
# NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

## DEPARTMENT OF COMPUTER SCIENCE

## INFORMATION SECURITY LAB

| | |
|---|---|
| **Name** | Ayesha Imran |
| **Class** | CS-A |
| **Lab** | 05 |
| **Course** | Information Security |
| **Date** | 16-October-25 |
| **Submitted To** | Lec. Attiya Ashraf |

# IN LAB TASKS

## Part 1: Verify OpenSSL Installation

1. OPEN SSL VERSION:

COMMAND: openssl version

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl version
OpenSSL 3.0.13 30 Jan 2024 (Library: OpenSSL 3.0.13 30 Jan 2024)
```

## Part 2: Introduction to OpenSSL Command-Line Tools

1. OPEN SSL HELP:

COMMAND: openssl help

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl help
help:

Standard commands
asn1parse         ca                ciphers           cmp
cms               crl               crl2pkcs7         dgst
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
fipsinstall       gendsa            genpkey           genrsa
help              info              kdf               list
mac               nseq              ocsp              passwd
pkcs12            pkcs7             pkcs8             pkey
pkeyparam         pkeyutl           prime             rand
rehash            req               rsa               rsautl
s_client          s_server          s_time            sess_id
smime             speed             spkac             srp
storeutl          ts                verify            version
x509

Message Digest commands (see the `dgst' command for more details)
blake2b512        blake2s256        md4               md5
rmd160            sha1              sha224            sha256
sha3-224          sha3-256          sha3-384          sha3-512
sha384            sha512            sha512-224        sha512-256
```

```
Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      base64
bf                bf-cbc            bf-cfb            bf-ecb
bf-ofb            camellia-128-cbc  camellia-128-ecb  camellia-192-cbc
camellia-192-ecb  camellia-256-cbc  camellia-256-ecb  cast
cast-cbc          cast5-cbc         cast5-cfb         cast5-ecb
cast5-ofb         des               des-cbc           des-cfb
des-ecb           des-ede           des-ede-cbc       des-ede-cfb
des-ede-ofb       des-ede3          des-ede3-cbc      des-ede3-cfb
des-ede3-ofb      des-ofb           des3              desx
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            seed              seed-cbc          seed-cfb
seed-ecb          seed-ofb          sm4-cbc           sm4-cfb
```

## 2. Information about an individual openssl command:

COMMAND: openssl esc -help

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl enc -help
Usage: enc [options]

General options:
 -help               Display this summary
 -list               List ciphers
 -ciphers            Alias for -list
 -e                  Encrypt
 -d                  Decrypt
 -p                  Print the iv/key
 -P                  Print the iv/key and exit
 -engine val         Use engine, possibly a hardware device

Input options:
 -in infile          Input file
 -k val              Passphrase
 -kfile infile       Read passphrase from file

Output options:
 -out outfile        Output file
 -pass val           Passphrase source
 -v                  Verbose output
 -a                  Base64 encode/decode, depending on encryption flag
 -base64             Same as option -a
```

```
Encryption options:
 -nopad              Disable standard block padding
 -salt               Use salt in the KDF (default)
 -nosalt             Do not use salt in the KDF
 -debug              Print debug info
 -bufsize val        Buffer size
 -K val              Raw key, in hex
 -S val              Salt, in hex
 -iv val             IV in hex
 -md val             Use specified digest to create a key from the passphrase
 -iter +int          Specify the iteration count and force the use of PBKDF2
                     Default: 10000
 -pbkdf2             Use password-based key derivation function 2 (PBKDF2)
                     Use -iter to change the iteration count from 10000
 -none               Don't encrypt
 -*                  Any supported cipher

Random state options:
 -rand val           Load the given file(s) into the random number generator
 -writerand outfile  Write random data to the specified file

Provider options:
 -provider-path val  Provider load path (must be before 'provider' argument if
equired)
```

## 3. The detailed Information :

COMMAND: man openssl

```
OPENSSL(1SSL)                       OpenSSL                       OPENSSL(1SSL)

NAME
       openssl - OpenSSL command line program

SYNOPSIS
       openssl command [ options ... ] [ parameters ... ]

       openssl no-XXX [ options ]

DESCRIPTION
       OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer
       (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and
       related cryptography standards required by them.

