Jose Louie Mark Año
2008-56492

CS 253
OpenSSL Project
May 24, 2017

Documentation

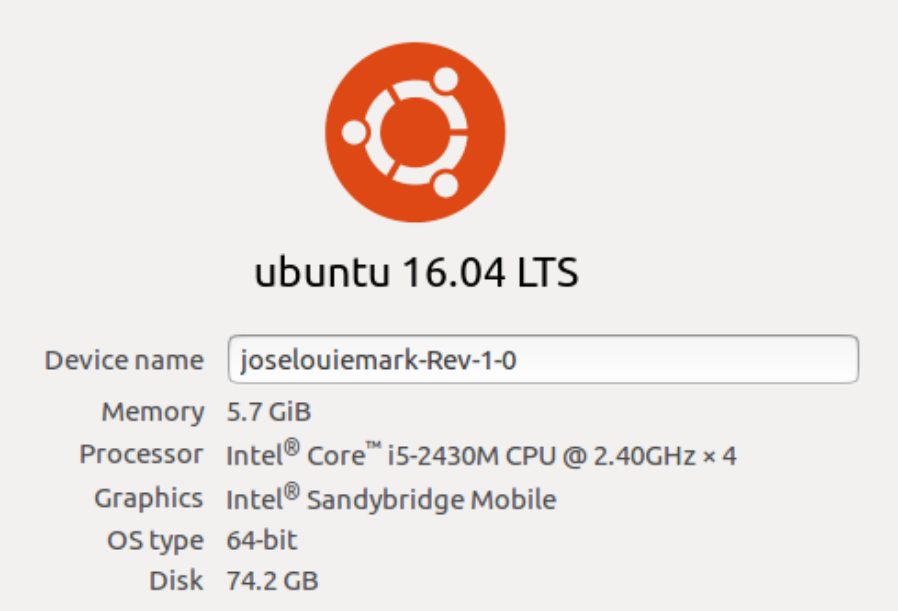
I. Introduction

This project is about cryptography exercises using OpenSSL, specifically symmetric encryption, hashing and public key encryption. First, the setup specifications are discussed, together with minimum requirements. Then it is followed by the exercises themselves. The first part of the exercise is the explanation of the file structure and setup of base folder. Together with this documentation are the files that contain the results and scripts which is within the *project* folder.

II. Setup

1. Specifications

The operating system is 64-bit Ubuntu 16.04 LTS. The operating system is ran in a Lenovo Z470 Laptop with i5-2430M @ 2.40GHz CPU, 6MB ram and allocated disk size of 6GB.



I chose the operating system since it already came with OpenSSL version 1.0.2g updated as of march 1, 2016.

```
OpenSSL> version
OpenSSL 1.0.2g  1 Mar 2016
OpenSSL>
```

The file used for this is *lena512color.tiff* provided in the spec. Also, in general, we would be using "*cs253*" as passwords.

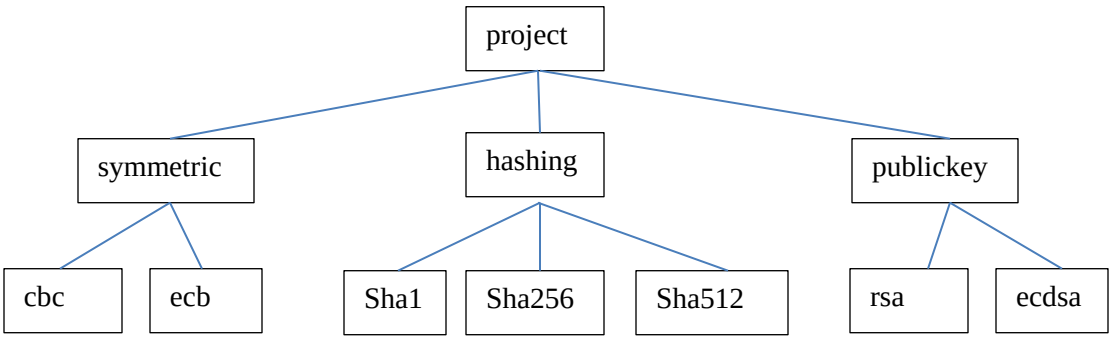
2. Minimum Specifications

This can be replicated in any operating system, as long as it ran OpenSSL with version 1.0.2 or above.

