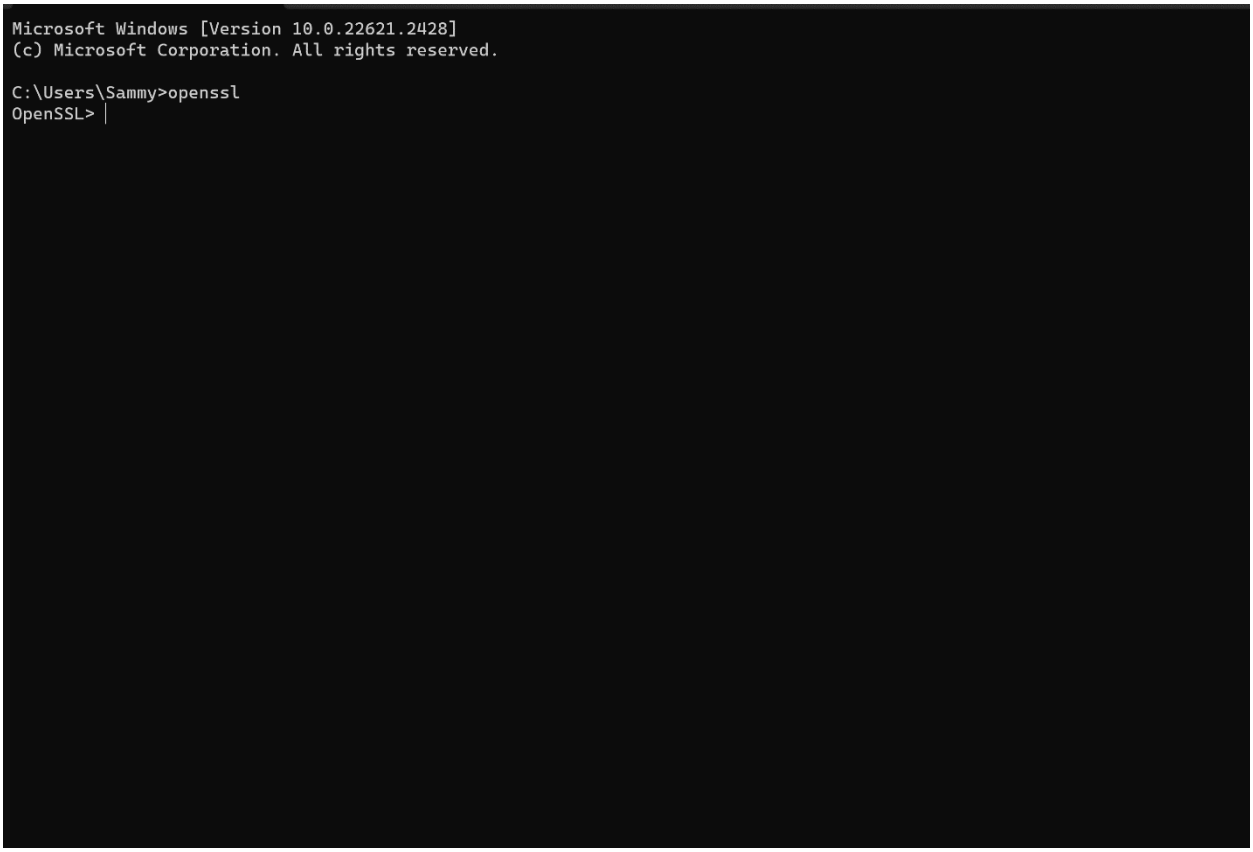
**Project 2 (OpenSsl)**

**Arnaldo Fajardo**

**Part I of the project.**

Watching the video gave me knowledge about OpenSsl. The video taught me about some important commands to be able to complete the project. The challenging part on the video was to be able to create private and public key because I did the project using windows OS, and the person on the video used Kali Linux. However, the video was a great deal of informative to get the project started.

