:ca:d3:81:
        0b:66:cd:e3:a1:be:99:19:71:78:e4:6b:41:ec:02:fb:56:8d:
        2a:c4:3f:df:fb:67:c1:92:a8:5a:60:bd:c2:eb:15:3c:04:a0:
        76:fd:f1:e0:b6:1b:16:2c:ef:75:6e:10:42:1c:68:8e:2e:71:
        d7:13:f6:1d:0e:bc:ca:dc:93:33:2f:e1:52:75:3c:80:6c:83:
        ea:97:fc:b6:5f:bc:47:e8:5a:ea:85:9d:36:a1:b1:28:24:f9:
        10:22:03:b6:2f:2d:3e:d7:34:85:0f:46:c6:6f:e3:7f:58:57:
        5b:89:58:2b:91:c5:a7:1a:84:0e:55:63:8a:2e:6b:8e:50:e3:
        f8:ac:ce:f4:8c:3a:b9:73:b7:f3:99:c2:30:a7:8f:6a:82:16:
        59:39:48:4d:b7:92:72:57:53:f3:19:f0:15:b3:8c:b1:25:02:
        94:8f:6f:f2:41:3e:1a:8c:65:50:d4:3f:45:99:7c:b6:49:57:
        ea:d2:40:89:18:14:22:6a:a3:ec:f7:91:bd:69:4f:45:7f:e2:
        01:c8:16:7d:6e:a6:3a:62:ea:e3:57:e7:40:a3:b8:e7:40:8e:
        97:65:5a:43:0b:b2:05:10
```

-----BEGIN CERTIFICATE REQUEST-----
MIIFPzCCAycCAQAwga4xCzAJBgNVBAYTAklOMRIwEAYDVQQIDAlUZWxhbm
dhbmEx
EjAQBgNVBAcMCUh5ZGVyYWJhZDEWMBQGA1UECgwNSUlUIEh5ZGVyYWJ
hZDEgMB4G
A1UECwwXRGVwYXJ0bWVudCBvZiBDU0UsIElJVEgxEzARBgNVBAMMCndvc
mRsZS5j
b20xKDAmBgkqhkiG9w0BCQEWGWVzMThidGVjaDExMDE5QGlpdGguYWMua
W4wggIi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDNf/xAPwHJYoTAS/SyH7
Rxk39n
SCrpYFpRcMhSLx5L133WbJbaiaGytP6UyhQTDoYQjAU7JVg1XCqSaTZJA0BH
Wfzs
uIK6pjU+uCK88Wo6LLV3M7BtSecnXUiGhW/m0gc9T+iAkW5un2NvlBYPMupPt
HT/
zClPv2SiC9VE06RxfUgwtX+dyv61WPRWGE6Yv7Eo8sOiHKjuGhMhutyv8zFUqV
GZ
EtUXpGfAwsAAY8i3PBbXzGWrh7X3ZM5LebCDTZBai0PymLAK+ZLziZsyafGBD
Clb
cPCh5KOboqTArg90eDcNe1YT6/y9GFWp/IJ+rqu3VUr5dOb3TRQvvKun/Sw6bp
Ox
zuKeeF+kp4sq5y1GTBLGJMU5NbHGKciDXFhglkGem2CoUiYMTLrheZJWZiPt0i
Hg
G6kGYvIjtQotyBQIyxRpurAGdYEXTkPUZ8wUtYuFEqctQIHnxytwRfU6IgeYdnMs
VC6xvJk+nB01VTLRsAgOilUuaS/udi0j6r2JCCWRBGInXkt+079fXp3jKjBuR9px
mi1iOCf8Ut5FWu08f9SLQfRxNcnX4PgMo5l4+oCaUKC5XUYNFCHcCjMs8egmx
+Hk

```
vkzUFJf/8iL/NvLmd4+Dy+iWhEZ2cKXTaYTb80UXu34mYfosMC3TO0FdBnmK/v
pt
bLpNC0rr95H7K/ec8wIDAQABoEswHQYJKoZIhvcNAQkHMRAMDmVzMThidGV
