# CRYPTOGRAPHY PROGRAM USING PYTHON AND OPENSSL

Submitted by:

Rahino C. Qujano

## Machine's Operating System during Development

- Windows 10 Home Single Language

## Machine's Essential Software to install dependent programs and to run the code itself

- Python 3.9.1 – For running the program source code
- Chocolatey v0.10.15 – For installing OpenSSL
- OpenSSL 1.1.1i 8 Dec 2020 – For providing command line cryptography tools

## Installing program dependencies

1. Install Python 3.9.x from https://www.python.org/downloads/
2. Verify Python from shell using the command
   >> python --version
3. Install Chocolatey by referring to https://chocolatey.org/install
4. Verify Chocolatey from shell using the command
   >> choco –version
5. Install OpenSSL by typing the given command on an administrative shell
   >> choco install openssl
6. Verify OpenSSL from shell using the command
   >>openssl version

## Code Tour

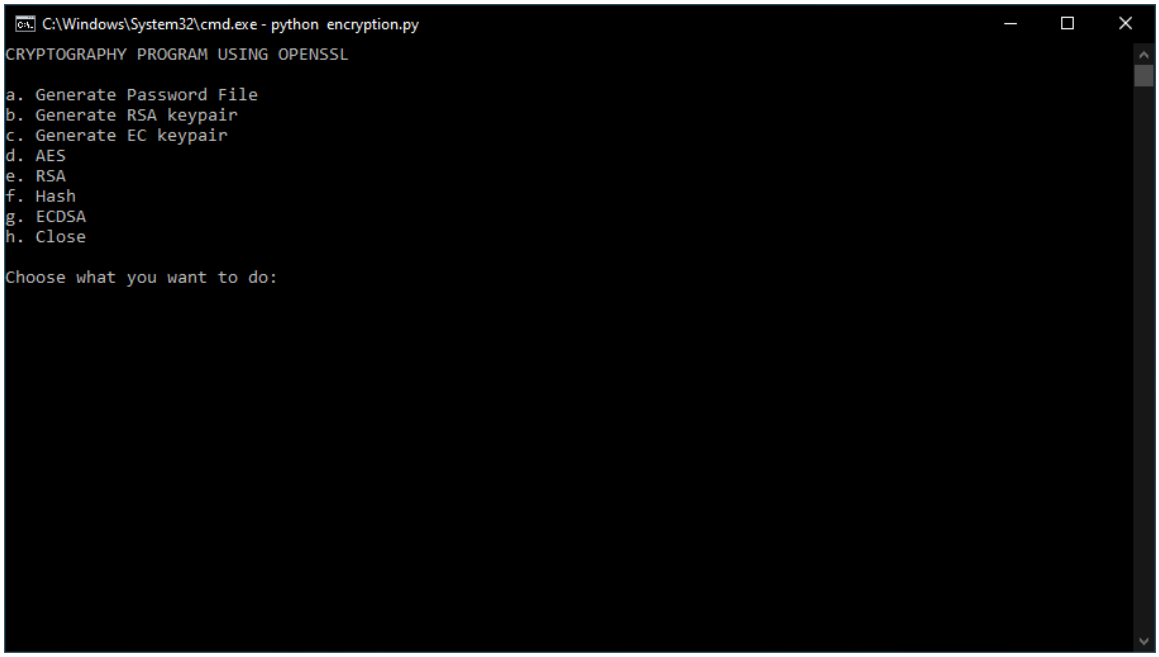
The program's source code was organized into two main sections. The first section is the Primitive Operations designated by a single line comment which contains multiple functions that encapsulates a particular OpenSSL cryptographic operation like AES encryption, RSA encryption, Hashing and more. The second section is the Simplified Operations which is primarily dominated by a single function called "interface" that provides a user interface and rudimentary error handling. Interface is then called at the main function which is then called after a standalone-use checking. The program imports two standard python modules namely "os" and "subprocess".

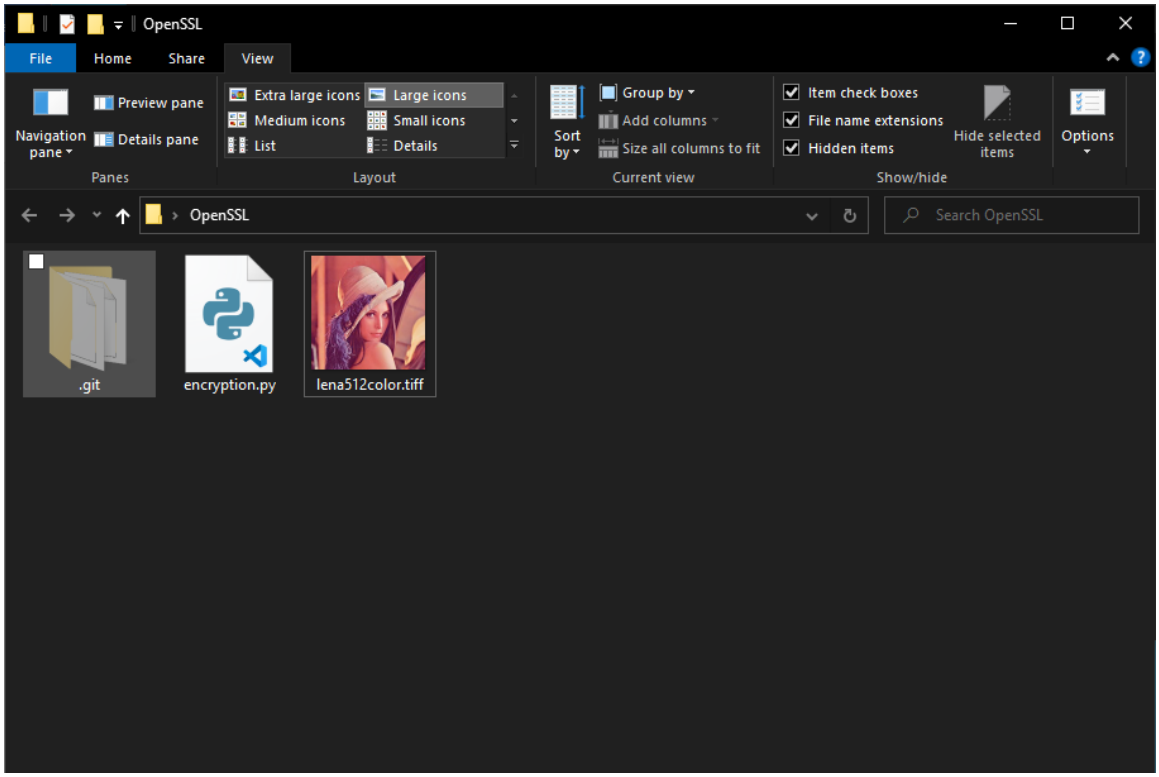
## Program Usage

The program can either be use as a module to be imported to other files or it can be used as a standalone program. Program usage will describe how the program will be used as a standalone python program.

The program provides four main encryption services, Symmetric Encryption using AES, Asymmetric Encryption using RSA, Hashing and Digital Signature using Elliptic Curve Digital Signature Algorithm but three additional optionalities were added to provide more controls for the user, these are Password File, RSA key pair and Elliptic Curve keypair generation.

