**DDC Cryptography Project.**

*Due Saturday 20 February 2021.*

Follow the detailed submission instructions at the end of this specs sheet.

Cryptography Exercises using OpenSSL

1. Download and install OpenSSL, if you do not have it yet.

[https://www.openssl.org](https://www.openssl.org/)

For each of the following exercises, you are given the freedom on how you would use OpenSSL (i.e., accessed via a shell script, or called from within your own program, etc.).

Reference all sources that you use in your answers.

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.

Fully document the process (in a document) of how you performed the encryption using OpenSSL, and the results of the encryption.

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.

4. PUBLIC KEY ENCRYPTION.

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

Using OpenSSL, generate an ECDSA signature on the same Lena image.

If you need to use a hash function, use SHA-256.

For all other details not outlined in this spec sheet, you have the freedom to choose or decide on the design detail. For example, you can define your own passwords or passphrases as basis for key generation.

TO BE SUBMITTED:

1. All source code, scripts, and documentation (in PDF) are to be housed in a git repository. You may use a public git repository (e.g., create an account on [github.com](http://github.com/) or [bitbucket.com](http://bitbucket.com/)) or your own private git repository.

2. Submit an accessible link to your git repository via e-mail to [roselia,delacruz@bulsu.edu.ph](mailto:spfestin@dcs.upd.edu.ph" \t "_blank) by the deadline. I should be able to clone your git repo given the link and run (compile) your code/scripts from my machine. You should provide sufficient documentation for me to replicate your environment, e.g., what operating system, programming language, etc.

Cryptography Exercises using OpenSSL

**1. Download and install OpenSSL, if you do not have it yet.**

Just followed the instruction from this site, <https://tecadmin.net/install-openssl-on-windows/>.

Then downloaded Win64 OpenSSL v1.1.1i (EXE) and installed it. Then add the needed path in environment variables.

**2. SYMMETRIC ENCRYPTION.**

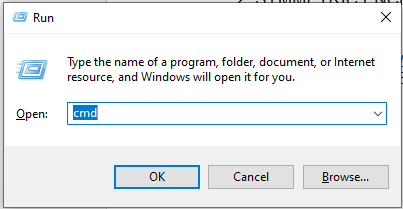
Study the OpenSSL Library and use it to perform symmetric AES encryption on the 512x512 Color (24-bit) Lena image (<http://www.ece.rice.edu/~wakin/images/lena512color.tiff>)

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

C:\Users\Carlo\Downloads\lena512color.tiff

Before going to command prompt, download the tiff file in the link then drag a copy of file in desktop for the openssl find it easily.

1. Run cmd



1. Then change directory to desktop and command openssl.

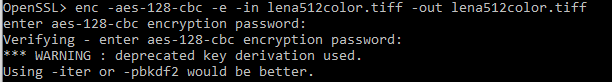


Using my brother’s pc to run this command.

1. Then type this for aes-128-cbc line of code.



Then press enter, and your password for that file is needed and the password for this is: pass



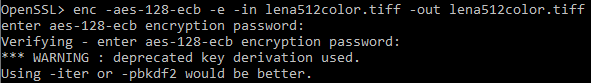
The output image should be this:



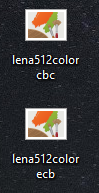
1. Then for aes-128-ecb, drag another copy of the tiff image on the desktop. then type this command.



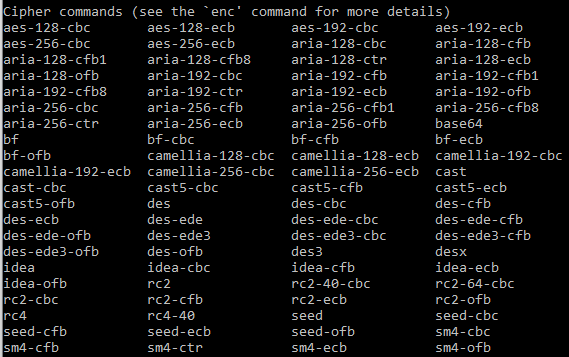
Then press enter and add your password. the pass for this is: pass



So the output is the same as cbc.