jaDEx
MDI4MCoGCSqGSIb3DQEJAjEdDBt3b3JkbGUgY29vcGVyYXRpb24gUHZ0IEx0
ZC4w
DQYJKoZIhvcNAQELBQADggIBAFF8qHioJD/WfZ0kDIFHS5wXAJX6kdO551Qu
QGgu
lJPGk4vgj+Pbyk2jLtibwiGS7Ik/e48hYs76yxpztOMDHsRyvBt+z59NcMYfWyv4
khLFeJDg3wCUU5bhjCfLc3aWGKed4/pQF6CXHwFndvFBHOPlIG3ZourY03vw0
Jv2
ppuIzo/N3ZxBIRVVY3QUbpYMqPIAIWMkkhixFxxYmlEFXgPYD7VP9Vn8iX+D/e
yE
FixcZXzka+ir1t0DhW6b52Fp++tm4rLrHMIbLKb/kECFpxHo6V9sA6oqr96jXqmY
jmmHXK0BdM9Af/mkeNVkvrWJEeBr1m8V7ecgkRKTRFq/8c5iWRrvhxQry5dv1f
hB
T/sIdWLD6tVex7hI3zynqsrTgQtmzeOhvpkZcXjka0HsAvtWjSrEP9/7Z8GSqFpg
vcLrFTwEoHb98eC2GxYs73VuEEIcaI4ucdcT9h0OvMrckzMv4VJ1PIBsg+qX/LZf
vEfoWuqFnTahsSgk+RAiA7YvLT7XNIUPRsZv439YV1uJWCuRxacahA5VY4oua
45Q
4/iszvSMOrlzt/OZwjCnj2qCFlk5SE23knJXU/MZ8BWzjLEIApSPb/JBPhqMZVDU
P0WZfLZJV+rSQIkYFCJqo+z3kb1pT0V/4gHIFn1upjpi6uNX50CjuOdAjpdlWkML
sgUQ
-----END CERTIFICATE REQUEST-----
```

- **openssl pkey -in <public-key-name> -text -pubin**
  - **Alice (publicA.pem) :**

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0gOUbl4En8z2FSH
m/68q
tT+JzkWBsoIGI9lIJYbH4mBvZD2GXC94aYGok08v/Kh44L/tMG5tcO6NWmH430/
D
vrjjBSntEtf/U8P+b3vf09JeTaM2UCz/x0Wow1QqKeNqZfd6OZ95SX3vsb/HUbvY
rmfoftiU2xATyZXANslBRglPVg9CTxKi+Jd6+MRh19fkY6XBn9XJ7P486O84Nt5T
gttkCyA20vh9nY9L4DENSXdw1D/P8yIWechiYOfp1ogjmMo05XB/F4D2eHXRNkr
K
uUJJQ4dTcCIvguOuoeP0mAIQs6ZgauEW5n1oqrgxGmm1Uyfcdp6kcdTh86f7sz
wy
rwIDAQAB
-----END PUBLIC KEY-----
RSA Public-Key: (2048 bit)
Modulus:
   00:d2:03:94:6e:5e:04:9f:cc:f6:15:21:e6:ff:af:
   2a:b5:3f:89:ce:45:81:b2:89:46:23:d9:65:25:86:
    c7:e2:60:6f:64:3d:86:5c:2f:78:69:81:a8:93:4f:
    2f:fc:a8:78:e0:bf:ed:30:6e:6d:70:ee:8d:5a:61:
    f8:df:4f:c3:be:b8:e3:05:29:ed:12:d7:ff:53:c3:
    fe:6f:7b:df:d3:d2:5e:4d:a3:36:50:2c:ff:c7:45:
    a8:c3:54:2a:29:e3:6a:65:f7:7a:39:9f:79:49:7d:
    ef:b1:bf:c7:51:bb:d8:ae:67:e8:7e:d8:94:db:10:
    13:c9:95:c0:36:c9:41:46:09:4f:56:0f:42:4f:12:
    a2:f8:97:7a:f8:c4:61:d7:d7:e4:63:a5:c1:9f:d5:
    c9:ec:fe:3c:e8:ef:38:36:de:53:82:db:64:0b:20:
    36:d2:f8:7d:9d:8f:4b:e0:31:0d:49:77:70:d4:3f:
    cf:f3:22:16:79:c8:62:60:e7:e9:d6:88:23:98:ca:
    34:e5:70:7f:17:80:f6:78:75:d1:36:4a:ca:b9:42:
    49:43:87:53:70:22:2f:82:e3:ae:a1:e3:f4:98:02:
    10:b3:a6:60:6a:e1:16:e6:7d:68:aa:b8:31:1a:69:
    b5:53:27:dc:76:9e:a4:71:d4:e1:f3:a7:fb:b3:3c:
    32:af
Exponent: 65537 (0x10001)
```

- **Bob (publicB.pem) :**

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuQSrPxzqq3g53eYl08kW
cmSwjDiG58MxZXlckOdQJt1/FC8tKcYgyV05hOo2cwvDrQw+r02ASnasyyVudTMc
1BuElHTE3Qcn1XRpWPQV2V8rUBpoDChmA1EiuZpIKi6/YytmHxZNkYKcgAoBRHBy
Yn/8q9myYkAP+zpVG5p4+7/lFORVVW5iq78vY1Ayx8nzq9g6tMY6mQneyaZbJLjX
ZtG0AOVxJhQPE8dh1eMDtA0KACmR6OHGdxs+57DBLVpyo7swe+wq5xs3K7qUivFI
j8Y2O58T5IXEs+zB5ix548ptkd/FVJGzRLIiHCSgfxgC/Fn8aGJaI8VqXZD+ed2z
GQIDAQAB
-----END PUBLIC KEY-----
RSA Public-Key: (2048 bit)
Modulus:
    00:b9:04:ab:3f:1c:ea:ab:78:39:dd:e6:25:d3:c9:
    16:72:64:b0:8c:38:86:e7:c3:31:65:79:5c:90:e7:
    50:26:dd:7f:14:2f:2d:29:c6:20:c9:5d:39:84:ea:
    36:73:0b:c3:ad:0c:3e:af:4d:80:4a:76:ac:cb:25:
    6e:75:33:1c:d4:1b:84:94:74:c4:dd:07:27:d5:74:
    69:58:f4:15:d9:5f:2b:50:1a:68:0c:28:66:03:51:
    22:b9:9a:48:2a:2e:bf:63:2b:66:1f:16:4d:91:82:
    9c:80:0a:01:44:70:72:62:7f:fc:ab:d9:b2:62:40:
    0f:fb:3a:55:1b:9a:78:fb:bf:e5:14:e4:55:55:6e:
    62:ab:bf:2f:63:50:32:c7:c9:f3:ab:d8:3a:b4:c6:
```

```
    3a:99:09:de:c9:a6:5b:24:b8:d7:66:d1:b4:00:e5:
    71:26:14:0f:13:c7:61:d5:e3:03:b4:0d:0a:00:29:
    91:e8:e1:c6:77:1b:3e:e7:b0:c1:2d:5a:72:a3:bb:
    30:7b:ec:2a:e7:1b:37:2b:ba:94:8a:f1:65:8f:c6:
    36:3b:9f:13:e4:85:c4:b3:ec:c1:e6:2c:79:e3:ca:
    6d:91:df:c5:54:91:b3:44:b9:62:1c:24:a0:7f:18:
    02:fc:59:fc:68:62:5a:97:c5:6a:5d:90:fe:79:dd:
    b3:19
Exponent: 65537 (0x10001)
```

- **cat <encrypted-private-key >**