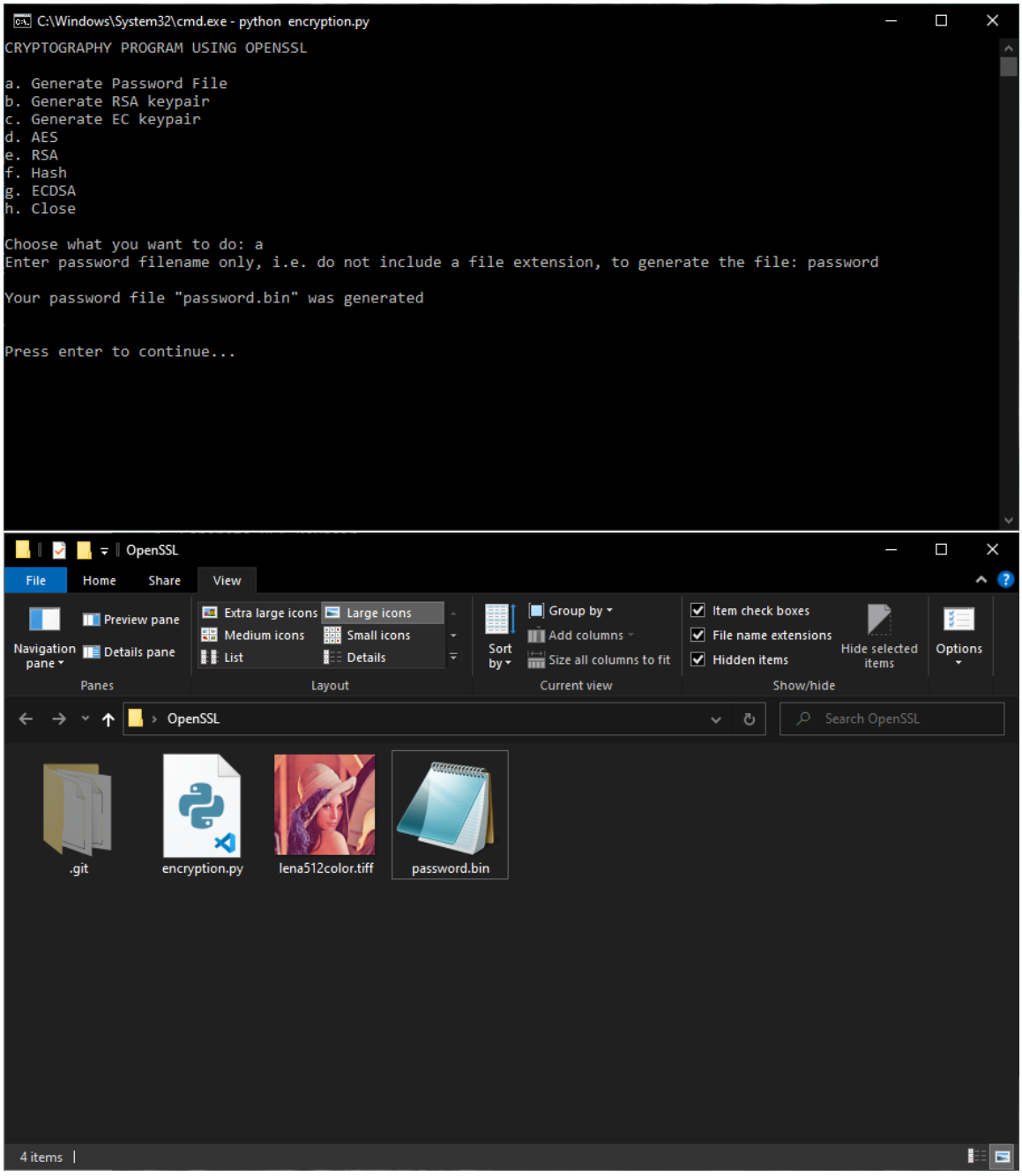
To lunch the program, type “python encryption.py” in the command prompt or any shell and upon execution, you will see the display of the program.



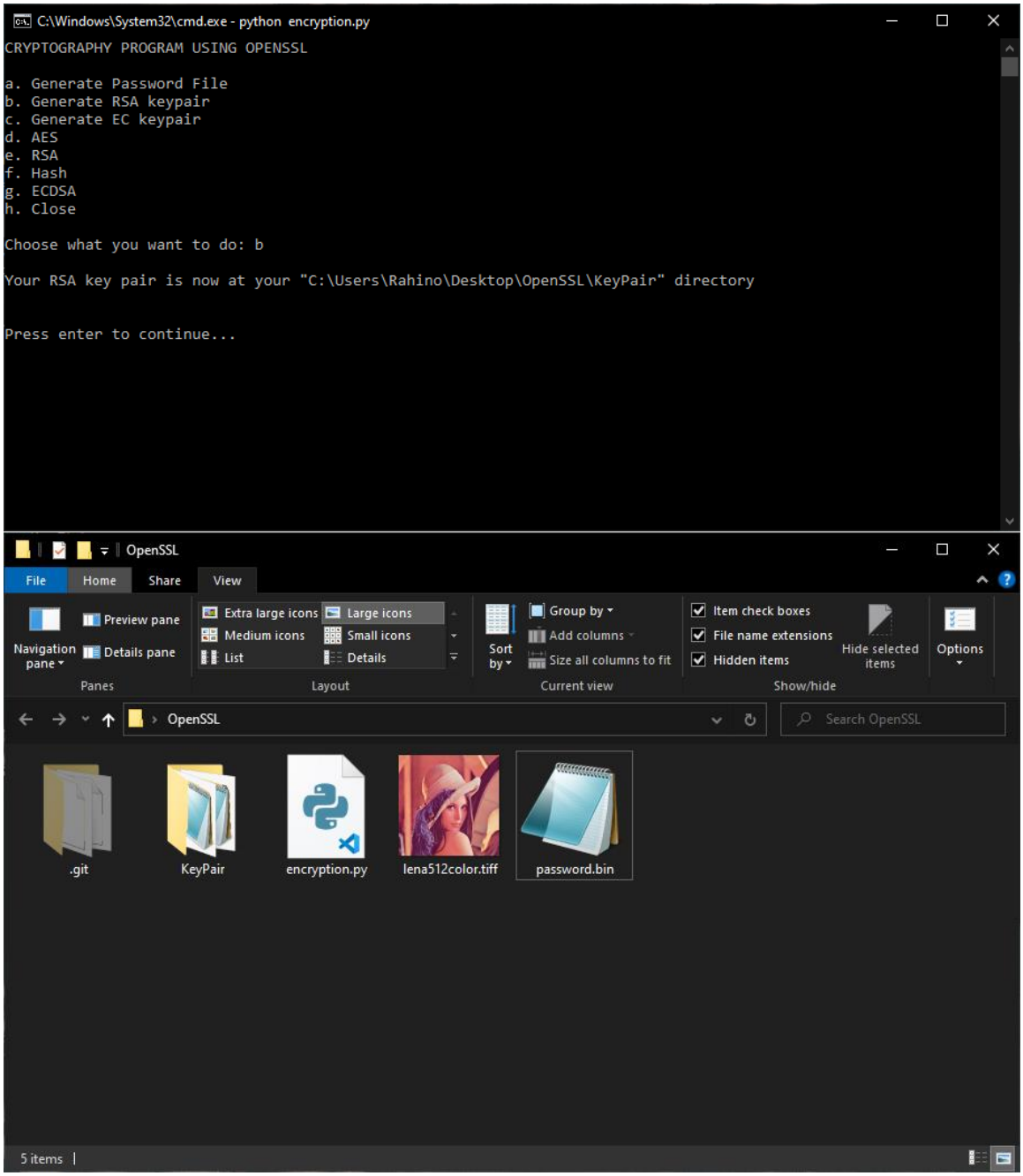
a. **Generate Password File** – option a. provides you a way to generate a password based on 64-bit random hexadecimal number.

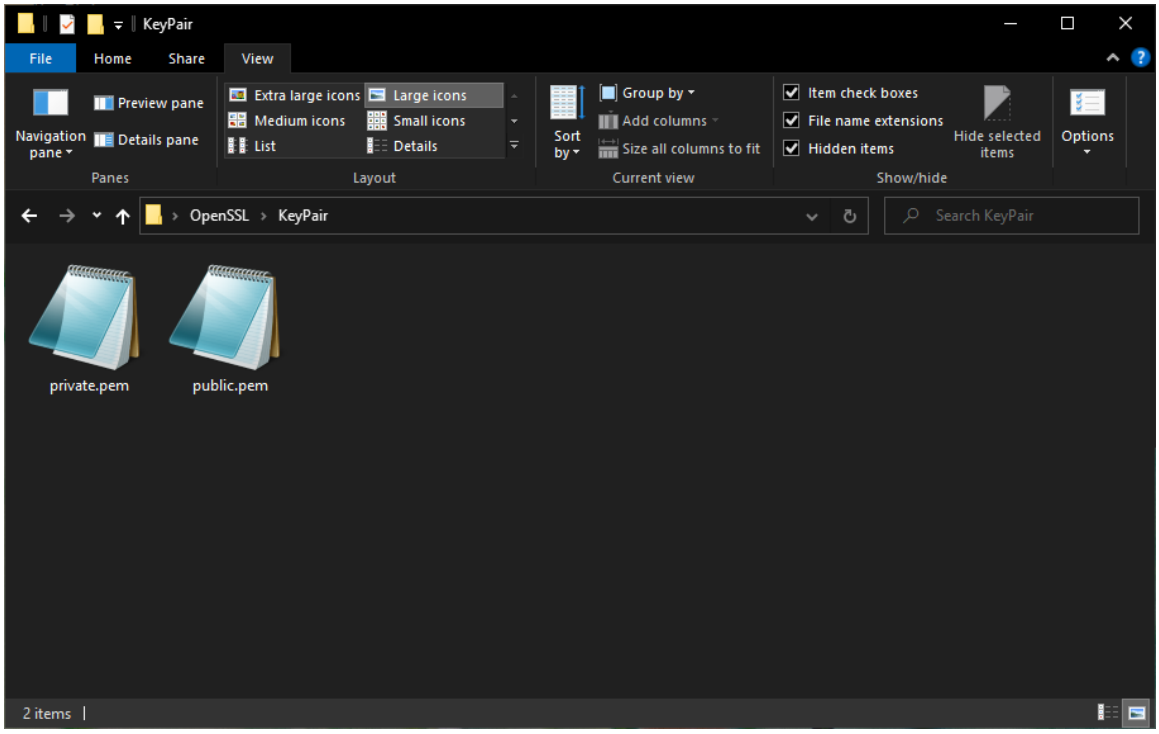
1. Press “A” key and hit Enter key
2. Type your desired filename for your password file and hit Enter
3. Check your Directory where the encryption.py file is located and you will see your file created with .bin file extension