       The openssl program is a command line program for using the various
       cryptography functions of OpenSSL's crypto library from the shell.  It
       can be used for

         o  Creation and management of private keys, public keys and parameters
         o  Public key cryptographic operations
         o  Creation of X.509 certificates, CSRs and CRLs
         o  Calculation of Message Digests and Message Authentication Codes
 Manual page openssl(1ssl) line 1 (press h for help or q to quit)
```

4. Different Crypto Algoruthms:

COMMAND: openssl list-cipher-algorithms or with openssl enc -ciphers

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl enc -ciphers
Supported ciphers:
-aes-128-cbc               -aes-128-cfb               -aes-128-cfb1
-aes-128-cfb8              -aes-128-ctr               -aes-128-ecb
-aes-128-ofb               -aes-192-cbc               -aes-192-cfb
-aes-192-cfb1              -aes-192-cfb8              -aes-192-ctr
-aes-192-ecb               -aes-192-ofb               -aes-256-cbc
-aes-256-cfb               -aes-256-cfb1              -aes-256-cfb8
-aes-256-ctr               -aes-256-ecb               -aes-256-ofb
-aes128                    -aes128-wrap               -aes192
-aes192-wrap               -aes256                    -aes256-wrap
-aria-128-cbc              -aria-128-cfb              -aria-128-cfb1
-aria-128-cfb8             -aria-128-ctr              -aria-128-ecb
-aria-128-ofb              -aria-192-cbc              -aria-192-cfb
-aria-192-cfb1             -aria-192-cfb8             -aria-192-ctr
-aria-192-ecb              -aria-192-ofb              -aria-256-cbc
-aria-256-cfb              -aria-256-cfb1             -aria-256-cfb8
-aria-256-ctr              -aria-256-ecb              -aria-256-ofb
-aria128                   -aria192                   -aria256
-bf                        -bf-cbc                    -bf-cfb
-bf-ecb                    -bf-ofb                    -blowfish
-camellia-128-cbc          -camellia-128-cfb          -camellia-128-cfb1
-camellia-128-cfb8         -camellia-128-ctr          -camellia-128-ecb
```