The commands were found at “help” command:



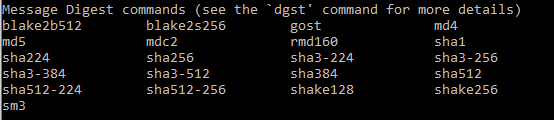
References:

Jaken H. (2018 )“*A Walkthrough on AES Image Encryption* “Retrieved from [https://medium.com/@  
JakenH/today-im-going-teach-you-how-to-encrypt-a-simple-picture-that-you-can-download-here-141ce9b0b738](https://medium.com/@JakenH/today-im-going-teach-you-how-to-encrypt-a-simple-picture-that-you-can-download-here-141ce9b0b738)

**3. HASHING.**

For hashing, using OpenSSL, hash the same Lena 512x512 image using the following hash functions:

SHA-1, SHA-256, SHA-512.

The commands were found at “help” command. 

**For SHA-1:**

Type this command:



Then the output is:



**For SHA-256:**

Type this command:



Then the output is:



**For SHA-512:**

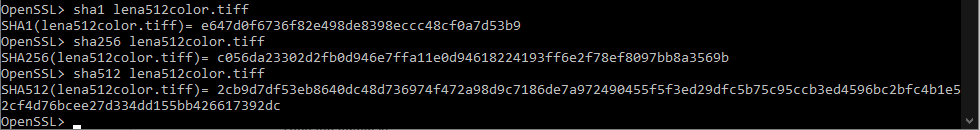
Type this command:



Then the output is:



Overall documentation for Hashing:



References: OSXDaily (n.a.) (2012) “Verify SHA1 Hash with openssl” Retrieved from <https://osxdaily.com/2012/02/09/verify-sha1-hash-with-openssl/>

4. PUBLIC KEY ENCRYPTION.

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

C:\Users\Carlo\Downloads\lena512color.tiff

Using OpenSSL, generate an ECDSA signature on the same Lena image.

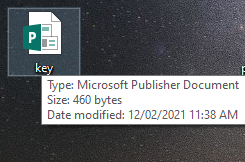
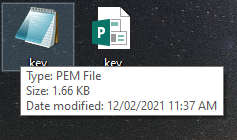
If you need to use a hash function, use SHA-256.

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

Before encrypting we need to type these commands to generate keys for private and for public:



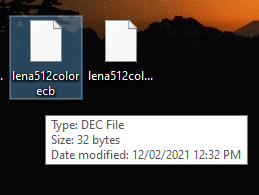
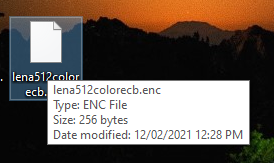




Private key Public key

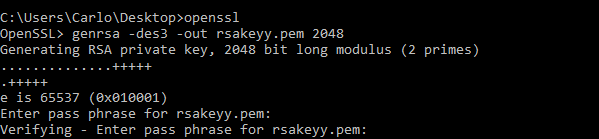
Encrypt and Decrypt using RSAUTL command:

No description available.



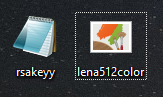
ENCRYPTED File DECRYPTED File

Encrypt using RSA command:

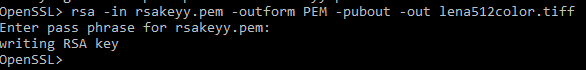


After applying the command, the output would be this:

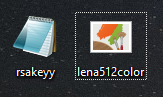
It is named: rsakeyy.pem



Encrypt using RSA command:



Then the output would be this:

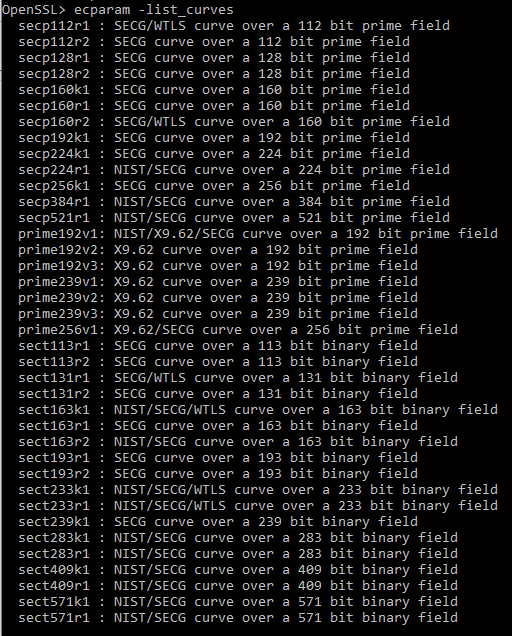


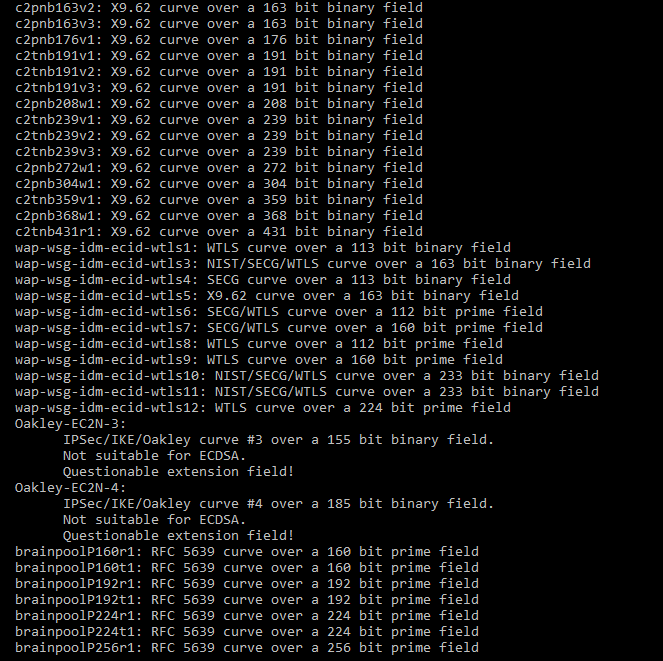
Using OpenSSL, generate an ECDSA signature on the same Lena image.

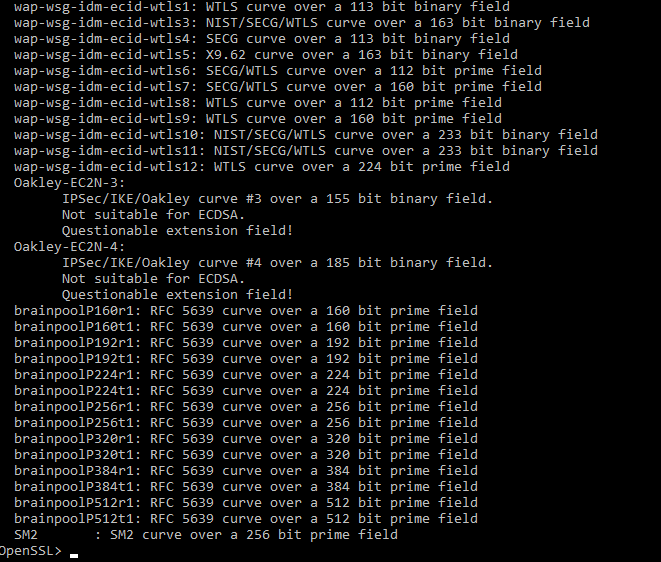
C:\Users\Carlo\Downloads\lena512color.tiff

If you need to use a hash function, use SHA-256.

At first you need to find the supporting code for ECDSA by typing the ecparam list\_curves:

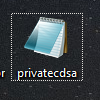




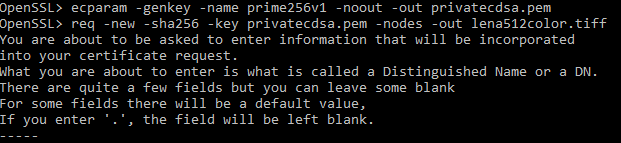


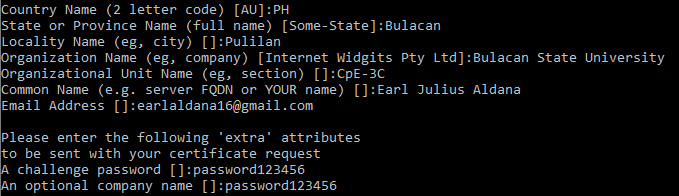
Then type this code:

ecparam -genkey -name prime256v1 -noout -out privatecdsa.pem



Then type the following command:





The image changed again using ECDSA command:



References:

In openssl, just typed the command rsa -help and rsautl –help if need assistance to code.

<https://www.openssl.org/docs/manmaster/man1/openssl-rsautl.html?fbclid=IwAR1Ekn0TXFUaJ-0JLNIsJqE30f32GPXZ60_t0gS_tiHon0udoQWjjASwrd0>

<https://www.openssl.org/docs/manmaster/man1/openssl-rsa.html?fbclid=IwAR3xDR6NQKjLy1H-ip48s6ESJnScdkf085rAFE5JJrvGIjMo7TMfi-vWxFk>

<https://gist.github.com/thinkerbot/706137>