**Start the OpenSSL command line:**



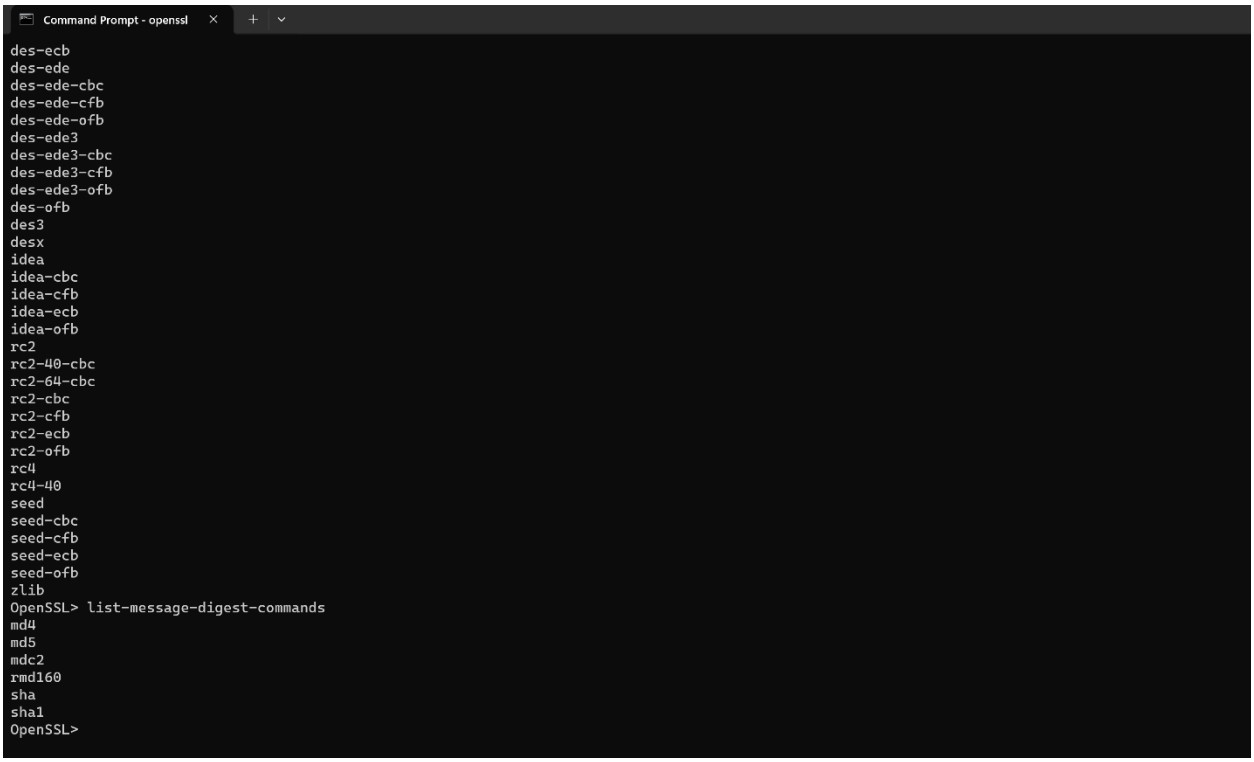
## List commands by type: List-standard-commands.

```
C:\Users\Sammy>openssl
OpenSSL> list-standard-commands
asn1parse
ca
ciphers
cms
crl
crl2pkcs7
dgst
dh
dhparam
dsa
dsaparam
ec
ecparam
enc
engine
errstr
gendh
gendsa
genpkey
genrsa
nseq
ocsp
passwd
pkcs12
pkcs7
pkcs8
pkey
pkeyparam
pkeyutl
prime
rand
req
rsa
rsautl
s_client
s_server
s_time
sess_id
```

**List commands by type: List-cipher-commands.**

```
Command Prompt - openssl

OpenSSL> list-cipher-commands
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
camellia-128-cbc
camellia-128-ecb
camellia-192-cbc
camellia-192-ecb
camellia-256-cbc
camellia-256-ecb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
cast5-ofb
des
des-cbc
des-cfb
des-ecb
des-ede
des-ede-cbc
des-ede-cfb
des-ede-ofb
des-ede3
des-ede3-cbc
des-ede3-cfb
des-ede3-ofb
des-ofb
des3
desx
```

**List commands by type: List-message-digest-commands. It is shown on the bottom part of the picture.**




.