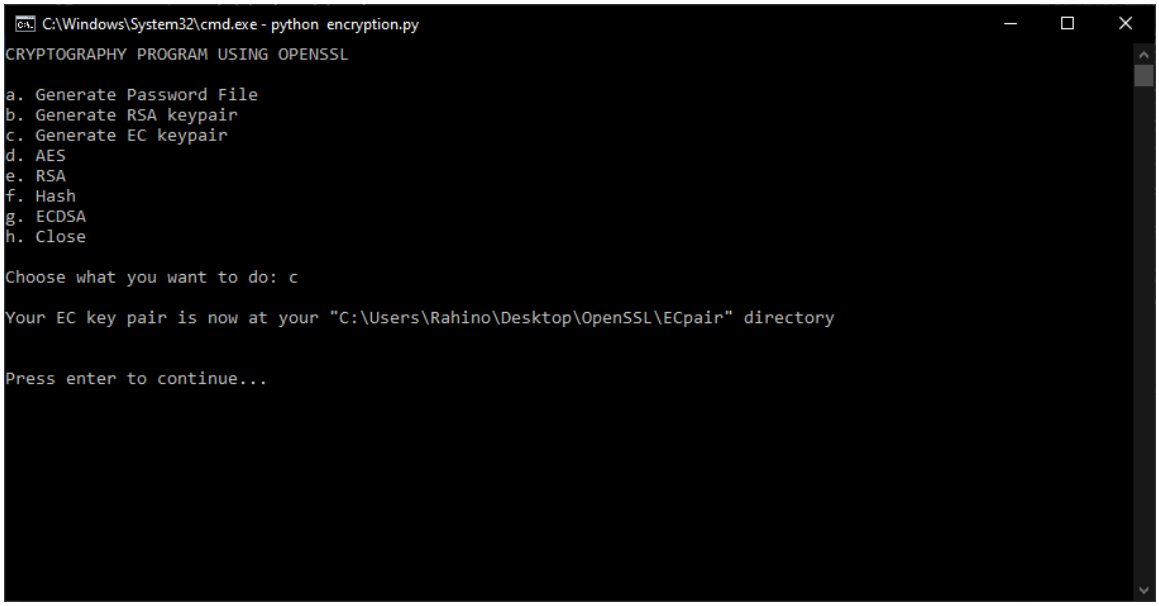
b. **Generate RSA keypair** – option b. provides you a way to Generate 2048-bit private key and its corresponding public key

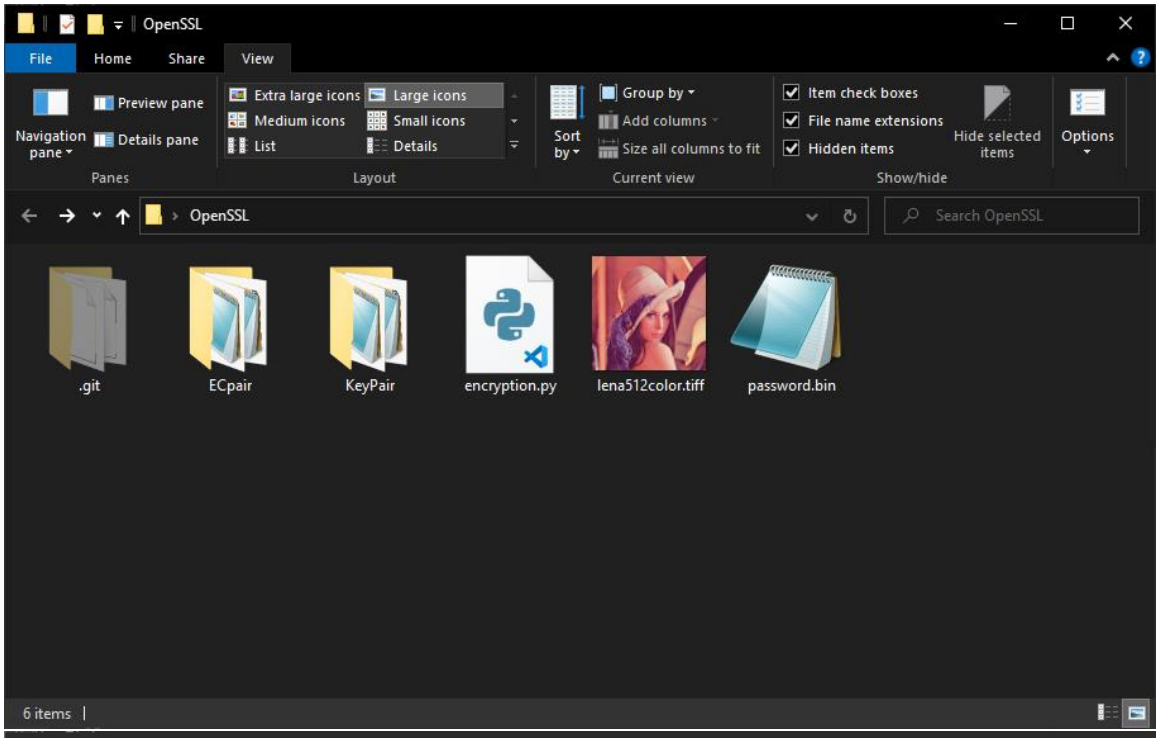
1. Press “B” key and hit Enter key
2. The RSA key pair will be automatically generated inside a folder name KeyPair

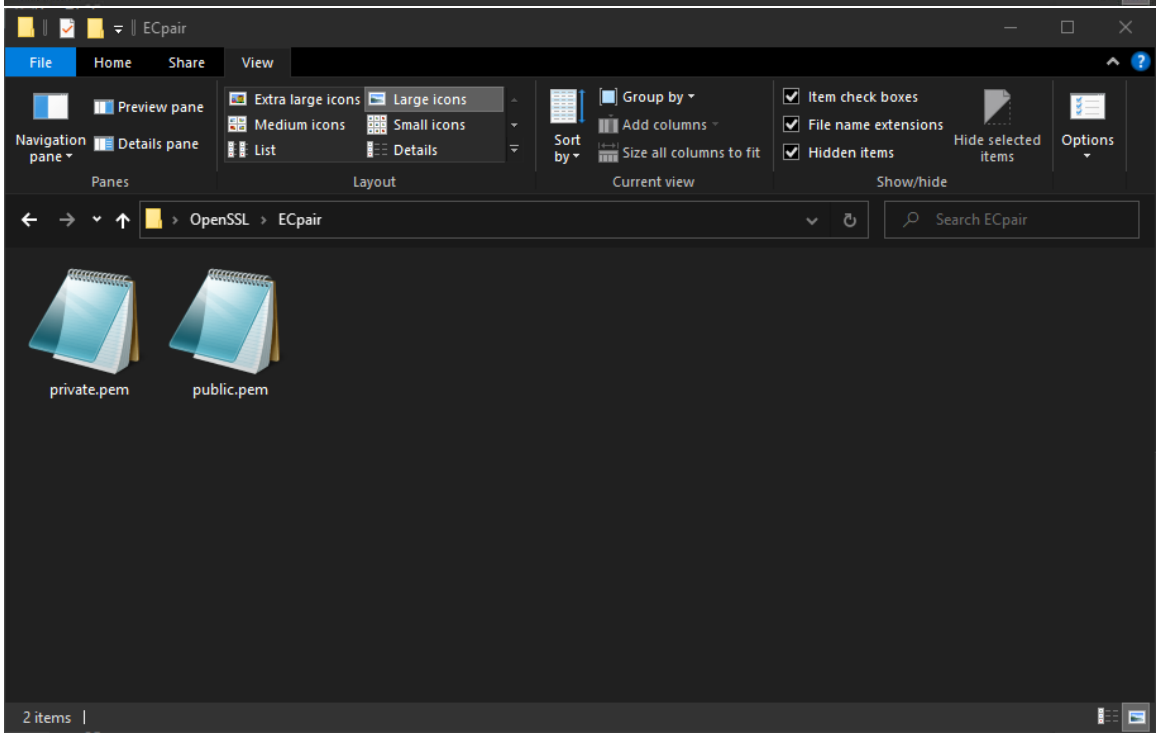





c. **Generate EC keypair** – option c. provides you a way to Generate elliptic curve key pair using secp384r1 EC curve
    1. Press “C” key and hit Enter key
    2. The EC key pair will be automatically generated inside a folder name ECpair

