Course Name:	**Applied Cryptography**	**Semester:**	**V**
Date of Performance:	**28/10/2025**	**DIV/ Batch No:**	**D-2**
Student Name:	**Shreyans Tatiya**	**Roll No:**	**16010123325**

Experiment No: 9

Title: **Implementation of OpenSSL**

Aim and Objective of the Experiment:

Demonstrate simple operations and functions using OpenSSL

COs to be achieved:

CO4: Understand Authentication Mechanisms and Evaluate Cryptographic Hash Functions

Books/ Journals/ Websites referred:

1.

Theory:

OpenSSL at command line : OpenSSL is a program and library that supports many different cryptographic operations. Each of the operations supported by OpenSSL have a variety of options, such as input/output files, algorithms, algorithm parameters and formats.

- Symmetric key encryption
- Public/private key pair generation
- Public key encryption
- Hash functions
- Certificate creation
- Digital signatures

Symmetric Key :

Symmetric-key algorithms are algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. The keys may be identical or there may be a simple transformation to go between the two keys.

Public -Private Key :

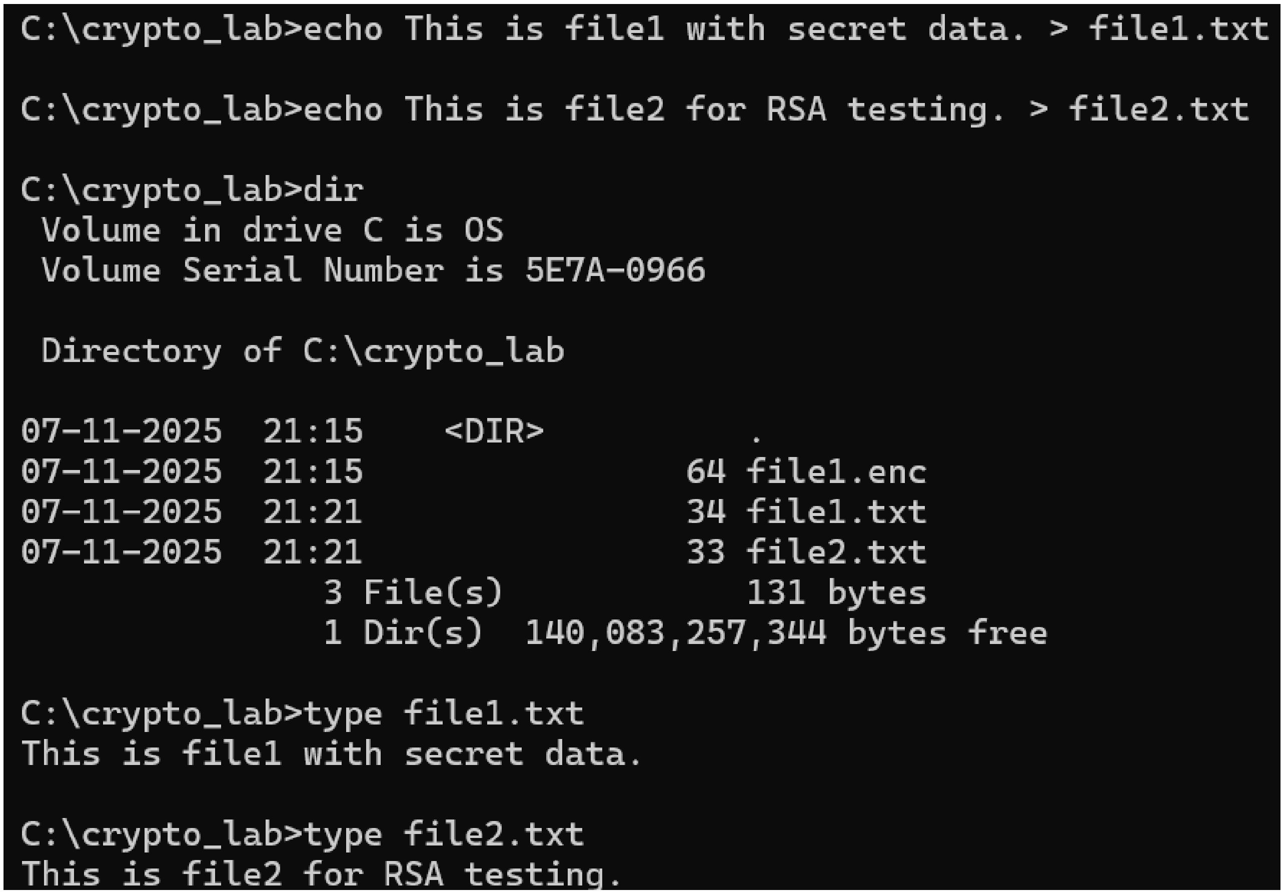
In a public key encryption system, any person can encrypt a message using the receiver's public key. That encrypted message can only be decrypted with the receiver's private key. To be practical, the generation of a public and private key -pair must be computationally economical. The strength of a public key cryptography system relies on the computational effort (work factor in cryptography) required to find the private key from its paired public key. Effective security only requires keeping the private key private; the public key can be openly distributed without compromising security.