II. Exercises

1. Exercise File Structure

We would have a folder for each test. The base folder is the *project* folder. Within the *project* folder are three folders namely *symmetric*, *hashing* and *publickey*. Within *symmetric* folder are *cbc* and *ecb* folders, which contains cbc and ecb encryption tests respectively. Within *hashing* folder are *sha1*, *sha256* and *sha512* which contain tests for hashing sha-1, sha-256 and sha-512 respectively. Within *publickey* folder are *rsa* and *ecdsa* folders which contain tests for RSA encryption and ECDSA signature validation.



Throughout the excersise we will construct this file structure. So first we have to create the base *project* folder. Type in "*mkdir project*" then to go to it key in "*cd project*":

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253$ mkdir project
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253$ cd project
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$
```

2. SYMMETRIC

To do this exercise we first create *symmetric* folder by using "*mkdir symmetric*" and going to that folder using "*cd symmetric*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ mkdir symmetric
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ cd symmetric
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric$
```

2.1 Electronic Codebook (ECB) Exercise

2.1.1 Create ecb folder with test file

Create a *ecb* folder using "*mkdir ecb*" and go to that folder using command "*cd ecb*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric$ mkdir ecb
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric$ cd ecb
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$
```

Copy *lena512color.tiff* to the folder for our test file using "*cp [path to lena]* ."

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ cp [path of lena image] *
```

Check if the file is there using "*ls -lt*". The *lena512color.tiff* file should be there.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ ls -lt
total 772
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

2.1.2 Encrypt

Open OpenSSL by typing "*openssl*". Then key in the command "*aes-128-ecb -e -a -salt -in lena512color.tiff -out lena512color.tiff.enc*".

- *aes-128-ecb* is the encryption cipher to be used
- *-e* means encrypt
- *-a* means that the encrypted output will be base64 encoded
- *-salt* adds strength to the encryption and should always be used
- *-in* the file to be encrypted
- *-out* the file output encryption

Type password "*cs253*" twice.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ openssl
OpenSSL> aes-128-ecb -e -a -salt -in lena512color.tiff -out lena512color.tiff.enc
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
```

After that we exit OpenSSL by typing "*exit*". Check if encrypted file is there.

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ ls -lt
total 1816
-rw-rw-r-- 1 joselouiemark joselouiemark 1065180 May 23 23:54 lena512color.tiff.enc
-rw-rw-r-- 1 joselouiemark joselouiemark  786572 May 23 23:23 lena512color.tiff
```

If the reader wants to see the encrypted file, it is in /project/symmetric/ecb/lena512color.tiff.enc.

2.1.3 Decrypt

Open OpenSSL by typing "*openssl*". Then key in the command "*aes-128-ecb -d -a -salt -in lena512color.tiff -out lena512color.tiff.dec*".

- *aes-128-ecb* is the encryption cipher to be used
- *-d* means decrypt
- *-a* means that the encrypted output will be base64 encoded
- *-salt* adds strength to the encryption and should always be used
- *-in* the file to be encrypted
- *-out* the file output encryption

Type password "*cs253*" once.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ openssl
OpenSSL> aes-128-ecb -d -a -salt -in lena512color.tiff.enc -out lena512color.tiff.dec
enter aes-128-ecb decryption password:
OpenSSL>
```

Use "*exit*" to exit OpenSSL. Check if the file is there using "*ls -lt*". The *lena512color.tiff.dec* file should be there.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ ls -lt
total 2588
-rw-rw-r-- 1 joselouiemark joselouiemark  786572 May 24 00:05 lena512color.tiff.dec
-rw-rw-r-- 1 joselouiemark joselouiemark 1065180 May 24 00:02 lena512color.tiff.enc
-rw-rw-r-- 1 joselouiemark joselouiemark  786572 May 23 23:23 lena512color.tiff
```

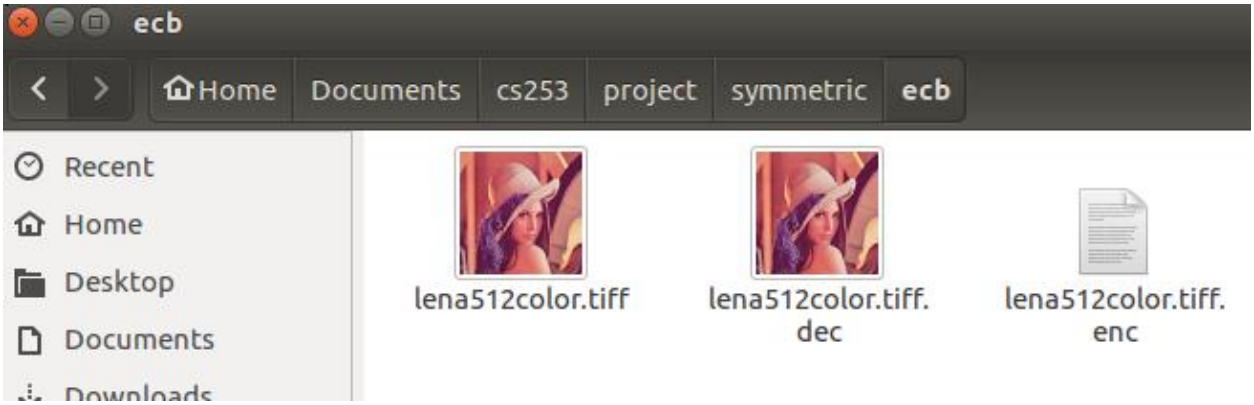
If the reader wants to see the encrypted file, it is in /project/symmetric/ecb/lena512color.tiff.dec.