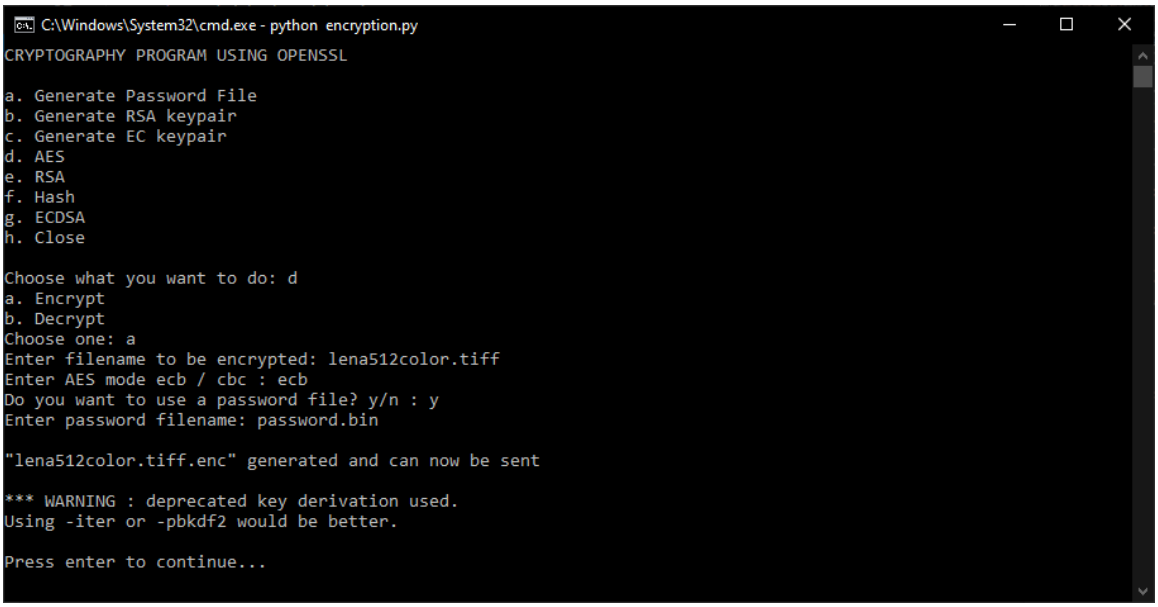


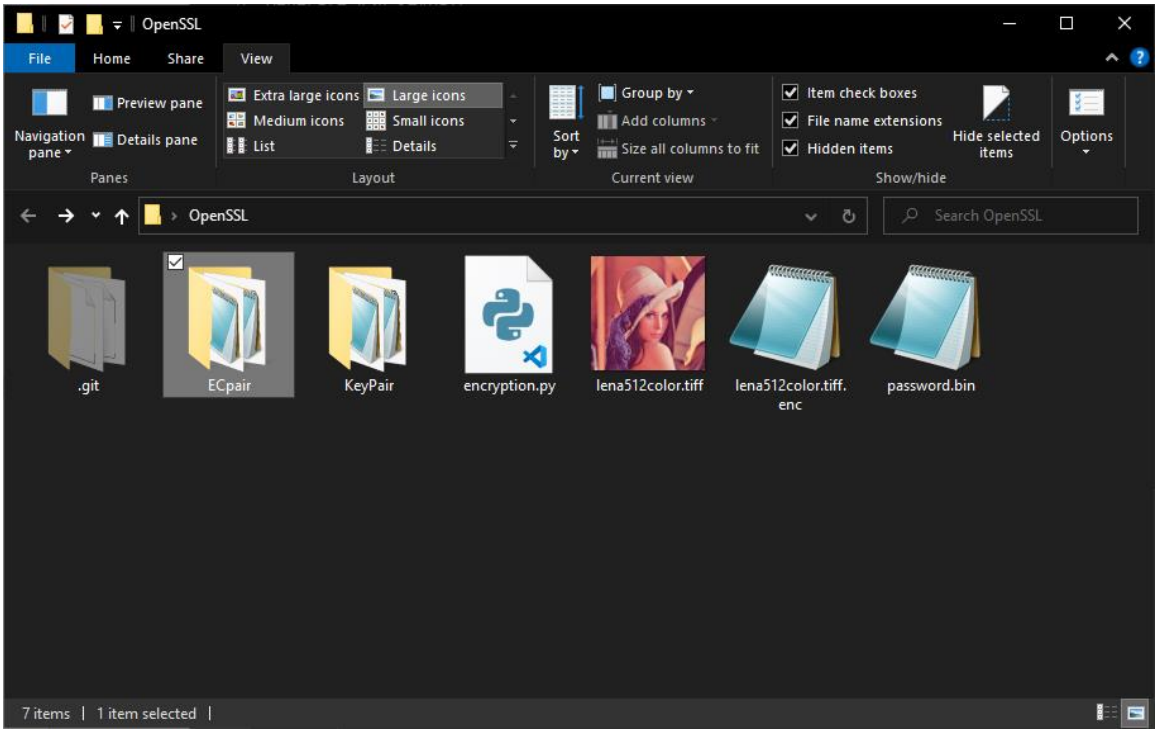

OpenSSL
File
Home
Share
View
Navigation pane
Preview pane
Details pane
Panes
Extra large icons
Large icons
Medium icons
Small icons
List
Details
Layout
Sort by
Group by
Add columns
Size all columns to fit
Current view
Item check boxes
File name extensions
Hidden items
Hide selected items
Show/hide
Options
OpenSSL
Search OpenSSL
.git
ECpair
KeyPair
encryption.py
lena512color.tiff
password.bin
6 items


ECpair
File
Home
Share
View
Navigation pane
Preview pane
Details pane
Panes
Extra large icons
Large icons
Medium icons
Small icons
List
Details
Layout
Sort by
Group by
Add columns
Size all columns to fit
Current view
Item check boxes
File name extensions
Hidden items
Hide selected items
Show/hide
Options
OpenSSL
ECpair
Search ECpair
private.pem
public.pem
2 items

d. **AES** – d. option provides you tools for symmetric encryption and decryption using Advance Encryption Standard defaulted to 128-bit block encryption
***ENCRYPTION***

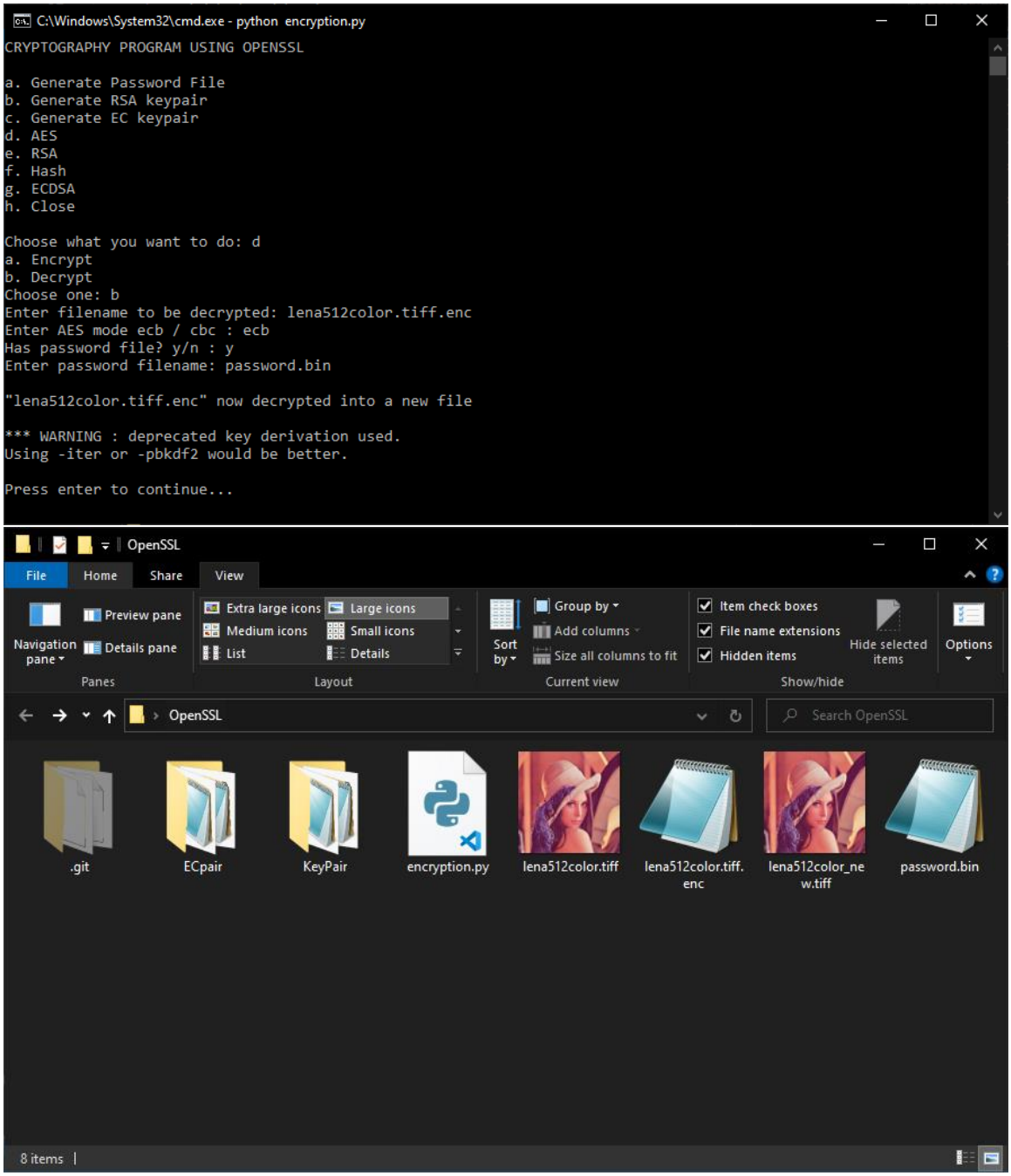
1. Press “D” key and hit Enter key
2. Press “A” key and hit Enter key to choose “Encrypt” option
3. Enter the filename you want to encrypt and include the file extension “lena512color.tiff”
4. Specify the mode of operation either “ecb” or “cbc”. The sample specifies “ecb”.
5. Specify whether you want to use file-based password or type-in password. The sample uses file-based password.
6. Specify the password file’s filename including its file extension. Sample uses “password.bin” the one generated earlier. If the user chooses not to use file-base password, this step will ask the type-in password from the user
7. Encrypted version of the file was generated with an extension of “.enc”