Public key cryptography systems often rely on cryptographic algorithms based on mathematical problems that currently admit no efficient solution, particularly those inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. Public key algorithms, unlike symmetric key algorithms, do not require a secure channel for the initial exchange of one or more secret keys between the parties.

Before creating a CSR, the applicant first generates a key pair, keeping the private key secret. The CSR contains information identifying the applicant (such as a distinguished name in the case of an X.509 certificate) which must be signed using the applicant's private key. The CSR also contains the public key chosen by the applicant. The CSR may be accompanied by other credentials or proofs of identity required by the certificate authority, and the certificate authority may contact the applicant for further information.

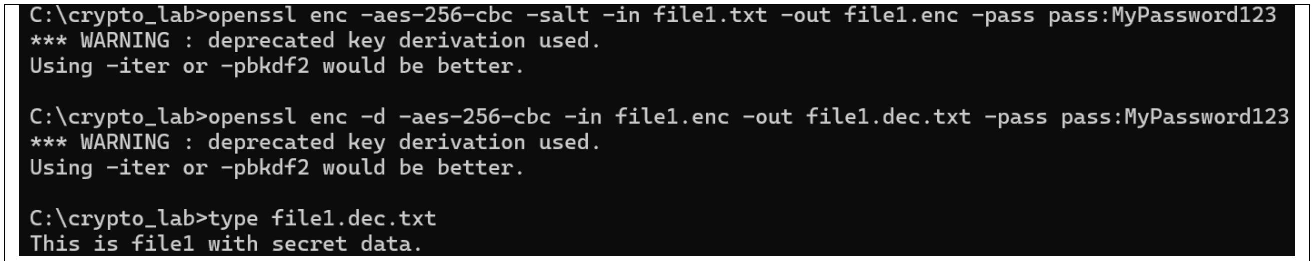
Working :

Creation of files



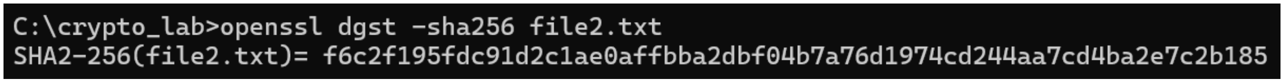
```
C:\crypto_lab>echo This is file1 with secret data. > file1.txt

C:\crypto_lab>echo This is file2 for RSA testing. > file2.txt

C:\crypto_lab>dir
 Volume in drive C is OS
 Volume Serial Number is 5E7A-0966

 Directory of C:\crypto_lab

07-11-2025  21:15    <DIR>          .
07-11-2025  21:15                64 file1.enc
07-11-2025  21:21                34 file1.txt
07-11-2025  21:21                33 file2.txt
               3 File(s)            131 bytes
               1 Dir(s)  140,083,257,344 bytes free

C:\crypto_lab>type file1.txt
This is file1 with secret data.

C:\crypto_lab>type file2.txt
This is file2 for RSA testing.
```

