CS 253 OpensSSL Project
Carlo R. Raquel
96-50527

For the implementation of this project, I have used Linux Mint 18.1, OpenSSL 1.0.2 and a bash shell. To perform a specific task, a shell script was written which can be executed on the command line of bash.

I. SYMMETRIC ENCRYPTION (AES)

A. Using ECB

To perform symmetric AES ecryption on the 512x512 Color (24.ibt) Lena image using ECB, run the shell script *aes128_ecb.sh* which contains the command:

openssl aes-128-ecb -salt -a -e -in lena512color.tiff -out cs253_output1.enc

It will prompt for a password and I used the password *cs253project.* It will output the encrypted file named *cs253_output1.enc*.

To perform decryption, run the script named *aes128_ecb_decrypt.sh* which contains the command

openssl aes-128-ecb -salt -a -e -in lena512color.tiff -out cs253_output1.enc

It will prompt for a password, enter *cs253project* as password and it will generate the decrypted file *lena_ecbdecrypted.tiff*.

B. Using CBC

To perform symmetric AES ecryption on the 512x512 Color (24.ibt) Lena image using CBC, run the shell script script *aes128_cbcsh* which contains the command

openssl aes-128-cbc -salt -a -e -in lena512color.tiff -out cs253_output2.enc

It will prompt for a password and I used the password *cs253project* and it will output the encrypted file named *cs253_output2.enc*.

To decrypt, run the script *aes128_cbc_decrypt.sh*. It will prompt for a password and I used cs253project as the password. It will decrypt the file and output *lena_ecbdecrypted.tiff*. The script contains the command

openssl aes-128-cbc -salt -a -d -in cs253_output2.enc -out lena_cbcdecrypted.tiff

II. HASHING

Using OpenSSL, to hash the Lena 512x512 image using the following hash functions: SHA-1, SHA-256, SHA-512, run the corresponding file to get the desired output.

A. Using SHA-1

Run the script *sha1_hash.sh* which contains the command

openssl dgst -sha1 lena512color.tiff

Output: SHA1(lena512color.tiff)= e647d0f6736f82e498de8398eccc48cf0a7d53b9

B. Using SHA-256

Run the script *sha1_hash.sh* which contains the command

openssl dgst -sha256 lena512color.tiff

Output: SHA256(lena512color.tiff)= c056da23302d2fb0d946e7ffa11e0d94618224193ff6e2f78ef8097bb8a3569b

C. Using SHA-512

Run the script *sha512_hash.sh* which contains the command

openssl dgst -sha512 lena512color.tiff

Output: SHA512(lena512color.tiff)=
2cb9d7df53eb8640dc48d736974f472a98d9c7186de7a972490455f5f3ed29dfc5b75c95ccb3ed4596bc2bfc4b1e52cf4d76bcee27d334dd155bb426617392dc

III. PUBLIC KEY ENCRYPTION

A. RSA

In order to perform public key encryption, we have to generate a private key and a public key. We can generate a private key of RSA-2048 using the command

openssl genrsa -out my_private_key.pem 2048

To generate the RSA public key, we can use the command

openssl rsa -in my_private_key.pem -outform PEM -pubout my_public_key.pem

If we use RSA to directly encrypt the Lena image, it will result to an error “data too large” as RSA can only be used to encrypt small data and not large images such as images and videos. Instead, we can use symmietric encryption like AES by running the script *rsa_encrypt.sh* which contains the following commands:

```
# Generate an RSA private key
openssl genrsa -out rsa-priv.pem 2048

# Generate an RSA public key
openssl rsa -in rsa-priv.pem -pubout -out rsa-pub.pem -outform PEM

# Generate an AES key
openssl rand -base64 128 > aes-key.txt

# Encrypt the AES key with the public RSA key and save to an encrypted file
openssl rsautl -encrypt -inkey rsa-pub.pem -pubin -in aes-key.txt -out aes-key.enc

# Encrypt the Lena image using the AES key
openssl aes-256-cbc -e -in lena512color.tiff -out lena_rsa.enc -pass file:./aes-key.txt
```

To decrypt the encrypted Lena image, *the rsa_decrypt.sh*. The script contains the following commands:

```
# Decrypt AES key
openssl rsautl -decrypt -inkey rsa-priv.pem -in aes-key.enc -out aes-key.dec

# Decrypt the encrypted Lena image using the decrypted AES key
openssl aes-256-cbc -d -in lena_rsa.enc -out lena_rsadec.tiff -pass file:./aes-key.dec
```

B. Using ECDSA

To generate a ECDSA signature on the Lena image, run the script *ecdsa_sign.sh* which contains the commands

```
# Generate EC private key
openssl ecparam -name secp256k1 -genkey -out ec-priv.pem

# Generate EC public key
openssl ec -in ec-priv.pem -pubout -out ec-pub.pem

# Generate a signature of the Lena image using the EC private key
openssl dgst -sha256 -sign ec-priv.pem lena512color.tiff > lena_signature.der
```

References:

[1] https://www.openssl.org/docs/man1.0.2/
[2] https://wiki.openssl.org/index.php/