2.1.3 Compare Decrypted file with Original file

For sanity check, we compare both files using diff command:

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$ diff lena512color.tiff lena512color.tiff.dec
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/ecb$
```

And by viewing the folder in the interface:



It seems like they are equal. Which means test was successful.

2.2 Cipher Block Chaining (CBC) Exercise

2.2.1 Create cbc folder with test file

Create a *cbc* folder using "*mkdir cbc*" and go to that folder using command "*cd cbc*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric$ mkdir cbc
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric$ cd cbc
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$
```

Copy *lena512color.tiff* to the folder for our test file using "*cp [path to lena]* ."

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ cp [lena image path] .
```

Check if the file is there using "*ls -lt*". The *lena512color.tiff* file should be there.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ ls -lt
total 772
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

2.2.2 Encrypt

Open OpenSSL by typing "*openssl*". Then key in the command "*aes-128-cbc -e -a -salt -in lena512color.tiff -out lena512color.tiff.enc*".

- *aes-128-cbc* is the encryption cipher to be used
- *-e* means encrypt
- *-a* means that the encrypted output will be base64 encoded
- *-salt* adds strength to the encryption and should always be used
- *-in* the file to be encrypted
- *-out* the file output encryption

Type password "*cs253*" twice.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ openssl
OpenSSL> aes-128-cbc -e -a -salt -in lena512color.tiff -out lena512color.tiff.enc
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
```

After that we exit OpenSSL by typing "*exit*". Check if encrypted file is there.

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ ls -lt
total 1816
-rw-rw-r-- 1 joselouiemark joselouiemark 1065180 May 24 00:15 lena512color.tiff.enc
-rw-rw-r-- 1 joselouiemark joselouiemark  786572 May 23 23:23 lena512color.tiff
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$
```

If the reader wants to see the encrypted file, it is in /project/symmetric/cbc/lena512color.tiff.enc.

2.2.3 Decrypt

Open OpenSSL by typing "*openssl*". Then key in the command "*aes-128-cbc -d -a -salt -in lena512color.tiff -out lena512color.tiff.dec*".

- *aes-128-cbc* is the encryption cipher to be used
- *-d* means decrypt
- *-a* means that the encrypted output will be base64 encoded
- *-salt* adds strength to the encryption and should always be used
- *-in* the file to be encrypted
- *-out* the file output encryption

Type password "*cs253*" once.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ openssl
OpenSSL> aes-128-cbc -d -a -salt -in lena512color.tiff.enc -out lena512color.tiff.dec
enter aes-128-cbc decryption password:
```

Use "*exit*" to exit OpenSSL. Check if the file is there using "*ls -lt*". The *lena512color.tiff.dec* file should be there.