AES Symmetric Encryption & Decryption



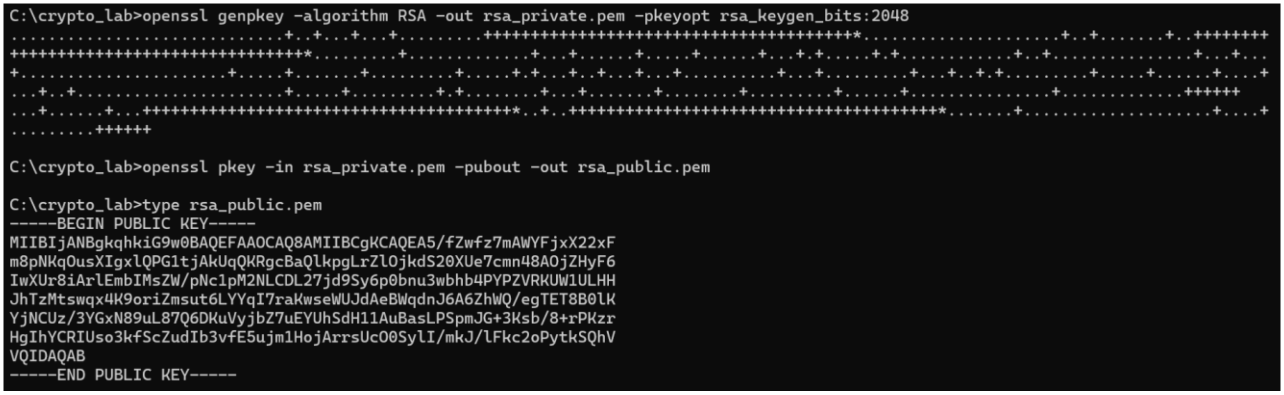
```
C:\crypto_lab>openssl enc -aes-256-cbc -salt -in file1.txt -out file1.enc -pass pass:MyPassword123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\crypto_lab>openssl enc -d -aes-256-cbc -in file1.enc -out file1.dec.txt -pass pass:MyPassword123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\crypto_lab>type file1.dec.txt
This is file1 with secret data.
```

Hashing



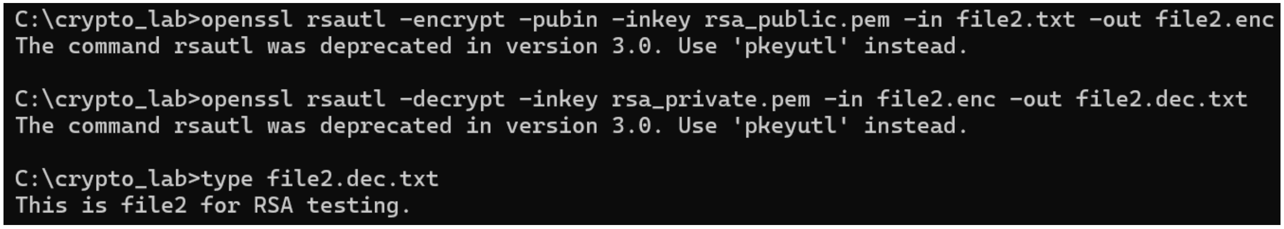
```
C:\crypto_lab>openssl dgst -sha256 file2.txt
SHA2-256(file2.txt)= f6c2f195fdc91d2c1ae0affbba2dbf04b7a76d1974cd244aa7cd4ba2e7c2b185
```