**Here you can see the command ‘Help’ which is invalid, but provide us other commands.**

```
Administrator: Command Prompt - openssl

C:\Windows\System32>openssl
OpenSSl> help
openssl:Error: 'help' is an invalid command.

Standard commands
asn1parse         ca                ciphers           cms
crl               crl2pkcs7         dgst              dh
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
gendh             gendsa            genpkey           genrsa
nseq              ocsp              passwd            pkcs12
pkcs7             pkcs8             pkey              pkeyparam
pkeyutl           prime             rand              req
rsa               rsautl            s_client          s_server
s_time            sess_id           smime             speed
spkac             srp               ts                verify
version           x509

Message Digest commands (see the `dgst' command for more details)
md4               md5               mdc2              rmd160
sha               sha1

Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       base64            bf
bf-cbc            bf-cfb            bf-ecb            bf-ofb
camellia-128-cbc  camellia-128-ecb  camellia-192-cbc  camellia-192-ecb
camellia-256-cbc  camellia-256-ecb  cast              cast-cbc
cast5-cbc         cast5-cfb         cast5-ecb         cast5-ofb
des               des-cbc           des-cfb           des-ecb
des-ede           des-ede-cbc       des-ede-cfb       des-ede-ofb
des-ede3          des-ede3-cbc      des-ede3-cfb      des-ede3-ofb
des-ofb           des3              desx              idea
idea-cbc          idea-cfb          idea-ecb          idea-ofb
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            seed              seed-cbc          seed-cfb
seed-ecb          seed-ofb          zlib

OpenSSL>
```

**Performance of OpenSSL: the Speed command.**

```
OpenSSL> speed
Doing mdc2 for 3s on 16 size blocks: 1143919 mdc2's in 2.27s
Doing mdc2 for 3s on 64 size blocks: 344402 mdc2's in 2.33s
Doing mdc2 for 3s on 256 size blocks: 93192 mdc2's in 2.64s
Doing mdc2 for 3s on 1024 size blocks: 31349 mdc2's in 2.33s
Doing mdc2 for 3s on 8192 size blocks: 5411 mdc2's in 2.47s
Doing md4 for 3s on 16 size blocks: 2735845 md4's in 2.52s
Doing md4 for 3s on 64 size blocks: 2419826 md4's in 2.42s
Doing md4 for 3s on 256 size blocks: 1844141 md4's in 2.56s
Doing md4 for 3s on 1024 size blocks: 980657 md4's in 2.59s
Doing md4 for 3s on 8192 size blocks: 307360 md4's in 2.42s
Doing md5 for 3s on 16 size blocks: 2900884 md5's in 2.41s
Doing md5 for 3s on 64 size blocks: 2032430 md5's in 2.52s
Doing md5 for 3s on 256 size blocks: 1415615 md5's in 2.59s
Doing md5 for 3s on 1024 size blocks: 631031 md5's in 2.33s
Doing md5 for 3s on 8192 size blocks: 173488 md5's in 2.50s
Doing hmac(md5) for 3s on 16 size blocks: 5814191 hmac(md5)'s in 2.44s
Doing hmac(md5) for 3s on 64 size blocks: 3085760 hmac(md5)'s in 2.52s
Doing hmac(md5) for 3s on 256 size blocks: 1861141 hmac(md5)'s in 2.58s
Doing hmac(md5) for 3s on 1024 size blocks: 708144 hmac(md5)'s in 2.64s
Doing hmac(md5) for 3s on 8192 size blocks: 144422 hmac(md5)'s in 2.53s
Doing sha1 for 3s on 16 size blocks: 3374980 sha1's in 1.97s
Doing sha1 for 3s on 64 size blocks: 2330522 sha1's in 2.63s
Doing sha1 for 3s on 256 size blocks: 1632175 sha1's in 2.59s
Doing sha1 for 3s on 1024 size blocks: 816083 sha1's in 2.55s
Doing sha1 for 3s on 8192 size blocks: 153128 sha1's in 2.59s
Doing sha256 for 3s on 16 size blocks: 11291721 sha256's in 2.31s
Doing sha256 for 3s on 64 size blocks: 4604287 sha256's in 2.20s
Doing sha256 for 3s on 256 size blocks: 1978937 sha256's in 2.13s
Doing sha256 for 3s on 1024 size blocks: 575372 sha256's in 1.81s
Doing sha256 for 3s on 8192 size blocks: 69559 sha256's in 2.03s
Doing sha512 for 3s on 16 size blocks: 7315276 sha512's in 2.47s
Doing sha512 for 3s on 64 size blocks: 6544902 sha512's in 2.70s
Doing sha512 for 3s on 256 size blocks: 1933575 sha512's in 2.47s
Doing sha512 for 3s on 1024 size blocks: 737847 sha512's in 2.45s
Doing sha512 for 3s on 8192 size blocks: 110734 sha512's in 2.17s
Doing whirlpool for 3s on 16 size blocks: 3610900 whirlpool's in 2.22s
Doing whirlpool for 3s on 64 size blocks: 2457482 whirlpool's in 2.17s
Doing whirlpool for 3s on 256 size blocks: 780435 whirlpool's in 2.58s
Doing whirlpool for 3s on 1024 size blocks: 212716 whirlpool's in 2.61s
```