  - **Alice (encrypted privateA.pem):**

-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIYHjziYtlPYgCAggA
MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBASCMURamtQHQNgL1Izfnq5BIIE
0GFijXRjkf3+JwwzK35zE3cX1sCLzkr4yFsL5MHKG2H6pw+U3CymIMoO3JyLQZpQ
k5bUQsWihexe71GdXMnDgWG3nyQVH2PtPA9b4D1L3j6wQUrkIXyBtErFibSP9YPI
xIJN8rkJNeslhW6E9gseDMEe3t3EQryXUo/ZDxYb5i+RqH1bEjqLIt8fZjfChO2Q
qVzRJ6z3vIaeTf6fhRqLRh5GOyYu4uqp8vJ88QYXaR1hAVupGmFXgUgpl3/Ftr2g
eyo/rezEGw6wGchW3PXJs137cRap0y/k0cqJYc1XmTIP9U4Y10qgFCHPSmDhwyRb
J9fgkIOTCUEnybQDvoU4dI6KKmAIOnrrS3rrLyxEjSdabuhycxh6PeyCb2/YobiC
IHWun+jsYde0mDdqTqE0LQiB1dM2a287yZ9Pwpkw/yOoU0wkcWvn7uUzJNIpF1Qo
ym/UDnt50NDU2uy4/yLBF2500pGIbwPBROb+gh9tM5jpAwAcTtYevgeqFLxRLZiJ
mAtFuKop/KQEpH4hSNpTTYBHQj4CTM2WKLL3W1etemm5Q8NyHzJtRpFspswG1ktu
q21NhWK6CIjBLXX3YRfHG031ieLtj3Aj6BmLj0p2ZXUXuV42DIC8Swbjcs970saO
zFFpKF84GW/8nNZX/KIVuWqHWPuHra3wGK32rywjkoAnn2LygJFWd6/WqS6RV3xI
hSepWEhyOxT0SKeByzG649oJIxUsZuAFIgODgZ06askkff+RWNwVszUuNWfBNQjL
yIFav+6y38vQa+x0JUc8fLzWjeH6eyl4L0gaUpyLSrJfmvOLYgVms1PmLylQtWi/
me61F32m0r3lfGMxXMKlw4QOpSuOiZtykeIgSRjOLVXOJof6lTO3ESb9y0xPLUWP
aLSmud0QbiNF10gSqwpnWQCYH+8ZBaZgY1Chnf5fTXICCI7RhKGXP02TnJB3XTX8
kLy6tqy/EM20XVo8vJJVZqKECaAO3JWknbfhtDJcEC0AtxHNiIcUzAkOhJx42I4E

e8r3rNfAPe3h5IRzp3LIYr8X3a5oPm4XCK8LMeMD989Ot3M7v9oBrd48WyxgGII
N
kX1/HZoTHM+axp176EZWmWyaGIrq0nWiGvEODoU35WZ//ciGsXgOWGjA1CIf
XRw1
f+4qbDEMrc+AATjQjVw2PGqx9dxJr/rZxwI93tZT2oD0UICVSH/ehppzZkd81YFb
EJm2iZNKq0wiOGmEkcxrvtQ+Wzhcflv3zipwmmKHs9IOpeG1ZABWphjWCahIPs
/I
EugZpEWadIi9cVRcr4B2PnmR3Tbosjgeetg9mRXSx8NFc6ADT6y7d7wOugkuIC
H7
F7ADF9iMJFoR9RaX0O63w4HxYrqX0TnYUM9YSkaUOmIRRCz0UAQte5f9Erv3
ycbR
F7PzsI+y+LHQKBq5YgFcs/T0o+b4QjZSWp+UhYxLM483Xe1Nk48NmVIXM+VM
LYBU
twpmRpmmf/qmB5BmM0miIkZOIi8y7XEYOfmOGo84xz2Rf6b4EvPW7heYhhcDtj
4S
Q73yVUeQXyhatI12CEaMMdS3Ymnj4XrOUfidFzAWVGOA+g9/0TH59LprXbyTaI
mf
F8+ve854vj/hvaZLuca0mGRTy+M87bXhvruMTNDBeV3S
