EXPERIMENT 2

Familiarization with OpenSSL

2.1 Aim

To familiarize Open SSL, which is a command line open source tool used for secure communication.

2.2 Theory

OpenSSL is an open source project that provides a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library.

The OpenSSL program provides a rich variety of commands, each of which often has a wealth of options and arguments. The list parameters standard-commands, digest commands, and cipher-commands output a list (one entry per line) of the names of all standard commands, message digest commands, or cipher commands, respectively, that are available in the present OpenSSL utility.

2.3 Standard Commands

2.3.1 Encryption/ Decryption using AES

Encrypting a plain text file with AES:

Create the text file (openssl.txt) :

openssl enc -aes-256-cbc -pbkdf2 -in openssl.txt –out aes_enc.bin.

The password used here is 2833

```
C:\Users\jithu\Desktop\M1 SCLAB>openssl enc -aes-256-cbc -pbkdf2 -in openssl.txt -out aes_enc.bin
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
```

Figure 1: AES encryption

Decryption:

openssl enc -aes-256-cbc -d -pbkdf2 -in aes_enc.bin -pass

```
C:\Users\jithu\Desktop\M1 SCLAB>openssl enc -aes-256-cbc -d -pbkdf2 -in aes_enc.bin
enter AES-256-CBC decryption password:
Experiment 2 open ssl
```

Figure 2: AES decryption

2.3.2 RSA - public and private key generation - Encryption / Decryption

Create RSA key pair of 1024 bits:

openssl genrsa -out keypair.pem 1024

To view:

type keypair.pem

```
C:\Users\jithu\Desktop\M1 SCLAB>type  keypair.pem
-----BEGIN PRIVATE KEY-----
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAO+tuhzFBGgXKEDA
USe9x2+kKtJNjxJrr+lEhm6sC3WLWJ2er6eN2Oel6FOhKcJm+Pd+aMGxD+V9FFEy
qq3f/1H/B+PXyOQTMi8W/dJcvhIOPR/9J11oKz9lLE8USTtf2FtIx/KXMVcX6L6C
vMBPwRBvbZuT/VNDNcFPnWOLDBIxAgMBAAECgYA/5dD4Wo4SMhpJKPx5296PpPTF
nGfIQW1kgyBT4QotDDlTzDqIMQjiuRh3M0Au08UMNmc3PD1AsVf5TcizEkS+02cs
fIvIv9t5VUsO8KNYczJgglriU6KWFgDeaT89YFtpMpbjwxwecCtQG9plbw3TCFzy
rjKTQTR24wqgzdOtAQJBAPubqhdZVeOjMMUOgOjzHFGnbPxSmRlkd1nBc8Bm5HKz
yrjjkGjook4SfFj4CJw4hNFEZrGxazFUQqUlXNn/lPsCQQDz3MEy87KWLZHERf7+
xxNp5+4+yeoSsJo0V1WNlsFdTp9cUNUj+IH4lJ0Ekv/EmWTYiO10CU52Gs/nZicn
XxXDAkA2MZ09Ujyxv2Ct0SXrBxI+dDWBU7kuQvmCF6z68C6cliVAFaPxNMpgzpKC
B0qze2kAOh90fqZ8BY4lTBrnyakpAkBgHeeMVJ9UL/vfN5ONChwtxvuRhfYbjb4J
2U5gM0Zdp4qKUVXhCqb3umO1FNd4vtkPu9CTxsNFK8Y8N3hBPWoPAkEAw2CZA54u
LTDluYlvJ1h8W6iT63WOui+NI7R4tMgjgRJOFn0cAZOC3eEkeMxlJQy5Wd2T914D
usCzieg0XWit6g==
-----END PRIVATE KEY-----
```

Figure 3 : Private key

For detailed view:

openssl rsa -in keypair.pem -text –noout

```
C:\Users\jithu\Desktop\M1 SCLAB>openssl rsa -in keypair.pem -text -noout
Private-Key: (1024 bit, 2 primes)
modulus:
    00:ef:ad:ba:1c:c5:04:68:17:28:40:c0:51:27:bd:
    c7:6f:a4:2a:d2:4d:8f:12:6b:af:e9:44:86:6e:ac:
    0b:75:8b:58:9d:9e:af:a7:8d:d8:e7:a5:e8:53:a1:
    29:c2:66:f8:f7:7e:68:c1:b1:0f:e5:7d:14:51:32:
    aa:ad:df:ff:51:ff:07:e3:d7:c8:e4:13:32:2f:16:
    fd:d2:5c:be:12:0e:3d:1f:fd:27:5d:68:2b:3f:65:
    2c:4f:14:49:3b:5f:d8:5b:48:c7:f2:97:31:57:17:
    e8:be:82:bc:c0:4f:c1:10:6f:6d:9b:93:fd:53:43:
    35:c1:4f:9d:63:8b:0c:12:31
publicExponent: 65537 (0x10001)
privateExponent:
    3f:e5:d0:f8:5a:8e:12:32:1a:49:28:fc:79:db:de:
    8f:a4:f4:c5:9c:67:c8:41:6d:64:83:20:53:e1:0a:
    2d:0c:39:53:cc:3a:88:31:08:e2:b9:18:77:33:40:
    2e:d3:c5:0c:36:67:37:3c:3d:40:b1:57:f9:4d:c8:
    b3:12:44:be:d3:67:2c:7c:8b:c8:bf:db:79:55:4b:
    0e:f0:a3:58:73:32:60:82:5a:e2:53:a2:96:16:00:
    de:69:3f:3d:60:5b:69:32:96:e3:c3:1c:1e:70:2b:
    50:1b:da:65:6f:0d:d3:08:5c:f2:ae:32:93:41:34:
    76:e3:0a:a0:cd:d3:ad:01
prime1:
    00:fb:9b:aa:17:59:55:e3:a3:30:c5:0e:80:e8:f3:
    1c:51:a7:6c:fc:52:99:19:64:77:59:c1:73:c0:66:
    e4:72:b3:ca:b8:e3:90:68:e8:a2:4e:12:7c:58:f8:
    08:9c:38:84:d1:44:66:b1:b1:6b:31:54:42:a5:25:
    5c:d9:ff:94:fb
prime2:
    00:f3:dc:c1:32:f3:b2:96:2d:91:c4:45:fe:fe:c7:
    13:69:e7:ee:3e:c9:ea:12:b0:9a:34:57:55:8d:96:
    c1:5d:4e:9f:5c:50:d5:23:f8:81:f8:94:9d:04:92:
    ff:c4:99:64:d8:88:ed:74:09:4e:76:1a:cf:e7:66:
    27:27:5f:15:c3
exponent1:
    36:31:9d:3d:52:3c:b1:bf:60:ad:d1:25:eb:07:12:
    3e:74:35:81:53:b9:2e:42:f9:82:17:ac:fa:f0:2e:
    9c:96:25:40:15:a3:f1:34:ca:60:ce:92:82:07:4a:
    b3:7b:69:00:3a:1f:74:7e:a6:7c:05:8e:25:4c:1a:
    e7:c9:a9:29
exponent2:
    60:1d:e7:8c:54:9f:54:2f:fb:df:37:93:8d:0a:1c:
    2d:c6:fb:91:85:f6:1b:8d:be:09:d9:4e:60:33:46:
    5d:a7:8a:8a:51:55:e1:0a:a6:f7:ba:63:b5:14:d7:
    78:be:d9:0f:bb:d0:93:c6:c3:45:2b:c6:3c:37:78:
    41:3d:6a:0f
coefficient:
    00:c3:60:99:03:9e:2e:2d:30:e5:b9:89:6f:27:58:
```

Figure 4: Detailed view

To encrypt the key file:

openssl rsa -in keypair.pem -des3 -out enc-key.pem

Enter password

```
C:\Users\jithu\Desktop\M1 SCLAB>openssl rsa -in keypair.pem -des3 -out enc-key.pem
writing RSA key
Enter pass phrase:
Verifying - Enter pass phrase:
```

Figure 5: RSA encryption

To extract public key from key.pem file:

openssl rsa -in keypair.pem -pubout -out pub-key.pem

To view: type pub-key.pem

```
C:\Users\jithu\Desktop\M1 SCLAB>type pub-key.pem
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDvrbocxQRoFyhAwFEnvcdvpCrS
TY8Sa6/pRIZurAt1i1idnq+njdjnpehToSnCZvj3fmjBsQ/lfRRRMqqt3/9R/wfj
18jkEzIvFv3SXL4SDj0f/SddaCs/ZSxPFEk7X9hbSMfylzFXF+i+grzAT8EQb22b
k/1TQzXBT51jiwwSMQIDAQAB
-----END PUBLIC KEY-----
```

Figure 6: Public key

Encrypt a file using generated key file (default: pub key):

openssl pkeyutl -encrypt -in openssl.txt -pubin -inkey pub-key.pem -out rsa enc.bin

For decryption:

openssl pkeyutl -decrypt -in rsa enc.bin -inkey keypair.pem -out dec-rsa-openssl.txt

2.4 Result

OpenSSL has been successfully familiarized.