RSA Key Pair Generation



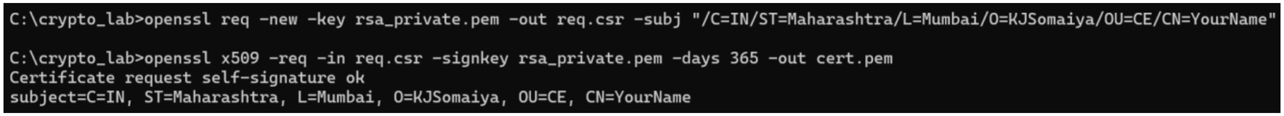
```
C:\crypto_lab>openssl genpkey -algorithm RSA -out rsa_private.pem -pkeyopt rsa_keygen_bits:2048
.............................+..+...+...+.........+++++++++++++++++++++++++++++++++++++++++*.....................+..+.......+..++++++++
+++++++++++++++++++++++++++++++*.........+.............+...+......+.....+......+...+.+.....+.+...........+..+...............+...+...
+.....................+.....+.......+.........+.....+.+...+..+..+...+..........+...+.........+...+..+.+.........+.....+......+....+
...+..+.....................+.....+.........+.+........+...+.......+........+.........+......+...............+............++++++
...+......+...++++++++++++++++++++++++++++++++++++++++*..+..++++++++++++++++++++++++++++++++++++++++*.......+...................+....+
.........++++++

C:\crypto_lab>openssl pkey -in rsa_private.pem -pubout -out rsa_public.pem

C:\crypto_lab>type rsa_public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5/fZwfz7mAWYFjxX22xF
m8pNKqOusXIgxlQPG1tjAkUqQKRgcBaQlkpgLrZlOjkdS20XUe7cmn48AOjZHyF6
IwXUr8iArlEmbIMsZW/pNc1pM2NLCDL27jd9Sy6p0bnu3wbhb4PYPZVRKUW1ULHH
JhTzMtswqx4K9oriZmsut6LYYqI7raKwseWUJdAeBWqdnJ6A6ZhWQ/egTET8B0lK
YjNCUz/3YGxN89uL87Q6DKuVyjbZ7uEYUhSdH11AuBasLPSpmJG+3Ksb/8+rPKzr
HgIhYCRIUso3kfScZudIb3vfE5ujm1HojArrsUcO0SylI/mkJ/lFkc2oPytkSQhV
VQIDAQAB
-----END PUBLIC KEY-----
```

RSA Encryption and Decryption

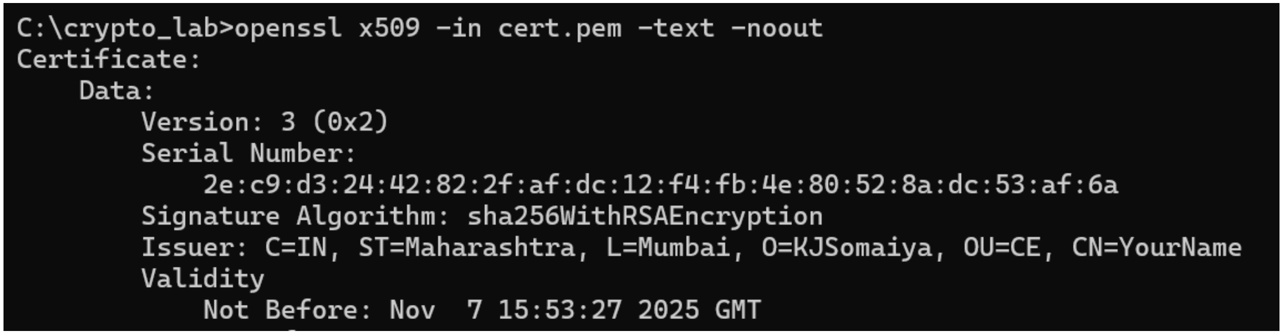


```
C:\crypto_lab>openssl rsautl -encrypt -pubin -inkey rsa_public.pem -in file2.txt -out file2.enc
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

C:\crypto_lab>openssl rsautl -decrypt -inkey rsa_private.pem -in file2.enc -out file2.dec.txt
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

C:\crypto_lab>type file2.dec.txt
This is file2 for RSA testing.
```