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ ls -lt
total 2588
-rw-rw-r-- 1 joselouiemark joselouiemark  786572 May 24 00:19 lena512color.tiff.dec
-rw-rw-r-- 1 joselouiemark joselouiemark 1065180 May 24 00:15 lena512color.tiff.enc
-rw-rw-r-- 1 joselouiemark joselouiemark  786572 May 23 23:23 lena512color.tiff
```

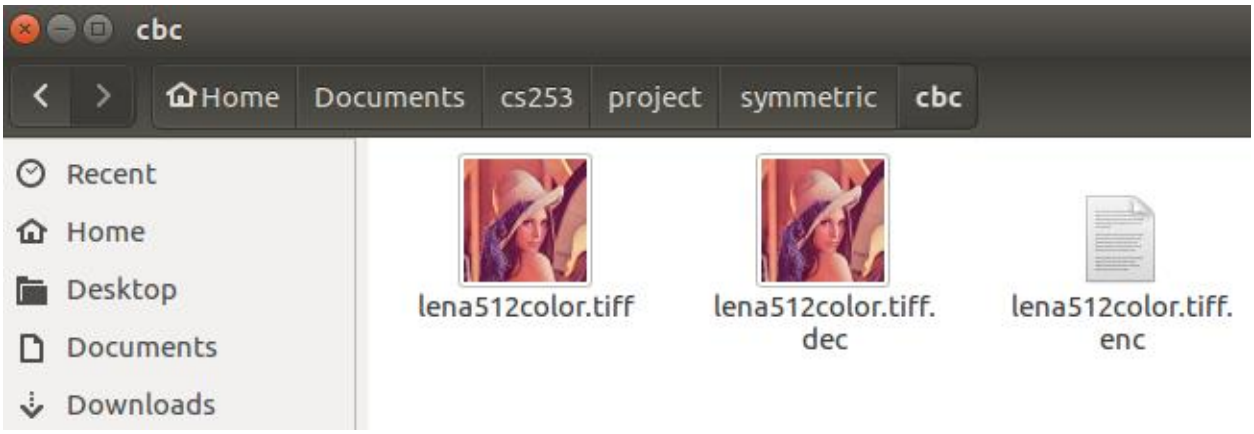
If the reader wants to see the encrypted file, it is in /project/symmetric/cbc/lena512color.tiff.dec.

2.1.3 Compare Decrypted file with Original file

For sanity check, we compare both files using diff command:

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$ diff lena512color.tiff lena512color.tiff.dec
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/symmetric/cbc$
```

And by viewing the folder in the interface:



It seems like they are equal. Which means test was successful.

3. HASHING

To do this exercise we first create *symmetric* folder by using "*mkdir hashing*" and going to that folder using "*cd hashing*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ mkdir hashing
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ cd hashing
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$
```

3.1 SHA1

3.1.1 Create sha1 folder with test file

Create a *sha1* folder using "*mkdir sha1*" and go to that folder using command "*cd sha1*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$ mkdir sha1
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$ cd sha1
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha1$
```

Copy *lena512color.tiff* to the folder for our test file using "*cp [path to lena]* ."

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha1$ cp [path of lena image] .
```

Check if the file is there using "*ls -lt*". The *lena512color.tiff* file should be there.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha1$ ls -lt
total 772
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

3.1.2 Create hash

Open OpenSSL by typing "*openssl*". Then key in the command "sha1 –out lena512color.tiff.enc lena512color.tiff".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha1$ openssl
OpenSSL> sha1 -out lena512color.tiff.hash lena512color.tiff
```

Exit OpenSSL. Check the hash produced by exiting OpenSSL and typing "cat lena512color.tiff.enc"

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha1$ ls -lt
total 776
-rw-rw-r-- 1 joselouiemark joselouiemark     66 May 24 01:12 lena512color.tiff.hash
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha1$ cat lena512color.tiff.hash
SHA1(lena512color.tiff)= e647d0f6736f82e498de8398eccc48cf0a7d53b9
```

This can be accessed through /project/hashing/sha256/lena512color.tiff.hash.

3.2 SHA256

3.2.1 Create sha256 folder with test file

Create a *sha256* folder using "*mkdir sha256*" and go to that folder using command "*cd sha256*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$ mkdir sha256
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$ cd sha256
```

Copy *lena512color.tiff* to the folder for our test file using "*cp [path to lena]* ."

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha256$ cp [file of lena image] .
```

Check if the file is there using "*ls -lt*". The *lena512color.tiff* file should be there.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha256$ ls -lt
total 772
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

3.2.2 Create hash

Open OpenSSL by typing "*openssl*". Then key in the command "sha256 –out lena512color.tiff.enc lena512color.tiff".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha256$ openssl
OpenSSL> sha256 -out lena512color.tiff.hash lena512color.tiff
```

Exit OpenSSL. Check the hash produced by exiting OpenSSL and typing "cat lena512color.tiff.enc"

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha256$ ls -lt
total 776
-rw-rw-r-- 1 joselouiemark joselouiemark     92 May 24 01:18 lena512color.tiff.hash
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha256$ cat lena512color.tiff.hash
SHA256(lena512color.tiff)= c056da23302d2fb0d946e7ffa11e0d94618224193ff6e2f78ef8097bb8a3569b
```

This can be accessed through /project/hashing/sha256/lena512color.tiff.hash.

3.3 SHA512

3.3.1 Create sha512 folder with test file

Create a *sha512* folder using "*mkdir sha512*" and go to that folder using command "*cd sha512*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$ mkdir sha512
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing$ cd sha512
```

Copy *lena512color.tiff* to the folder for our test file using "*cp [path to lena]* ."