# Part 3: Symmetric Encryption Using AES

1. Create a File to Encrypt:

COMMAND: echo "This is a sample plaintext file for encryption." > plaintext.txt

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ echo "This is a sample
 plaintext file for encryption." > plaintext.txt
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$
```

2. Encrypting a File:

COMMAND: openssl enc -aes-256-cbc -pbkdf2 -in plaintext.txt -out encrypted.txt

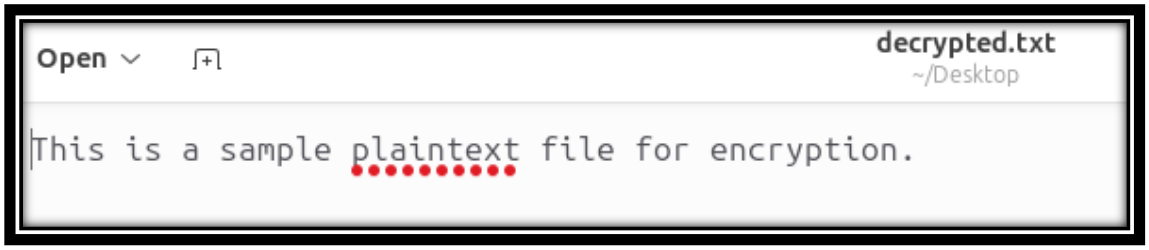
```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl enc -aes-256-c
bc -pbkdf2 -in plaintext.txt -out encrypted.txt
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
```

Encrypted File:


3. Decrypting a File:

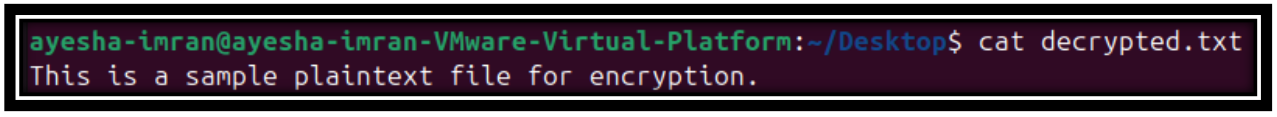
COMMAND: openssl enc -aes-256-cbc -d -pbkdf2 -in encrypted.txt -out decrypted.txt


Decrypted File:




4. Verification:

COMMAND: cat decrypted.txt



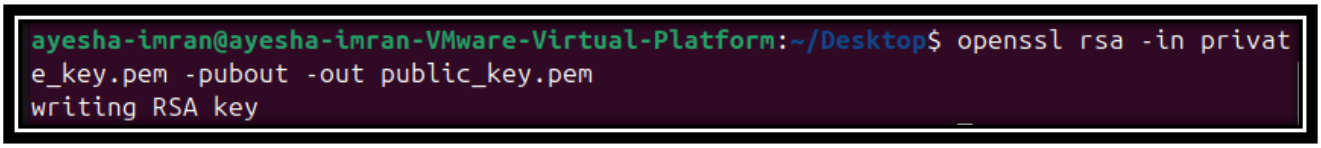

# Part 4: Asymmetric Key Generation Using RSA

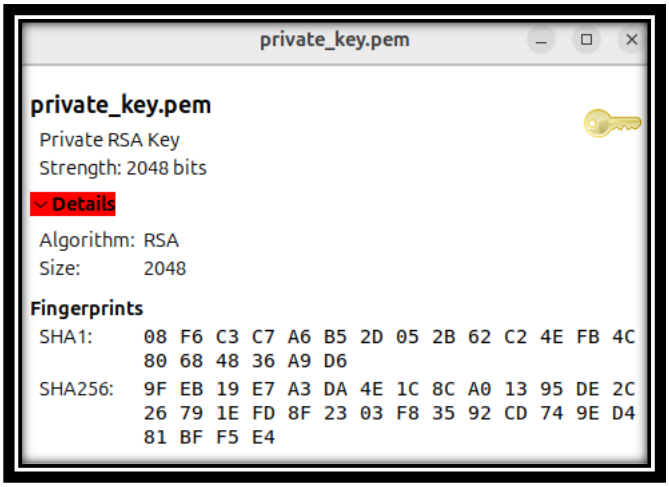
1. Generate an RSA Private Key:

COMMAND: openssl genpkey -algorithm RSA -out private_key.pem –pkeyopt rsa_keygen_bits:2048


2. Extract The Public Key:

COMMAND: openssl rsa -in private_key.pem -pubout -out public_key.pem




# Part 5: Encrypting and Decrypting Using RSA

1. Encrypting with the Public Key:

COMMAND: openssl pkeyutl -encrypt -in plaintext.txt -inkey public_key.pem -pubin –out encrypted_rsa.txt

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl pkeyutl -encry
pt -in plaintext.txt -inkey public_key.pem -pubin -out encrypted_rsa.txt
```


## 2. Decrypting with the Private Key:

COMMAND: openssl pkeyutl -decrypt -in encrypted_rsa.txt -inkey private_key.pem –out decrypted_rsa.txt

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ openssl pkeyutl -decry
pt -in encrypted_rsa.txt -inkey private_key.pem -out decrypted_rsa.txt
```


## 3. Verification:

COMMAND: cat decrypted_rsa.txt

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ cat decrypted_rsa.txt
This is a sample plaintext file for encryption.
```