JOHN PATRICK S. LA-ANAN CS253 PROJECT

NOTES:

Operating System Used: Windows 7

- Added OpenSSL bin folder in the *Path* environment variable

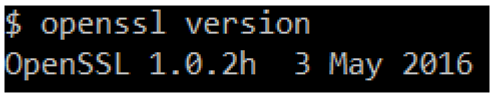
OpenSSL Version: OpenSSL 1.1.0 from (https://www.openssl.org/source/openssl-1.1.0e.tar.gz)

Script running: Used Docker command line (to be able to execute Linux commands) to run the scripts in the *scripts.txt* file

1. OPENSSL

Download and install OpenSSL, if you do not have it yet. https://www.openssl.org

STEP 1: Install OpenSSL and get its version.



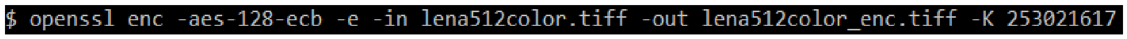
```
$ openssl version
OpenSSL 1.0.2h  3 May 2016
```

2. SYMMETRIC ENCRYPTION

Study the OpenSSL Library and use it to perform symmetric AES encryption on the 512x512 Color (24-bit) Lena image (http://www.ece.rice.edu/~wakin/images/lena512color.tiff)

Use both ECB and CBC mode, for AES-128.

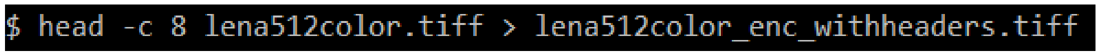
a) **AES-128 ECB**
STEP 1: Encrypt the file using *enc* with option *aes-128-ecb* to generate *lena512color_enc.tiff* file. The key used *253021617*.



```
$ openssl enc -aes-128-ecb -e -in lena512color.tiff -out lena512color_enc.tiff -K 253021617
```

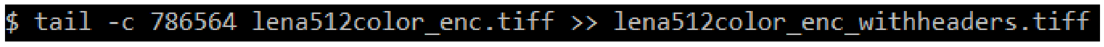
STEP 2: [1], [2] A TIFF file begins with an 8-byte information about its header. To generate the encrypted image file, the first 8 bytes of the original Lena image file should be extracted and append the last 786,564 bytes of the encrypted file (*lena512color_enc.tiff*). The 786,564 is the size of the Lena image file (*786,572 bytes*) less the 8 bytes of header information.

Get the header information and save to *lena_encrypted.tiff*



```
$ head -c 8 lena512color.tiff > lena512color_enc_withheaders.tiff
```

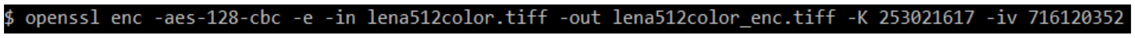
STEP 3: Get the last 786,564 bytes of the encrypted file and append to *lena_encrypted.tiff*



```
$ tail -c 786564 lena512color_enc.tiff >> lena512color_enc_withheaders.tiff
```

b) **AES-128 CBC**

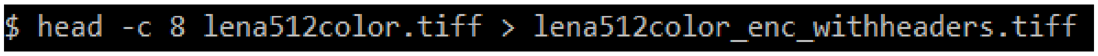
STEP 1: Encrypt the file using *enc* with option *aes-128-cbc* to generate *lena512color_enc.tiff* file. The key used *253021617* and the initialization vector *iv* is *716120352*.



```
$ openssl enc -aes-128-cbc -e -in lena512color.tiff -out lena512color_enc.tiff -K 253021617 -iv 716120352
```

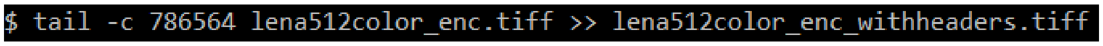
STEP 2: [1], [2] A TIFF file begins with an 8-byte information about its header. To generate the encrypted image file, the first 8 bytes of the original Lena image file should be extracted and append the last 786,564 bytes of the encrypted file (*lena512color_enc.tiff*). The 786,564 is the size of the Lena image file (*786,572 bytes*) less the 8 bytes of header information.

Get the header information and save to *lena_encrypted.tiff*



```
$ head -c 8 lena512color.tiff > lena512color_enc_withheaders.tiff
```

STEP 3: Get the last 786,564 bytes of the encrypted file and append to *lena_encrypted.tiff*



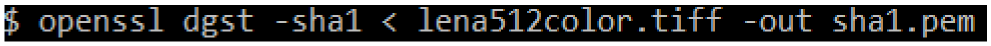
```
$ tail -c 786564 lena512color_enc.tiff >> lena512color_enc_withheaders.tiff
```

3. HASHING

Using OpenSSL, hash the same Lena 512x512 image using the following hash functions: SHA-1, SHA-256, SHA-512. Again, fully document the process and the results of the hash.