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha512$ cp [path of lena image] .
```

Check if the file is there using "*ls -lt*". The *lena512color.tiff* file should be there.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha512$ ls -lt
total 772
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

3.3.2 Create hash

Open OpenSSL by typing "*openssl*". Then key in the command "sha512 –out lena512color.tiff.enc lena512color.tiff".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha512$ openssl
OpenSSL> sha512 -out lena512color.tiff.hash lena512color.tiff
```

Exit OpenSSL. Check the hash produced by exiting OpenSSL and typing "cat lena512color.tiff.enc"

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha512$ ls -lt
total 776
-rw-rw-r-- 1 joselouiemark joselouiemark    156 May 24 01:25 lena512color.tiff.hash
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/hashing/sha512$ cat lena512color.tiff.hash
SHA512(lena512color.tiff)= 2cb9d7df53eb8640dc48d736974f472a98d9c7186de7a972490455f5f3ed29dfc5b75c95ccb3ed4596bc2bfc4b1e52cf4d76bcee27d334dd155b
b426617392dc
```

This can be accessed through /project/hashing/sha256/lena512color.tiff.hash.

4. PUBLIC KEY ENCRYPTION

To do this exercise we first create *symmetric* folder by using "*mkdir symmetric*" and going to that folder using "*cd symmetric*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ mkdir publickey
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ cd publickey
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey$
```

4.1 RSA Exercise

4.1.1 Create rsa folder

Create a *rsa* folder using "*mkdir rsa*" and go to that folder using command "*cd rsa*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey$ mkdir rsa
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey$ cd rsa
```

4.1.2 Create private and public keys

Originally, did the keys are produced using:

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ openssl
OpenSSL> genrsa -out -des3 -out privatekey.pem 2048
Generating RSA private key, 2048 bit long modulus
..+++
.....................................................................................+++
e is 65537 (0x10001)
OpenSSL> exit
```

and

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ openssl
OpenSSL> rsa -des3 -in privatekey.pem -out public.pem -outform PEM -pubout
writing RSA key
```

But this has an error when encrypting the file:

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ openssl
OpenSSL> rsautl -encrypt -inkey public.pem -pubin -in lena512color.tiff -out lena512color.tiff.enc
RSA operation error
140249027139224:error:0406D06E:rsa routines:RSA_padding_add_PKCS1_type_2:data too large for key size:rsa_pk1.c:153:
error in rsautl
```

The error means that data was too large for key size

So used this command instead to create private and public keys "*req-x509 -nodes -days 100000 -newkey rsa:2048 -keyout privatekey.pem -out publickey.pem -subj '/'*"

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ openssl
OpenSSL> req -x509 -nodes -days 100000 -newkey rsa:2048 -keyout privatekey.pem -out publickey.pem -subj '/'
Generating a 2048 bit RSA private key
......+++
.....................................+++
writing new private key to 'privatekey.pem'
-----
```

- *-x509* output a x509 structure
- *-nodes* encrypt the output key
- *-days* number of days x509 generated key is valid
- *-newkey rsa:bits*, generate RSA key of 'bits' in size
- *-keyout* file to send key to, private key
- *-out* output file, public kay

Make sure primary key and private keys are created:

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ ls -lt
total 8
-rw-rw-r-- 1 joselouiemark joselouiemark 1704 May 24 02:36 privatekey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark 1042 May 24 02:36 publickey.pem
```