-----END ENCRYPTED PRIVATE KEY-----

- **Bob (encrypted privateB.pem):**

-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIJnDBOBgkqhkiG9w0BBQ0wQTApBgkqhkiG9w0BBQwwHAQIw+X/mf1z788C
AggA
MAwGCCqGSIb3DQIJBQAwFAYIKoZIhvcNAwcECEAczf6gfqT6BIIJSNXUDqzXx
1jg
E+Rfhbrc6yGI/Ahu0u6U5XiFXoxUjVR3YOC+W306+xHm85X5Eu3IIo70fW7GXJ
oz
Q9hUW6r5VBtbEhmauaTvBOX6bHms19zArCUaPyQry6Y0TnJyxM4XOJcpfHBm
SbpY
eEaSxiTgGOAbl1R6gftdk1e3IXNkIaqJ79INo6z4WI3rHRHcMQl0LuotYHY+RKR
W
EnCsTupXBXMyUnB8EIf4UOeIQKbJ6X5nZLOMh6sgkdHIC4qKtCM+mBEjaAZH
MIJH
z04dFYKkUn20fgZu3VHLh/6TiEtFbn0A9WsHMeDNqO3K3U4aLXW8DDLHvJnq
6qpK
u/mN1cVUGGTN0YqQPP0NJZK1CcwEFXhJvATUzisDQmzsbaVedWbKUtZNFv
mhX4Yd
rPYuq+4J+IDcI+7jZmyOBuCH1YU5snMiSriIro21kr2yspbp9UdYXLiiO/g+Qsz/
JVGTtTMFTSVR7vjv9M9EBQ9BDvjG8frU4tfITHzciIa598NGdDBSIoNuPB9cWhB
W
Pw2uBXMQBwpN0rvJpEpfDALRXR4AmICjknBtWDNwEomNI8l9oc9CKZUnVn7v
TD8O
vfzQmztm/F4zRiIByEpcZkh6PSIPXCXP6Ot49Apy96cSdsEvTOk7rOotyHKDeI07

IYt2UqnogemiHP9gz003OQOEzK4SKwUdlEiuI1oSIYNvUoWrilv0jJSjaQ+TznGv
xqAf7ZCutKBsMkdBpmrp8GXFwdurft2MnRef3c7QgjIo6n35InniUrI/EJoyvKbo
dun5OvvQQ0z/tJb6kqpSkfZngYNJSGEn+JBAAMBhubaCDyWgRFxrSMBE3jO0
BpLj
K10FKshY0MbLYXxDFh+3bk+LE/hVIv0nVRJGw0uEd73QLVBExkfJGZbRngos2
GKM
wTLQNKct20PxI+dM4rkHfk5maIGoMIBzFiRI8s3uSNt5trIxCvunBKz1nMOOD2Bf
ksWqF87T56eK+ihPmwj/aljGiOAVsId3K533Iun/q60HLTvGRFxjotK/t/DXVzXs
63Qq/57hoAmtDUrFNrA1C1Z6CHQCQvOIvNXW2kDMmJfPR9SivsRg9+uInzZb
n2o1
hPQ5/Bt13fxbr9y9yOojFm1+YTtAKd7YRnoHW0padmaWaoq4jjIvDU38utkhcp2q
V9nn8M/XGOtId6yBOBqFaW+gV0iwfeGcTj4ePMJWpeHLJ7/nU9GJE6fiAP6GIB
PS
ITd9zvLFR/YohsHJ8orwNWCIUKoJ/1w2x+dGrcy7vWPYRPx6WTx3L676QISKsm
uu
cKV8I9r1k/1soVzNci0ugOHBJdnsazrmBtenjIZ6BCT9LgSzCevA4dyuUrffM8UJ
t8zg48p6ZLTprnx37VRSmmikmzPhq7oP/Z3sp98G2bgIogYONBK01RDvx1LQpN
+9
/u119jhE5ivQZMxoFzEg+XNOd2pCFULvo3666MBi3gmtXq4nihYUJ8+kca8Piiqq
p4vL9h8inxSomnXxIzTmAqLysg6YWOJu6nuFjis8jbVV+2w2qI5LGXgREk2S1AZ
m
UjIYUp6O+q6Ma+ErVy1t2k80J8PW5pOeS4DYGKqWS8zEKaE24OKUf8tudt/gG