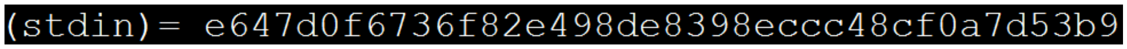
a) **SHA-1**

STEP 1: Generate hash and then output to sha1.pem.



```
$ openssl dgst -sha1 < lena512color.tiff -out sha1.pem
```

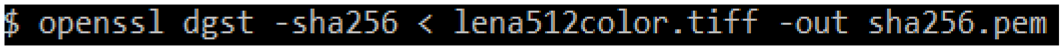
sha1.pem file contains the hash:



```
(stdin)= e647d0f6736f82e498de8398eccc48cf0a7d53b9
```

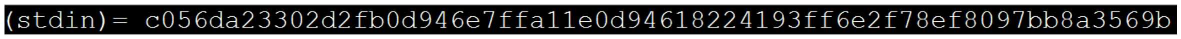
b) **SHA-256**

STEP 1: Generate hash and then output to sha256.pem.



```
$ openssl dgst -sha256 < lena512color.tiff -out sha256.pem
```

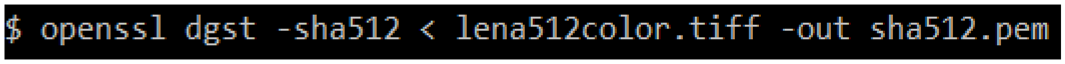
sha256.pem file contains the hash:



```
(stdin)= c056da23302d2fb0d946e7ffa11e0d94618224193ff6e2f78ef8097bb8a3569b
```

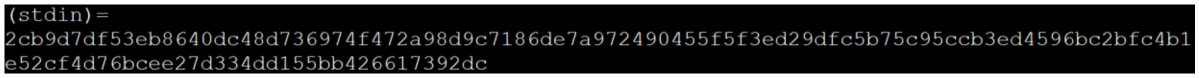
c) **SHA-512**

STEP 1: Generate hash and then output to sha512.pem.



```
$ openssl dgst -sha512 < lena512color.tiff -out sha512.pem
```

sha512.pem file contains the hash:

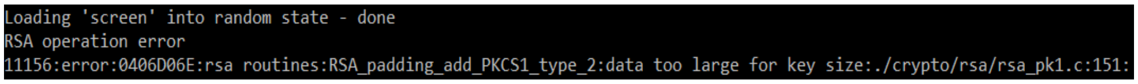


```
(stdin)=
2cb9d7df53eb8640dc48d736974f472a98d9c7186de7a972490455f5f3ed29dfc5b75c95ccb3ed4596bc2bfc4b1
e52cf4d76bcee27d334dd155bb426617392dc
```

4. PUBLIC KEY ENCRYPTION

a) Using OpenSSL, perform an RSA encryption on the Lena 512x512 image, using RSA-2048

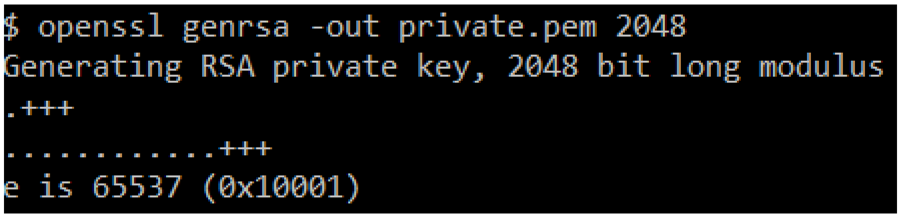
To perform RSA encryption, *rsautl* is used. However, it cannot be used for large files. The following error is displayed:



```
Loading 'screen' into random state - done
RSA operation error
11156:error:0406D06E:rsa routines:RSA_padding_add_PKCS1_type_2:data too large for key size:./crypto/rsa/rsa_pk1.c:151:
```

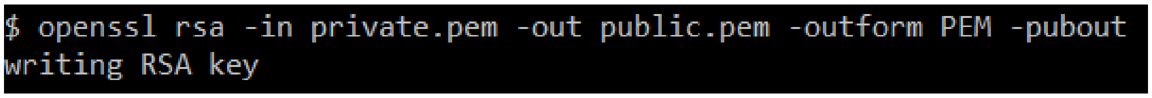
To do the RSA encryption for large files, the key first will be encrypted using *rsautl* and then encrypt the file using AES.

STEP 1: Generate the private key keys using *genrsa*.



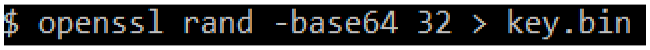
```
$ openssl genrsa -out private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
............+++
e is 65537 (0x10001)
```

STEP 2: Generate the public key using the generated private key.