This can be accessed through /project/publickey/rsa/privatekey.pem.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ cat privatekey.pem
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDCgV/rtuUlyXxl
Y5FzAayHmAYMhaswK9YOnGB/KcQeG1e4sR0o8wDm3Vtug8v9fNzcnEH9hZvmfWrH
HK1uYeSFJ/Vp0DUYUN1rjVUaMV68JiUxrxTnJ0OLI5vKm+G/bkwBP3vYWMdERfTN
BPJcU7wXjiETAqRVDq3tvLomTYwPWB25ZEnSY6L85BdrtN1Aw74PDcq1Jk+flLBl
dDGZ7Ja2XaDSXhwhmM40+8u0u2S5qcrP7icCxzJBcJ/sFJWmpa2yJPVAowD4bIsT
rakUwbjOSMyJC6vS+e073w5XhgNvFJDPfuruyEQ6kbs9dwybBXqB1uFO+ff4b/RP
68XZnvYBAgMBAAECggEAYl05HCSfV+/AaCnlaarkDKKWFIkY3MCefT8aEdUWaD9d
qnpvi8DwqNAzJ/2hEO2vaCXhvYtvHy437MAU+Siqf6iIlN2FrGlgwL511kw3Y6FK
2Ah54rhZkSkN5XXVwZDPASaQxAWJqH5UYcZppsV17tei5FFGt/k3fm4eGbbH6Q29
izc1+46OVQX9t5GFu10wE6LrGIjejL0N1i2bBi7i0iXlDkZKJwl1J5iwIXae6eXl
7RoVlnBuzcj/8+cTFaQ3on9W9kCORA9GgRFflbBMUXBk7khXJusVS39cz9xtrO6L
fQEqFB9zkYkowSNx/vci5nce0lRKdawnL4eL5CcAAQKBgQDwoquk4MQAH+jGqrjd
c42hCwNZ5oSdapJc//Zx+qvz+26Sp1Pi4/A1ev5WiPB2Yxn9Nh3ck2W3HCPs15Xp
x2bmdIoWhOIJvTMLVUAoN/1NQWobFxl7PHiC365s4rWvZKC0VR/l06e6d7MFUD0x
4ZR1x2XtwZrIU2MAa2+fQbh6AQKBgQDO7K5M2itF/eg1QHAfAAlfNnS/5AkPN05j
aB8oVuLTjdPFsQj4D/VDuuUfA+ZNZQnEohYIutRl/NmY0aCEvHgO4E5CQvu/+d/f
Mhjff2Q3YjoyoCJUrj9jX3+faliD+0fehUm7TsI2uC6RY6xwwRRgaKSNzJPJ9Qlk
mug/EM58AQKBgEkTPUi+sQbQqeHa6nTlqHblX0knxdIAER6I5JBMbXeUPUnv4nue
eM3k4ePphgyNn9DTZFIduN/Yfdhox6/MzCTj8yNPi00l1aB1bRxJlHSnjs9XPne8
L2SfBNswJbhzuJHK3/rtQZxQyV6ttUe2l10FBqEjEBwRppj0B1u3m4wBAoGADMst
a6sBie5M8R3u0Lvb2HsxdLvQo5Qz0Imbg477r0DRmyN6/nteGrXvfJ9tGdPparLq
0ddbUoLzoTM0zYApnCHaNNaXYiM86DvctdTWiWYVWxPzi3DKkoZRCbY1itN6ehOb
n20bLO7uUibrfn0xmVRe+DVx3ca8se+x9FRBsAECgYEA4qCCXV6gsEYCFB5LSyUV
Pg7n5VK2SYZTym6n+vy4ade3o5fKnLr1M/vmOsGSkErX5tNZKUwbWVRm7I4upj/X
C+tfCS013qvPvinXkXyGQI8PHNI13NWG5bm7lSc8rTd1s1eXBPIecHA2DzBO6cEX
A0R1lKKBKn1UEQggsVcICHI=
-----END PRIVATE KEY-----
```

This can be accessed through /project/publickey/rsa/publickey.pem.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ cat publickey.pem
-----BEGIN CERTIFICATE-----
MIIC1TCCAb2gAwIBAgIJAP9sPhEhZQ81MA0GCSqGSIb3DQEBCwUAMAAwIBcNMTcw
NTIzMTgzNjA4WhgPMjI5MTAzMDgxODM2MDhaMAAwggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQDCgV/rtuUlyXxlY5FzAayHmAYMhaswK9YOnGB/KcQeG1e4
sR0o8wDm3Vtug8v9fNzcnEH9hZvmfWrHHK1uYeSFJ/Vp0DUYUN1rjVUaMV68JiUx
rxTnJ0OLI5vKm+G/bkwBP3vYWMdERfTNBPJcU7wXjiETAqRVDq3tvLomTYwPWB25
ZEnSY6L85BdrtN1Aw74PDcq1Jk+flLBldDGZ7Ja2XaDSXhwhmM40+8u0u2S5qcrP
7icCxzJBcJ/sFJWmpa2yJPVAowD4bIsTrakUwbjOSMyJC6vS+e073w5XhgNvFJDP
furuyEQ6kbs9dwybBXqB1uFO+ff4b/RP68XZnvYBAgMBAAGjUDBOMB0GA1UdDgQW
BBQeDQflR6hviMtpgFN/oG0tnQWZhjAfBgNVHSMEGDAWgBQeDQflR6hviMtpgFN/
oG0tnQWZhjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQCGm4LfH+i4
pMmx/ytY9xjFbtnmqAzrZJrza9gnUC/CaxNpS4jlH62wOJTBmKAPeQDv0Rh3NbUY
n6XfOZ5w/RUuEnmPcp+eEyRPX9HifALUz+cY62IlrnJra6+p9RNicyQ0FgZroAyn
dpUWLIZpy1dstxXEuKhB//OonxxcL2rTTTcjzACJNUuVh5YeBvm7Ik5NjcqYc+Fg
nyVIVxYh/Q3rHH2v6rItW9o3WChpEAUB7vpjkhbH0NvPRfxN1omsN75I2bf5sOyh
FLvL5Dcg1PcC+dgPmIS7dHoThO0zI6HgH6srpXmi6/P3Q1CURs2C8RAbcySfP+gB
2pi9qPECQAzx
-----END CERTIFICATE-----
```