***DECRYPTION***

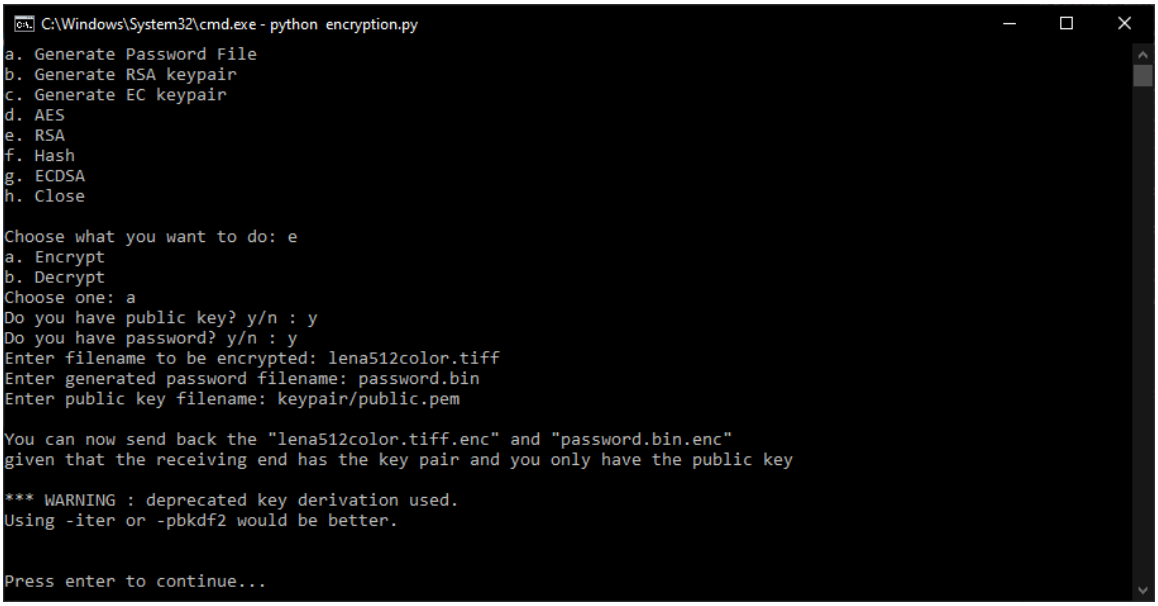
1. Press “D” key and hit Enter key
2. Press “B” key and hit Enter key to choose “Decrypt” option
3. Enter the filename you want to decrypt and include the file extension “lena512color.tiff.enc”
4. Specify the mode of operation either “ecb” or “cbc”. The sample specifies “ecb”. Take note, mode of decryption should match with mode of encryption.
5. Specify whether you want to use file-based password or type-in password. The sample uses file-based password.
6. Specify the password file’s filename including its file extension. Sample uses “password.bin” the one generated earlier. If the user chooses not to use file-base password, this step will ask the type-in password from the user
7. Decrypted version of the file was generated with “_new” appended on its filename

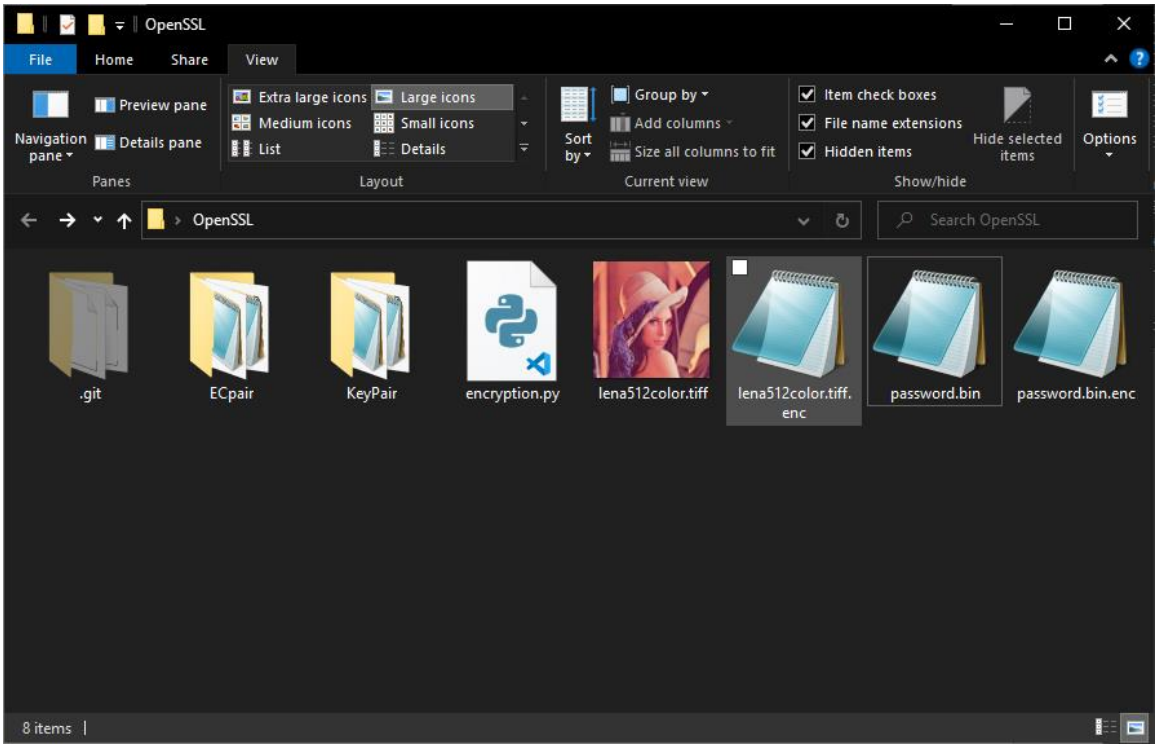

C:\Windows\System32\cmd.exe - python encryption.py
CRYPTOGRAPHY PROGRAM USING OPENSSL
a. Generate Password File
b. Generate RSA keypair
c. Generate EC keypair
d. AES
e. RSA
f. Hash
g. ECDSA
h. Close
Choose what you want to do: d
a. Encrypt
b. Decrypt
Choose one: b
Enter filename to be decrypted: lena512color.tiff.enc
Enter AES mode ecb / cbc : ecb
Has password file? y/n : y
Enter password filename: password.bin
"lena512color.tiff.enc" now decrypted into a new file
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Press enter to continue...
OpenSSL
File
Home
Share
View
Navigation pane
Preview pane
Details pane
Panes
Extra large icons
Large icons
Medium icons
Small icons
List
Details
Layout
Sort by
Group by
Add columns
Size all columns to fit
Current view
Item check boxes
File name extensions
Hidden items
Hide selected items
Options
Show/hide
OpenSSL
Search OpenSSL
.git
ECpair
KeyPair
encryption.py
lena512color.tiff
lena512color.tiff.enc
lena512color_new.tiff
password.bin
8 items

e. **RSA** – option e. provides you tools for public key encryption using both AES 128-bit CBC encryption and decryption for the main file and Rivest–Shamir–Adleman (RSA) encryption and decryption for the 64-bit password used as a key for the preceding AES encryption. AES was used for the main file because RSA can only encrypt files with sizes smaller than the RSA keypair bit, hence RSA public key was only used for the 64-bit key used in AES.