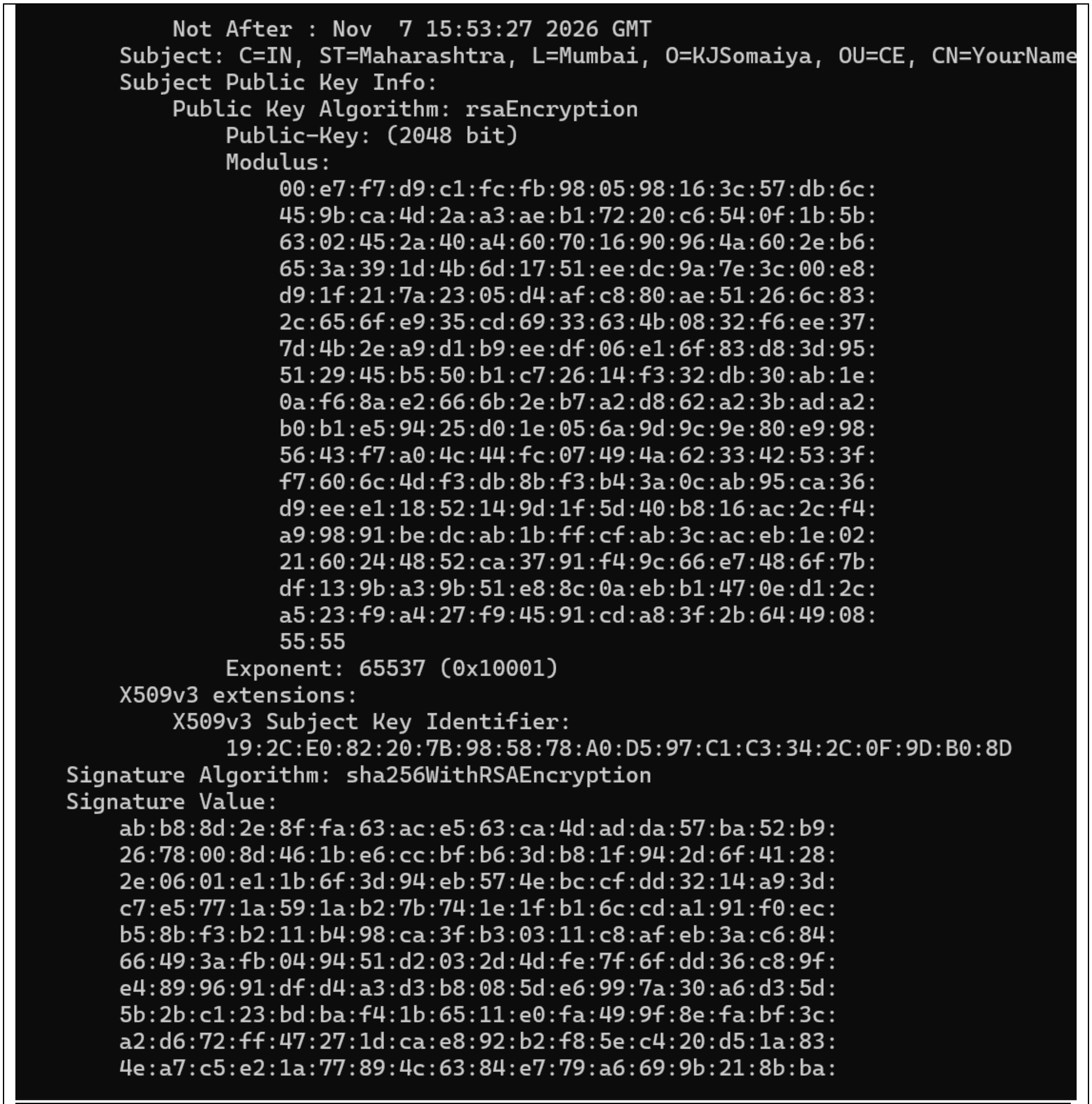
Certificate Creation & Viewing



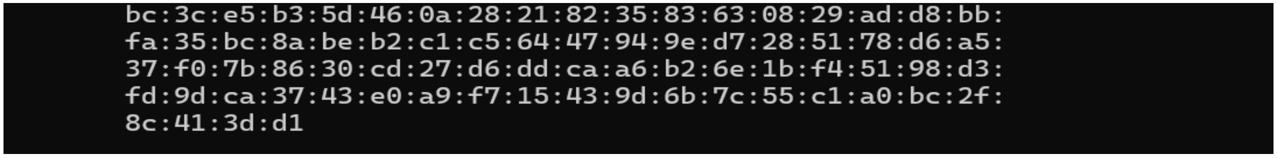
```
C:\crypto_lab>openssl req -new -key rsa_private.pem -out req.csr -subj "/C=IN/ST=Maharashtra/L=Mumbai/O=KJSomaiya/OU=CE/CN=YourName"

C:\crypto_lab>openssl x509 -req -in req.csr -signkey rsa_private.pem -days 365 -out cert.pem
Certificate request self-signature ok
subject=C=IN, ST=Maharashtra, L=Mumbai, O=KJSomaiya, OU=CE, CN=YourName
```



```
C:\crypto_lab>openssl x509 -in cert.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            2e:c9:d3:24:42:82:2f:af:dc:12:f4:fb:4e:80:52:8a:dc:53:af:6a
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=IN, ST=Maharashtra, L=Mumbai, O=KJSomaiya, OU=CE, CN=YourName
        Validity
            Not Before: Nov  7 15:53:27 2025 GMT
```

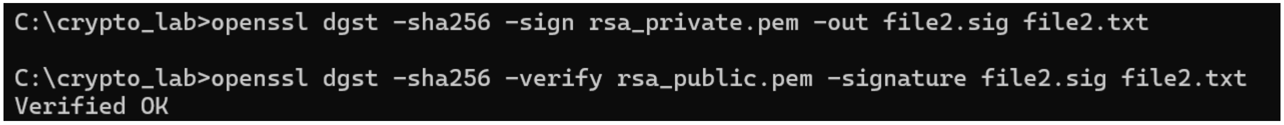


```
            Not After : Nov  7 15:53:27 2026 GMT
        Subject: C=IN, ST=Maharashtra, L=Mumbai, O=KJSomaiya, OU=CE, CN=YourName
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:e7:f7:d9:c1:fc:fb:98:05:98:16:3c:57:db:6c:
                    45:9b:ca:4d:2a:a3:ae:b1:72:20:c6:54:0f:1b:5b:
                    63:02:45:2a:40:a4:60:70:16:90:96:4a:60:2e:b6:
                    65:3a:39:1d:4b:6d:17:51:ee:dc:9a:7e:3c:00:e8:
                    d9:1f:21:7a:23:05:d4:af:c8:80:ae:51:26:6c:83:
                    2c:65:6f:e9:35:cd:69:33:63:4b:08:32:f6:ee:37:
                    7d:4b:2e:a9:d1:b9:ee:df:06:e1:6f:83:d8:3d:95:
                    51:29:45:b5:50:b1:c7:26:14:f3:32:db:30:ab:1e:
                    0a:f6:8a:e2:66:6b:2e:b7:a2:d8:62:a2:3b:ad:a2:
                    b0:b1:e5:94:25:d0:1e:05:6a:9d:9c:9e:80:e9:98:
                    56:43:f7:a0:4c:44:fc:07:49:4a:62:33:42:53:3f:
                    f7:60:6c:4d:f3:db:8b:f3:b4:3a:0c:ab:95:ca:36:
                    d9:ee:e1:18:52:14:9d:1f:5d:40:b8:16:ac:2c:f4:
                    a9:98:91:be:dc:ab:1b:ff:cf:ab:3c:ac:eb:1e:02:
                    21:60:24:48:52:ca:37:91:f4:9c:66:e7:48:6f:7b:
                    df:13:9b:a3:9b:51:e8:8c:0a:eb:b1:47:0e:d1:2c:
                    a5:23:f9:a4:27:f9:45:91:cd:a8:3f:2b:64:49:08:
                    55:55
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                19:2C:E0:82:20:7B:98:58:78:A0:D5:97:C1:C3:34:2C:0F:9D:B0:8D
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        ab:b8:8d:2e:8f:fa:63:ac:e5:63:ca:4d:ad:da:57:ba:52:b9:
        26:78:00:8d:46:1b:e6:cc:bf:b6:3d:b8:1f:94:2d:6f:41:28:
        2e:06:01:e1:1b:6f:3d:94:eb:57:4e:bc:cf:dd:32:14:a9:3d:
        c7:e5:77:1a:59:1a:b2:7b:74:1e:1f:b1:6c:cd:a1:91:f0:ec:
        b5:8b:f3:b2:11:b4:98:ca:3f:b3:03:11:c8:af:eb:3a:c6:84:
        66:49:3a:fb:04:94:51:d2:03:2d:4d:fe:7f:6f:dd:36:c8:9f:
        e4:89:96:91:df:d4:a3:d3:b8:08:5d:e6:99:7a:30:a6:d3:5d:
        5b:2b:c1:23:bd:ba:f4:1b:65:11:e0:fa:49:9f:8e:fa:bf:3c:
        a2:d6:72:ff:47:27:1d:ca:e8:92:b2:f8:5e:c4:20:d5:1a:83:
        4e:a7:c5:e2:1a:77:89:4c:63:84:e7:79:a6:69:9b:21:8b:ba:
```