4.1.3 Encrypt using public key

After both primary and private keys are generated, place the *lena512color.tiff* file to be encrypted.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ cp [path of lena image] .
```

Then make sure it is in the folder together with public key and private key.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ ls -lt
total 780
-rw-rw-r-- 1 joselouiemark joselouiemark   1704 May 24 02:36 privatekey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark   1042 May 24 02:36 publickey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

Then do the encryption using "*smime -encrypt -aes -in lena512color.tiff -binary -outform DEM -out lena512color.tiff.enc publickey.pem*" command.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ openssl
OpenSSL> smime -encrypt -aes256 -in lena512color.tiff -binary -outform DEM -out lena512color.tiff.enc publickey.pem
```

- *-encrypt* encrypt flag
- *-aes128* encrypt REM output with CBC AES
- *-out* output file
- *-binary* don't translate message to text
- *-outform* output format
- *-in* input file

Exit OpenSSL and check output file

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ ls -lt
total 1552
-rw-rw-r-- 1 joselouiemark joselouiemark 786958 May 24 02:39 lena512color.tiff.enc
-rw-rw-r-- 1 joselouiemark joselouiemark   1704 May 24 02:36 privatekey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark   1042 May 24 02:36 publickey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

lena512color.tiff.enc encrypted output can be accessed through /project/publickey/rsa/ lena512color.tiff.enc.

4.1.4 Decrypt using private key

Then do the encryption using "*smime -decrypt -in lena512color.tiff -binary -inform DEM -out lena512color.tiff.enc publickey.pem*" command.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ openssl
OpenSSL> smime -decrypt -in lena512color.tiff.enc -binary -inform DEM -inkey privatekey.pem -out lena512color.tiff.dec
```

- *-decrypt* decrypt flag
- *-out* output file
- *-binary* don't translate message to text
- *-inform* input format
- *-inkey* private key
- *-in* input file

Exit OpenSSL and check output file

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ ls -lt
total 2324
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 24 02:41 lena512color.tiff.dec
-rw-rw-r-- 1 joselouiemark joselouiemark 786958 May 24 02:39 lena512color.tiff.enc
-rw-rw-r-- 1 joselouiemark joselouiemark   1704 May 24 02:36 privatekey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark   1042 May 24 02:36 publickey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

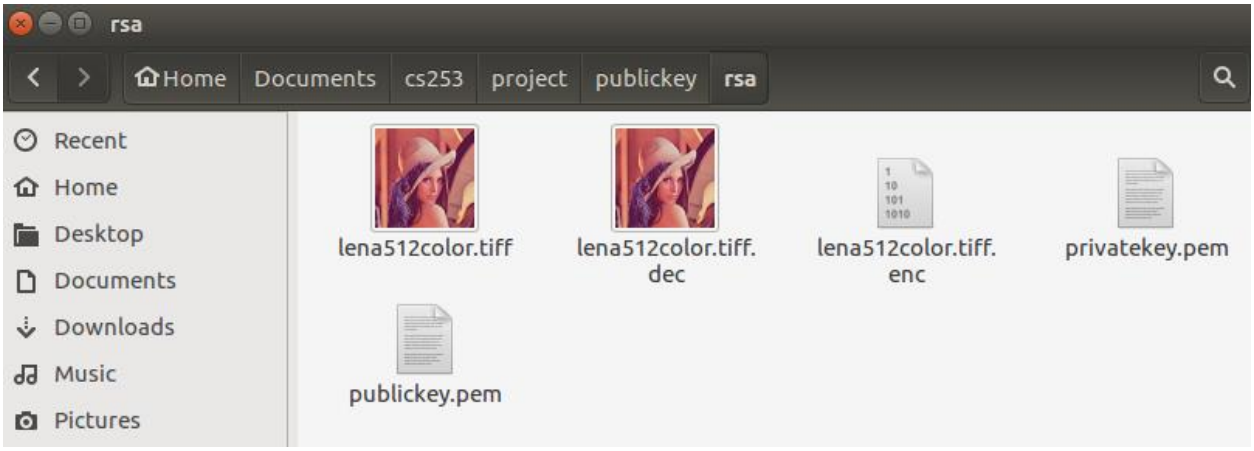
lena512color.tiff.dec decrypted output can be accessed through /project/publickey/rsa/ lena512color.tiff.dec.

4.1.5 Compare Decrypted File to Original File

For sanity check we compare decrypted file to original file using "*diff* " command.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$ diff lena512color.tiff.dec lena512color.tiff
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/rsa$
```