***ENCRYPTION***

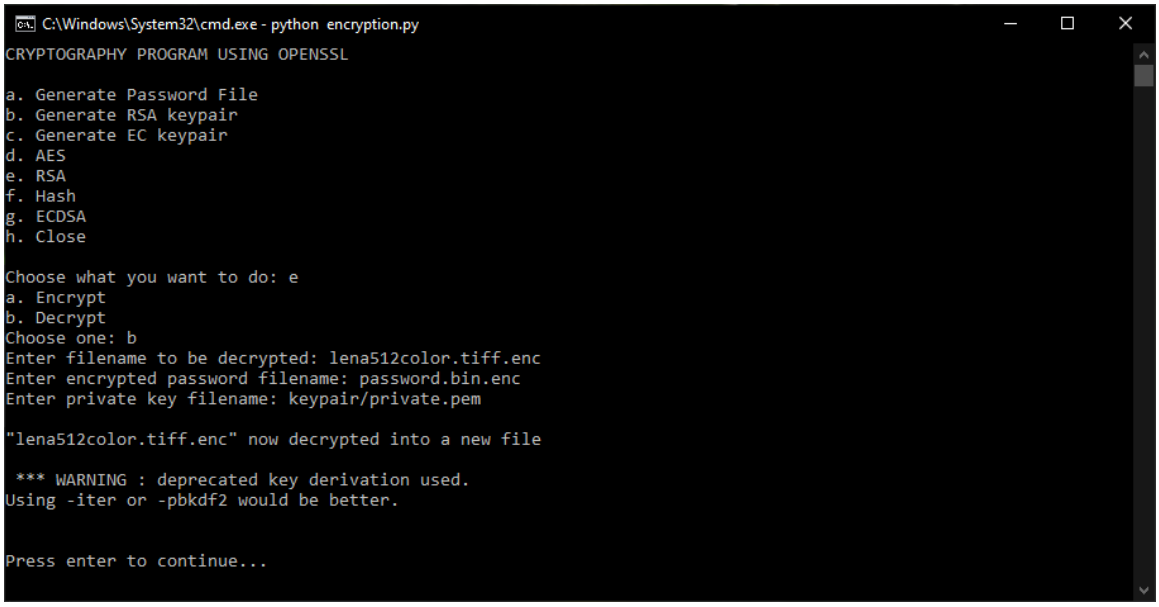
1. Press “E” key and hit Enter key
2. Press “A” key and hit Enter key to choose “Encrypt” option
3. Specify if you already have a public key. In this case sample already has the public key. In case you do not have, program will automatically generate RSA keypair.
4. Specify if you already have a password. In this case the password that the program generates is actually a 64-bit key for symmetric encryption. The sample already generates it, if you do not have, program will automatically generate the password file.
5. Specify the filename of the file to be encrypted including its file extension “lena512color.tiff”
6. Specify the password file’s filename including its file extension. Sample uses “password.bin” the one generated earlier
7. Specify the public key’s filename including its file extension. Sample uses “keypair/public.pem” inside the KeyPair folder.
8. Symmetrically encrypted target file and Asymmetrically encrypted AES key is now ready to be sent over given that the receiving end has the private key for the public key you used in AES key asymmetric encryption

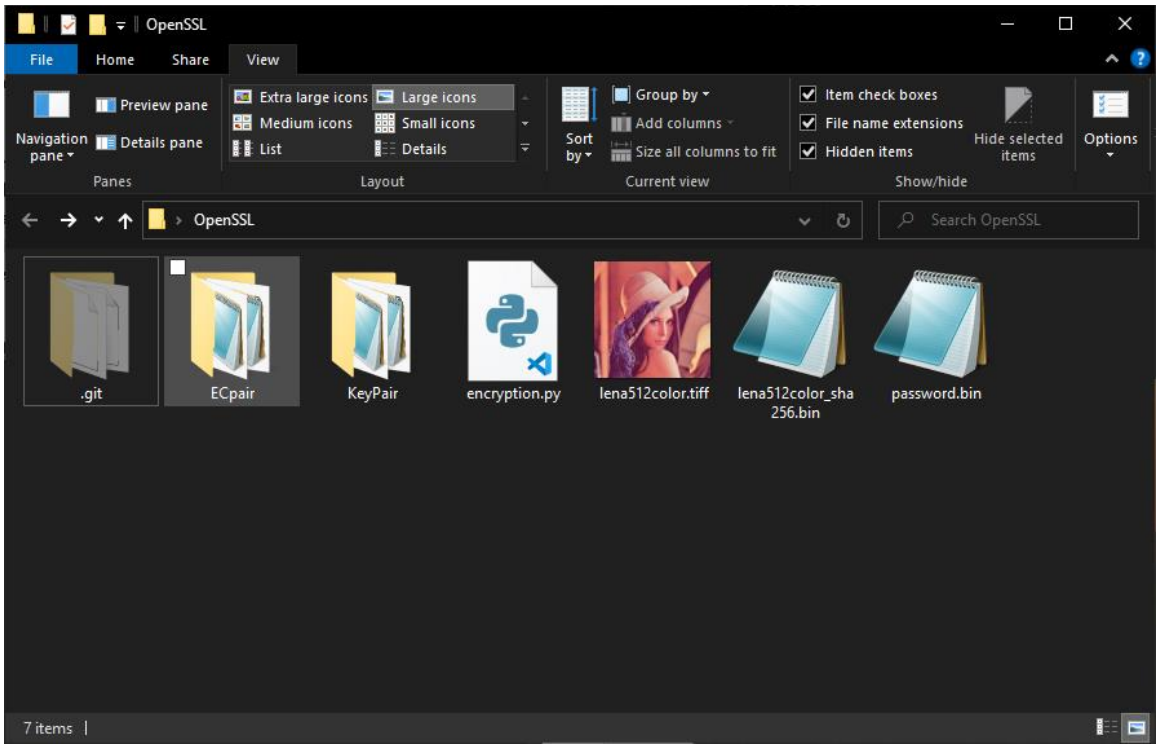





***DECRYPTION***

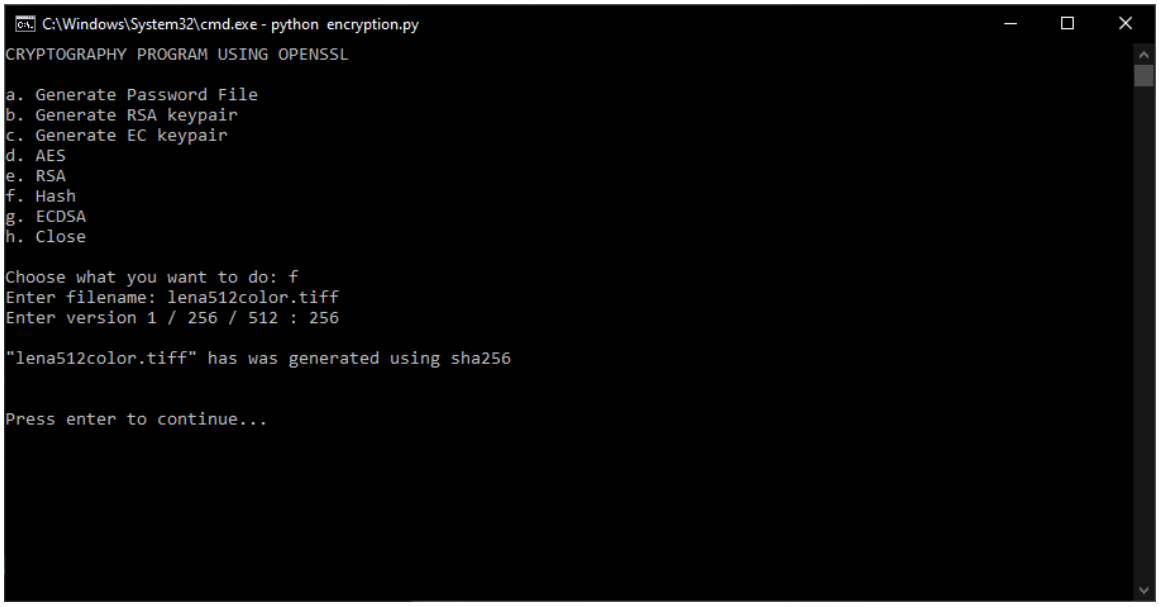
1. Press “E” key and hit Enter key
2. Press “B” key and hit Enter key to choose “Decrypt” option
3. Specify the filename of the file to be decrypted including its file extension “lena512color.tiff.enc”
4. Specify the encrypted password’s (AES key) filename including its file extension “password.bin.enc”
5. Specify the private key’s filename including its file extension. Sample uses “keypair/private.pem” inside the KeyPair folder.
6. Symmetrically encrypted target file and Asymmetrically encrypted AES key is now ready to be sent over given that the receiving end has the private key for the public key you used in AES key asymmetric encryption
7. Decrypted version of the file was generated with “_new” appended on its filename