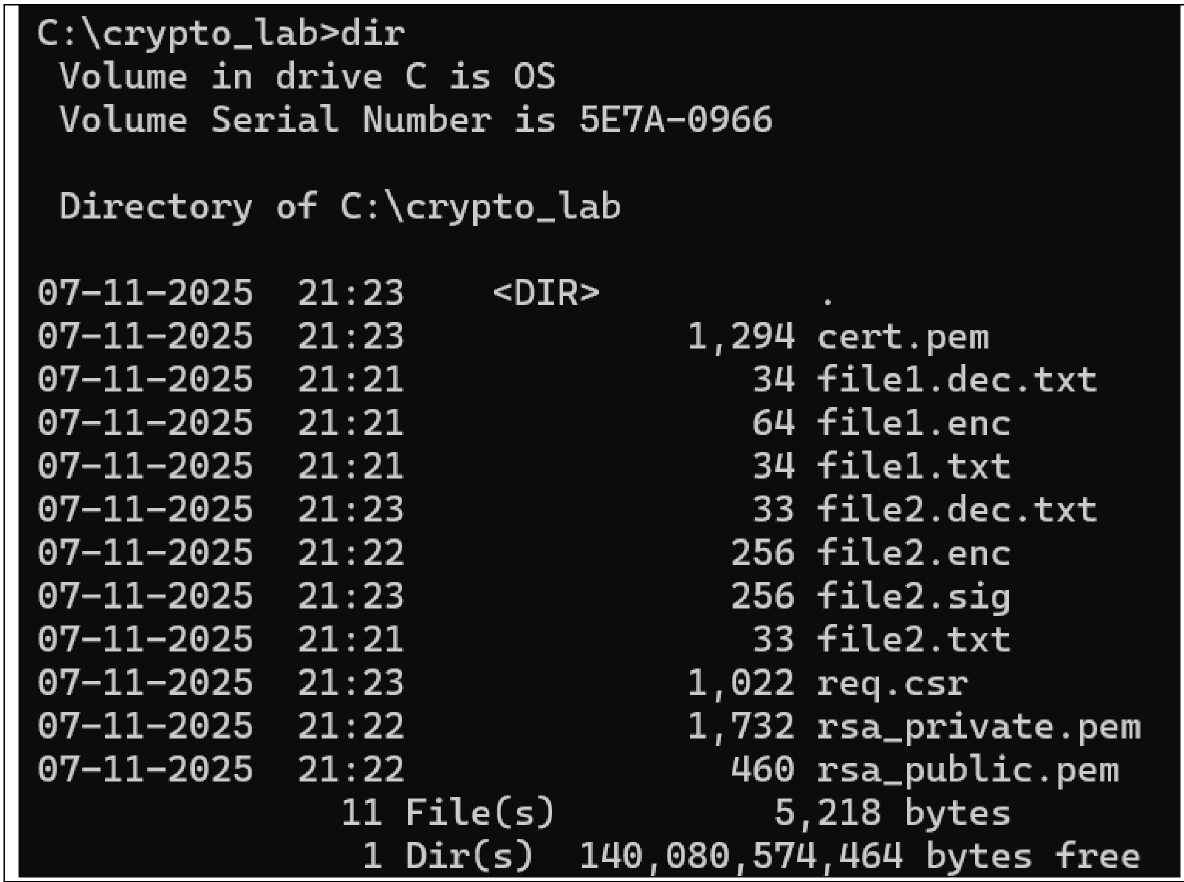
```
        bc:3c:e5:b3:5d:46:0a:28:21:82:35:83:63:08:29:ad:d8:bb:
        fa:35:bc:8a:be:b2:c1:c5:64:47:94:9e:d7:28:51:78:d6:a5:
        37:f0:7b:86:30:cd:27:d6:dd:ca:a6:b2:6e:1b:f4:51:98:d3:
        fd:9d:ca:37:43:e0:a9:f7:15:43:9d:6b:7c:55:c1:a0:bc:2f:
        8c:41:3d:d1
```