And we also check in the interface folder.



It seems like they are equal. Which means test was successful.

4.2 ECDSA Exercise

4.2.1 Create ecdsa folder

Create a *ecdsa* folder using "*mkdir ecdsa*" and go to that folder using command "*cd ecdsa*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ mkdir publickey
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project$ cd publickey
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey$
```

4.2.2 Create private key

Create private key using command "*ecparam -genkey name secp384r1 -out privatekey.pem*"

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ openssl
OpenSSL> ecparam -genkey -name secp384r1 -out privatekey.pem
```

Check if private key is really created.

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ ls -lt
total 4
-rw-rw-r-- 1 joselouiemark joselouiemark 359 May 24 02:55 privatekey.pem
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ vi privatekey.pem
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ cat privatekey.pem
-----BEGIN EC PARAMETERS-----
BgUrgQQAIg==
-----END EC PARAMETERS-----
-----BEGIN EC PRIVATE KEY-----
MIGkAgEBBDCCNP90sNLDPC77NXW9sA7SJtAtkQQnZ38ZAnJCot4FB4WQovYx+PT/
o0LuoLL/PXCgBwYFK4EEACKhZANiAAQErkMBX/ZC0MbRzNyCtMtUZs8nK8g7aCkH
kjIiFClWaBtJHTqv9USQwegw0KlifAgrWa3+bma5fvPDaaP9i0CO9rblvucRwBsT
/LgB6f66y4xfTjOlRTSIZQlQ9ZPRL/U=
-----END EC PRIVATE KEY-----
```

4.2.2 Create public key based on private key

Create public key using command "*ec -in privatekey.pem -pubout -out privatekey.pem*"

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ openssl
OpenSSL> ec -in privatekey.pem -pubout -out publickey.pem
read EC key
writing EC key
```

Check if public key is really created.

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ ls -lt
total 8
-rw-rw-r-- 1 joselouiemark joselouiemark 215 May 24 02:57 publickey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark 359 May 24 02:55 privatekey.pem
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ cat publickey.pem
-----BEGIN PUBLIC KEY-----
MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEBK5DAV/2QtDG0czcgrTLVGbPJyvIO2gp
B5IyIhQpVmgbSR06r/VEkMHoMNCpYnwIK1mt/m5muX7zw2mj/YtAjva25b7nEcAb
E/y4Aen+usuMX04zpUU0iGUJUPWT0S/1
-----END PUBLIC KEY-----
```

4.2.3 Create signature using privatekey and sha256

Create signature using command "*dgst -sha256 -sign privatekey.pem -out signature.bin lema512color.tiff*".

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ openssl
OpenSSL> dgst -sha256 -sign privatekey.pem -out signature.bin lena512color.tiff
```

Check if signature is created.

```
OpenSSL> exit
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ ls -lt
total 784
-rw-rw-r-- 1 joselouiemark joselouiemark    103 May 24 03:05 signature.bin
-rw-rw-r-- 1 joselouiemark joselouiemark    215 May 24 02:57 publickey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark    359 May 24 02:55 privatekey.pem
-rw-rw-r-- 1 joselouiemark joselouiemark 786572 May 23 23:23 lena512color.tiff
```

4.2.4 Verify signature using publickey and sha256

Verify signature using command “dgst -sha256 -verify publickey.pem -signature signature.bin lena512color.tiff”

```
joselouiemark@joselouiemark-Rev-1-0:~/Documents/cs253/project/publickey/ecdsa$ openssl
OpenSSL> dgst -sha256 -verify publickey.pem -signature signature.bin lena512color.tiff
Verified OK
OpenSSL>
```

With the result, the test is successful.