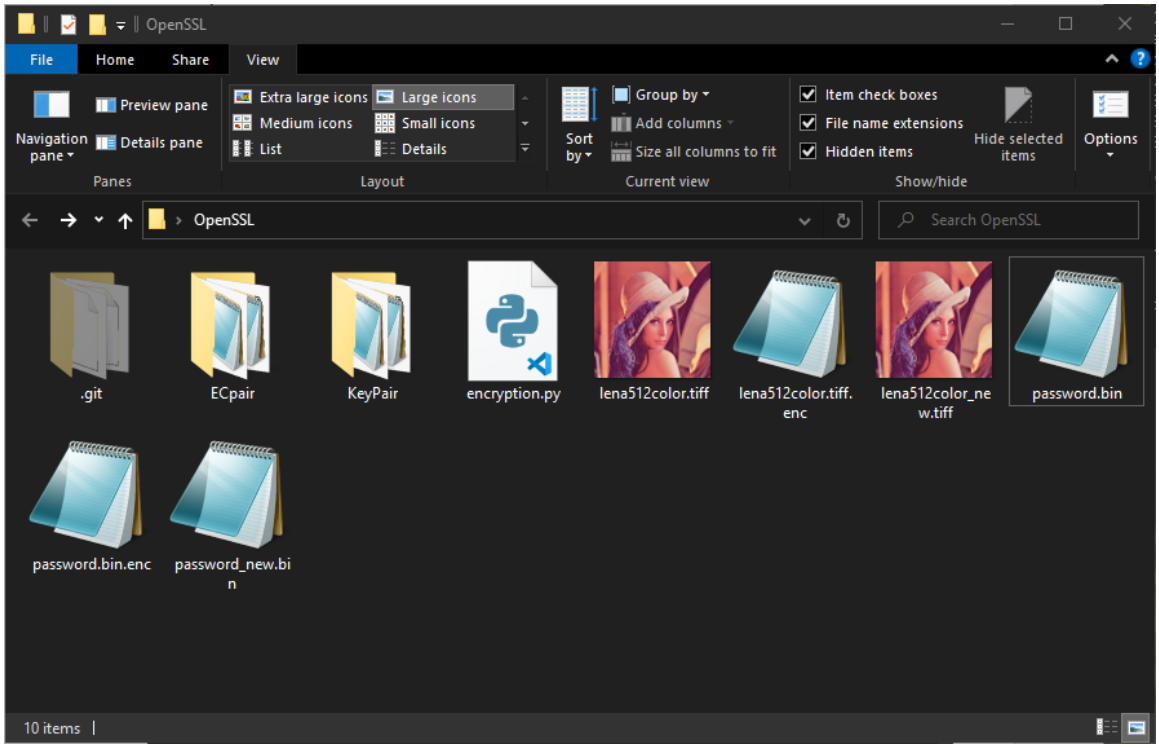

C:\Windows\System32\cmd.exe - python encryption.py
CRYPTOGRAPHY PROGRAM USING OPENSSL
a. Generate Password File
b. Generate RSA keypair
c. Generate EC keypair
d. AES
e. RSA
f. Hash
g. ECDSA
h. Close
Choose what you want to do: e
a. Encrypt
b. Decrypt
Choose one: b
Enter filename to be decrypted: lena512color.tiff.enc
Enter encrypted password filename: password.bin.enc
Enter private key filename: keypair/private.pem
"lena512color.tiff.enc" now decrypted into a new file
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Press enter to continue...


OpenSSL
File
Home
Share
View
Navigation pane
Preview pane
Details pane
Panes
Extra large icons
Large icons
Medium icons
Small icons
List
Details
Layout
Sort by
Group by
Add columns
Size all columns to fit
Current view
Item check boxes
File name extensions
Hidden items
Hide selected items
Show/hide
Options
OpenSSL
Search OpenSSL
.git
ECpair
KeyPair
encryption.py
lena512color.tiff
lena512color_sha256.bin
password.bin
7 items

f. **Hash** – option f. provides you tools for hashing a file using Secure Hash Algorithm

1. Press “F” key and hit Enter key
2. Enter the filename you want to hash including its file extension “lena512color.tiff”
3. Specify SHA version “1”, “256” or “512”. Sample uses “256”
4. Hash was generated with “_sha256” appended on its filename