Digital Signature (Sign + Verify)



```
C:\crypto_lab>openssl dgst -sha256 -sign rsa_private.pem -out file2.sig file2.txt

C:\crypto_lab>openssl dgst -sha256 -verify rsa_public.pem -signature file2.sig file2.txt
Verified OK
```

View All Generated Files



```
C:\crypto_lab>dir
 Volume in drive C is OS
 Volume Serial Number is 5E7A-0966

 Directory of C:\crypto_lab

07-11-2025  21:23    <DIR>          .
07-11-2025  21:23             1,294 cert.pem
07-11-2025  21:21                34 file1.dec.txt
07-11-2025  21:21                64 file1.enc
07-11-2025  21:21                34 file1.txt
07-11-2025  21:23                33 file2.dec.txt
07-11-2025  21:22               256 file2.enc
07-11-2025  21:23               256 file2.sig
07-11-2025  21:21                33 file2.txt
07-11-2025  21:23             1,022 req.csr
07-11-2025  21:22             1,732 rsa_private.pem
07-11-2025  21:22               460 rsa_public.pem
              11 File(s)          5,218 bytes
               1 Dir(s)  140,080,574,464 bytes free
```

Post Lab Subjective/Objective type Questions:

Discuss applications of OpenSSL

OpenSSL is a widely used open-source cryptographic library that secures digital communications. It offers tools for data encryption and decryption, SSL/TLS certificate generation and management, and enables secure web connections (HTTPS). The library supports digital signatures, hashing algorithms, and public-key-based authentication. OpenSSL is extensively utilized in web servers, email systems, VPNs, and other applications that require secure data transmission. It also facilitates file integrity verification, password protection, and SSL/TLS implementation in custom software, making it a vital component for maintaining confidentiality, integrity, and authenticity in network security.

Conclusion:

In conclusion, this experiment successfully implemented various cryptographic operations using OpenSSL, including encryption, decryption, hashing, key generation, certificate creation, and digital signatures through command-line tools. It strengthened the understanding of both symmetric and asymmetric key mechanisms and emphasized the crucial role of OpenSSL in ensuring data security, authentication, and integrity in real-world applications.