04/
wEznh7k6Og89ojkbgS3cczOZuvlw3gUcaHy7M2aEdeKSXr6SQ0oL+XMy2K1bM
UWE
AmisYQAZ5kq90TmHePa0TuzJ7MQiL0hwTr4vpg2+snCezAx6DeBTIb8gK6BHn
man
wWWYr+/SOUnEobgfr2ML/6bp+ExvXe1Y8nN9dRhCqnjP/vhw6OcXz9IdLoAyNnt
4
cb4D0zVGbuZXSHK7C4wQuIUusVXwIesy6TmArUc0aOwq5TunENM79NGwbR
znN3IX
J80IL4UKQxwGo3emw/yjj+AIy9cD2wxEVOiR5S3CIvusWQyIGaVoyzUItYzD2Oq
W
cFCA5RUqXuByj2xCnwJw8k1ge6DxXJ0rnvmc97DahQBYNtEt6s4shsyd/6VVjIBF
dBbvWd21BImIXanjpIY9oPSYXiHwS35u8k6BA8fA2Vc0UNV/qJeB2Ar6FdH48HL
3
jiMztRjK2/jYX59stFQMhIOuva5ppxbyBPjGHqyemgm5pRJRId5QTz3HzrYo6BmX
estpNTFGaQ4L91ABBfMO02CRF0g3fjsQTGf/II/2x4HD2BiBgkOdr8zLgpvHS1o0
RNjxfdVfbNfNMVsTE/yyC/sGJQq/Yfxzoz+MVXOZGICdIeSLtaf3wehKeiI3Iahy
Pbc8pNLdIdk521FFMgpMcSdYt2gybTkMC0Nec630nwSCEYdUSIs3Xc8WG1YU
LPkf
QsXhfwAW+8eW56gQJZUSCPssjz8Hkspt4Q3MWD7V/NZca/1ag5jazrhrTOc3v/
BK
DxeJa7M8P+O1Bs2/869gS59/CFjxIBkBi7V7fmOxyjUgcE6E6kqf7rmsduy4QYRr

```
bKK6dSBvR+oDptvFLXipnyVQ5jIffg/nv/28e3wxvd8zaI7MjFcCifUqyhd4uWZ+
NpSWDcQ+ap2iRpSKib+t1ygHWP27aIJJafmnYGI/fZh/ndsD4JJVZy9aHRwHQC
by
inLYNen2Aj74bWtdF+jXhhXR3Do5jZU+7EanWkzx9RkqZuZtOswF+fDarSI23eTX
TRPwDrT0WVZz/zzEFJLtEY211Mu6xoeXoWHq28vukFv6fOLndmlGXLh2q2kNT
pG+
9KMRHLJF7hvItBvw/9fM1aeBe2lf7UxHTPdJDnCnRL+hkYOinXn3pFA4O0N21E
uD
hAP4aSSEc6GfgsyppuvhWchT4gbxeZfrhUd28TQVRiXIPe7zuU5Vbboe0CAGuk
6f
qdZtzMmnJIUj+1JcCOVIkRp3dsFjNflR97Yr/OIJufA/Tb0sAB+mc/UqFngpjQyq
KjFo4SwKkYciea+t2NyEQd2xFLINKQrV/UhmG5Fkfgz6pekfvUZBaMos4sdMAxH
0
D8oqzPz+xISPzIXqVLSDP8T/h6eG9nN+6AxxB2X4bSo7kkBOSumYcNXhQ86gI
mJn
iJe3ZWH56EenAo9cXWcungLiu/uxgECGFH6m28LStT6asQwREDquVnBb7mTA
C88I
oTFWA/tZfcLD6nYoKjDwJHQ2ardLA1JuWILGH14mkS+EjovdmJVXXnjY6JxugR
mm
8e3VHXF9kOom8rY8VJtfmw==
-----END ENCRYPTED PRIVATE KEY-----
```

- **cat SA.key**
  Symmetric encryption algorithm : aes-256-cbc
  Passphrase : ES18BTECH11028
  Iterations : 1000
- **cat SB.key**
  Symmetric encryption algorithm : aes-256-cbc
  Passphrase : ES18BTECH11019
  Iterations : 1000