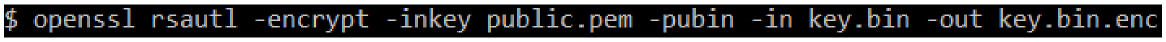
```
$ openssl rsa -in private.pem -out public.pem -outform PEM -pubout
writing RSA key
```

STEP 3: Generate a 256-bit random key



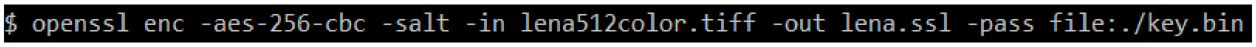
```
$ openssl rand -base64 32 > key.bin
```

STEP 4: Encrypt the 256-bit key using the public key.



```
$ openssl rsautl -encrypt -inkey public.pem -pubin -in key.bin -out key.bin.enc
```

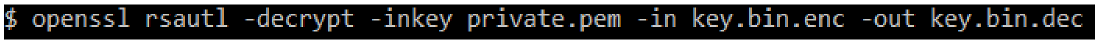
STEP 5: Encrypt the Lena image file using AES 256 CBC.



```
$ openssl enc -aes-256-cbc -salt -in lena512color.tiff -out lena.ssl -pass file:./key.bin
```

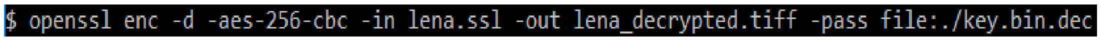
DECRYPTION:

STEP 1: Decrypt the key using *rsautl*.



```
$ openssl rsautl -decrypt -inkey private.pem -in key.bin.enc -out key.bin.dec
```

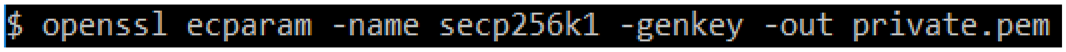
STEP 2: Decrypt file using the decrypted key



```
$ openssl enc -d -aes-256-cbc -in lena.ssl -out lena_decrypted.tiff -pass file:./key.bin.dec
```

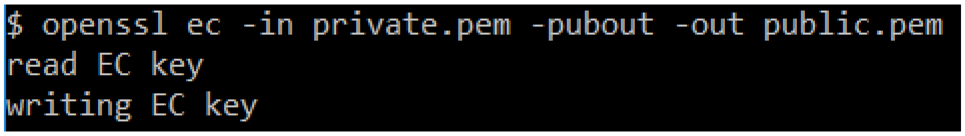
b) Using OpenSSL, generate an ECDSA signature on the same Lena image. If you need to use a hash function, use SHA-256

STEP 1: Generate a random key using *ecparam* to be the private key. The curve used is *secp256k1*, SECG curve over a 256-bit prime field. (*openssl ecparam –list_curves*).



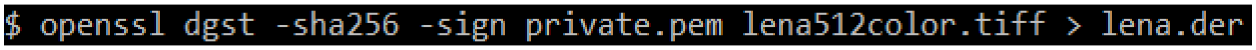
```
$ openssl ecparam -name secp256k1 -genkey -out private.pem
```

STEP 2: Generate the public key using the private key generated in Step 1



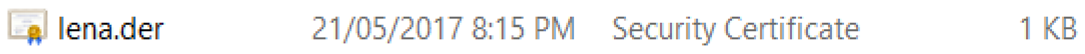
```
$ openssl ec -in private.pem -pubout -out public.pem
read EC key
writing EC key
```

STEP 3: Sign the Lena image file using the generated private and public key.



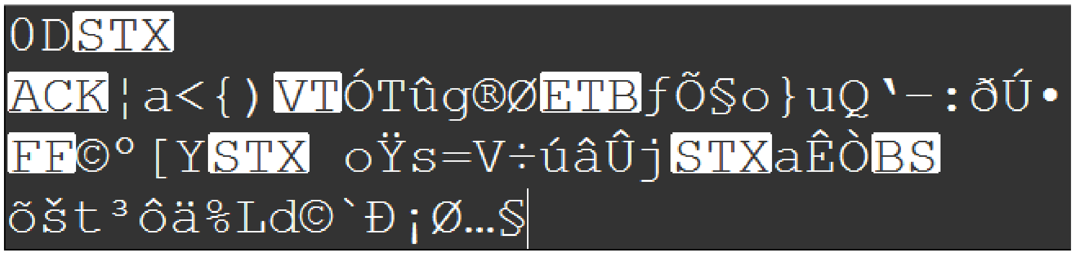
```
$ openssl dgst -sha256 -sign private.pem lena512color.tiff > lena.der
```

It will generate the file lena.der



```
lena.der          21/05/2017 8:15 PM    Security Certificate         1 KB
```

Containing the following file (view using Notepad++)



```
0DSTX
ACK¦a<{)VTÓTûg®ØETBƒÕ§o}uQ'−:ðÚ•
FF©°[YSTX oŸs=V÷úâÛjSTXaÊÒBS
õšt³ôä%Ld©`Đ¡Ø…§
```

REFERENCES:

[1] https://tools.ietf.org/html/rfc2306 (Last accessed May 22, 2017)

[2] http://paulbourke.net/dataformats/tiff/ (Last accessed May 22, 2017)