

Asymmetric Encryption using RSA

//Generate RSA Private Key

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

//Extract Public Key from Private Key

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

//Create a Test File

```
echo "This is a test file for RSA encryption." > test_file.txt
```

//Encrypt the Test File using RSA public key

```
openssl pkeyutl -encrypt -in test_file.txt -pubin -inkey public_key.pem -out test_file_encrypted.bin
```

//Open and verify that File is Encrypted

```
cat test_file_encrypted.bin
```

//Decrypt the Test File using RSA private key

```
openssl pkeyutl -decrypt -in test_file_encrypted.bin -inkey private_key.pem -out  
test_file_decrypted.bin
```

//Open and verify that File is Decrypted

```
cat test_file_decrypted.bin
```

Symmetric Encryption Using AES

//Create a Test File

```
echo "This is a sample file for AES encryption lab." > test_file.txt
```

//Generate a Secret Key for AES

```
openssl rand -base64 32 > aes_key.bin
```

//Encrypt the File Using AES

```
openssl enc -aes-256-cbc -salt -in test_file.txt -out encrypted_file.bin -pass file:aes_key.bin
```

//Verify the Encrypted File

```
cat encrypted_file.bin
```

//Decrypt the Encrypted File

```
openssl enc -d -aes-256-cbc -in encrypted_file.bin -out decrypted_file.txt -pass file:aes_key.bin
```

//Verify the Decrypted file

```
cat decrypted_file.txt
```