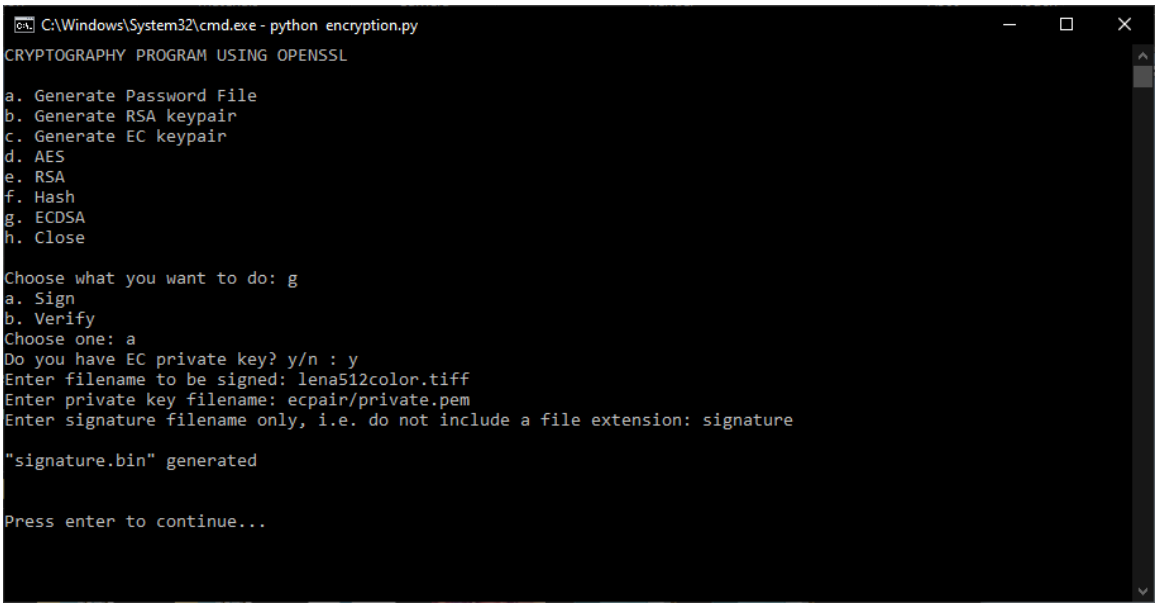


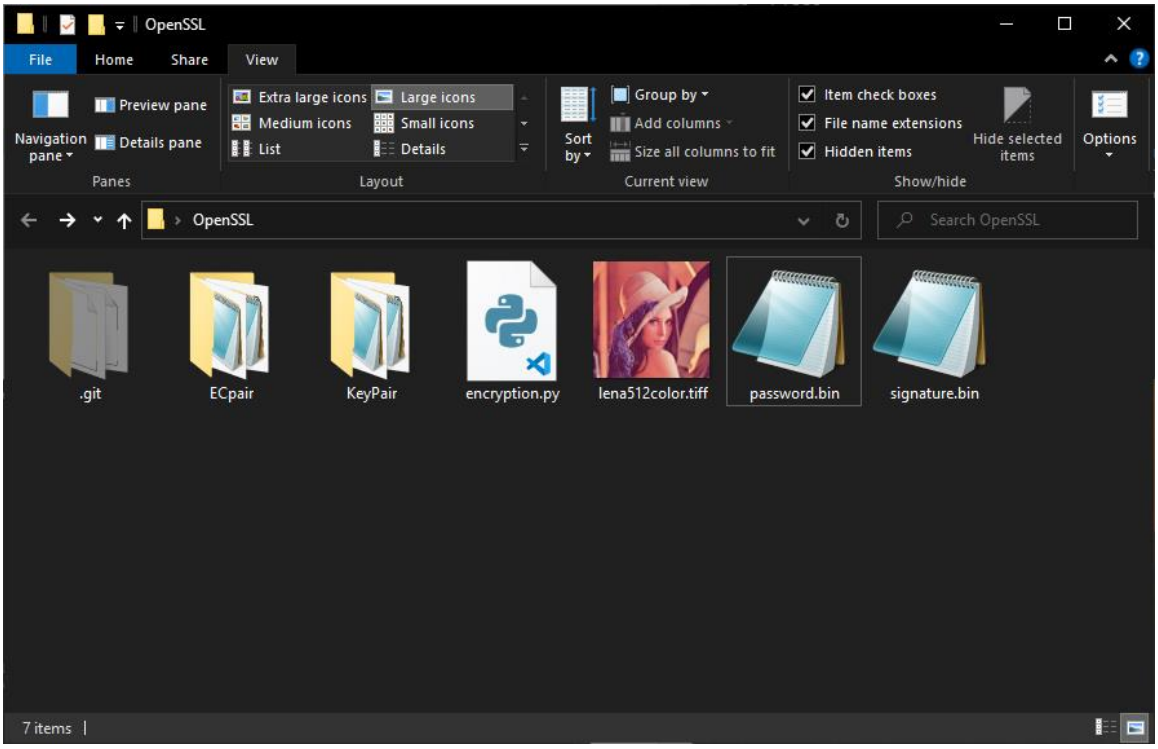



g. **ECDSA** – option g. provides you tools for signing and verifying files using Elliptic Curve Digital Signature Algorithm

***SIGNING***

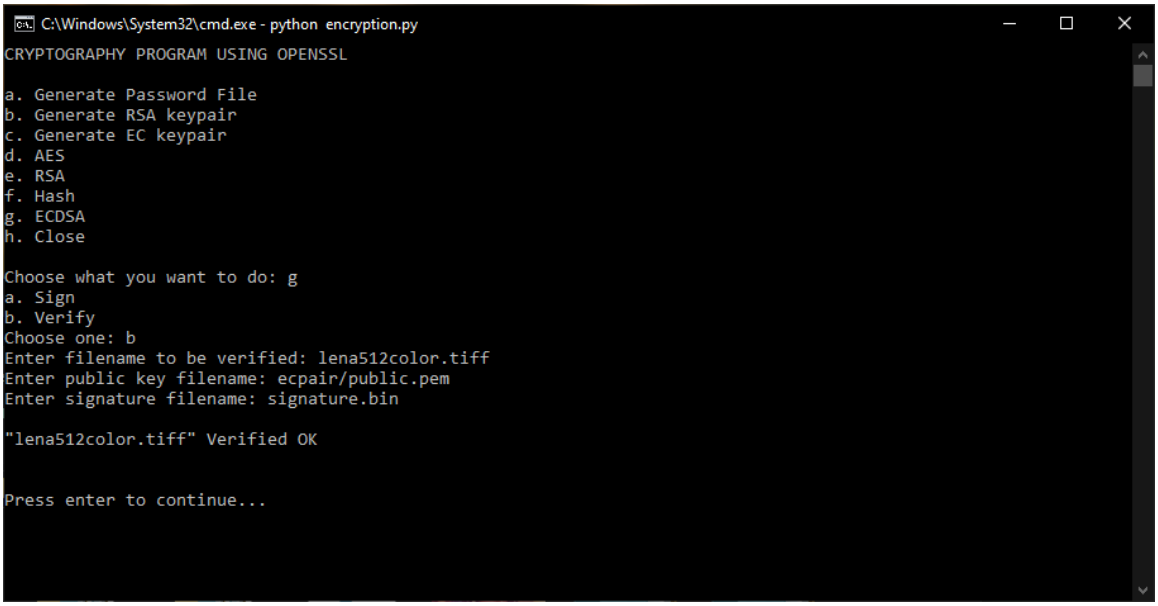
1. Press “G” key and hit Enter key
2. Press “A” key and hit Enter key to choose “Sign” option
3. Specify if you have EC private key. If not, the program will automatically generate EC keypair inside ECpair folder.
4. Specify the filename of the file to be signed including its file extension “lena512color.tiff”
5. Specify the EC private key’s filename including its file extension “ecpair/private.pem”
6. Specify the filename for your signature file. Sample used “signature”
7. “signature.bin” is now generated and can be used for file verification

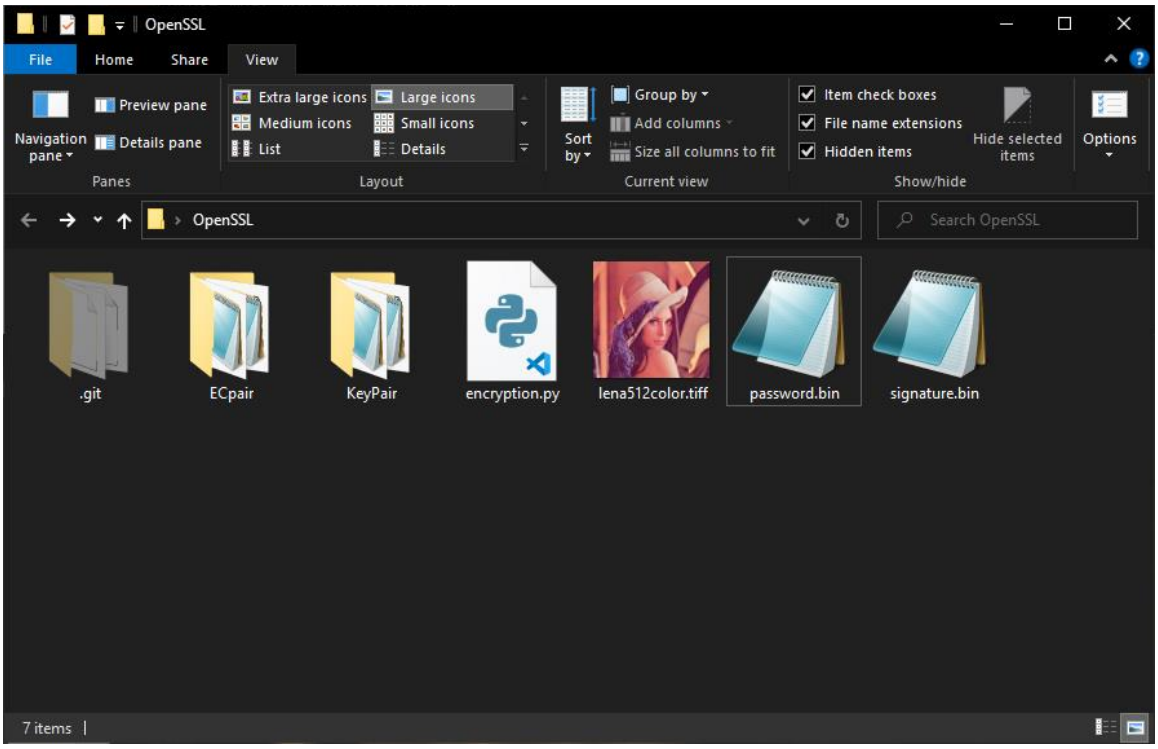





***VERIFICATION***

1. Press “G” key and hit Enter key
2. Press “B” key and hit Enter key to choose “Verify” option
3. Specify the filename of the file to be verified including its file extension “lena512color.tiff”
4. Specify the EC public key’s filename including its file extension “ecpair/public.pem”
5. Specify the filename for your signature file including its file extension “signature.bin”
6. Program prints “lena512color.tiff” Verified OK meaning that the file is authentic. Program will show an alternate print when the file is inauthentic.


C:\Windows\System32\cmd.exe - python encryption.py
CRYPTOGRAPHY PROGRAM USING OPENSSL
a. Generate Password File
b. Generate RSA keypair
c. Generate EC keypair
d. AES
e. RSA
f. Hash
g. ECDSA
h. Close
Choose what you want to do: g
a. Sign
b. Verify
Choose one: b
Enter filename to be verified: lena512color.tiff
Enter public key filename: ecpair/public.pem
Enter signature filename: signature.bin
"lena512color.tiff" Verified OK
Press enter to continue...


OpenSSL
File
Home
Share
View
Navigation pane
Preview pane
Details pane
Panes
Extra large icons
Large icons
Medium icons
Small icons
List
Details
Layout
Sort by
Group by
Add columns
Size all columns to fit
Current view
Item check boxes
File name extensions
Hidden items
Hide selected items
Show/hide
Options
OpenSSL
Search OpenSSL
.git
ECpair
KeyPair
encryption.py
lena512color.tiff
password.bin
signature.bin
7 items