**The Speed rsa2048 command. It can be seen at the bottom.**

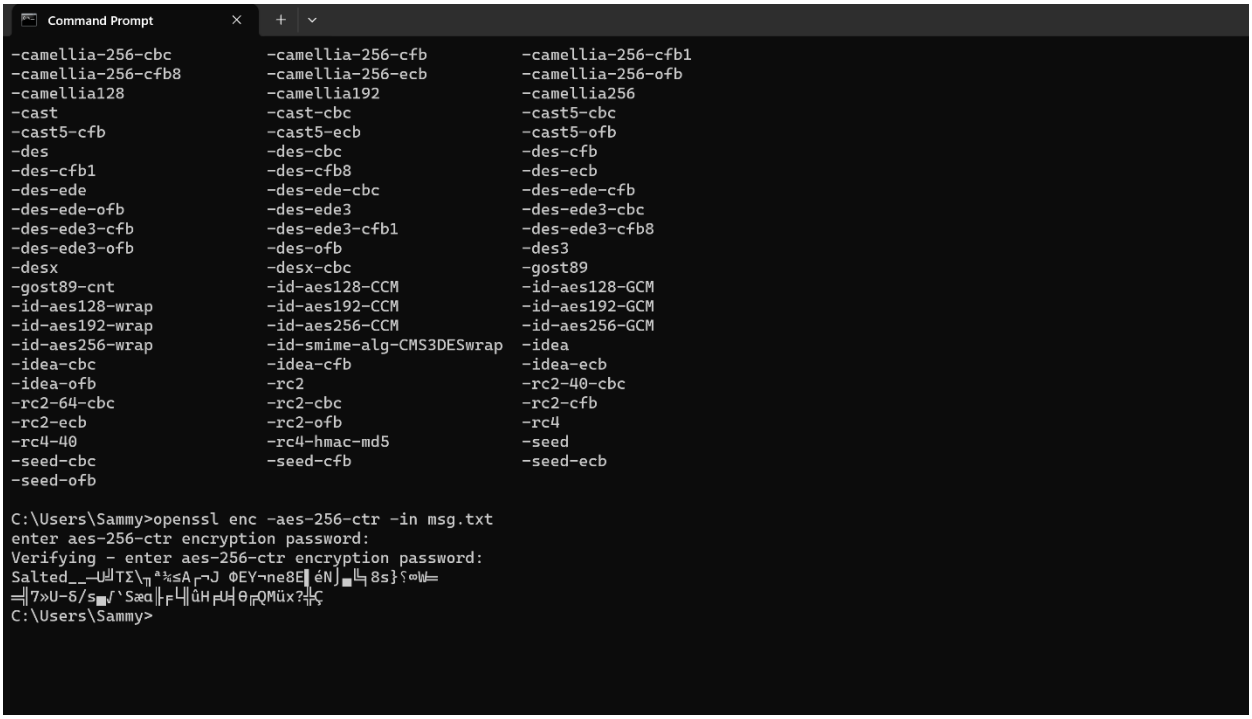
```
Command Prompt
Doing sha1 for 3s on 1024 size blocks: 816083 sha1's in 2.55s
Doing sha1 for 3s on 8192 size blocks: 153128 sha1's in 2.59s
Doing sha256 for 3s on 16 size blocks: 11291721 sha256's in 2.31s
Doing sha256 for 3s on 64 size blocks: 4604287 sha256's in 2.20s
Doing sha256 for 3s on 256 size blocks: 1978937 sha256's in 2.13s
Doing sha256 for 3s on 1024 size blocks: 575372 sha256's in 1.81s
Doing sha256 for 3s on 8192 size blocks: 69559 sha256's in 2.03s
Doing sha512 for 3s on 16 size blocks: 7315276 sha512's in 2.47s
Doing sha512 for 3s on 64 size blocks: 6544902 sha512's in 2.70s
Doing sha512 for 3s on 256 size blocks: 1933575 sha512's in 2.47s
Doing sha512 for 3s on 1024 size blocks: 737847 sha512's in 2.45s
Doing sha512 for 3s on 8192 size blocks: 110734 sha512's in 2.17s
Doing whirlpool for 3s on 16 size blocks: 3610900 whirlpool's in 2.22s
Doing whirlpool for 3s on 64 size blocks: 2457482 whirlpool's in 2.17s
Doing whirlpool for 3s on 256 size blocks: 780435 whirlpool's in 2.58s
Doing whirlpool for 3s on 1024 size blocks: 212716 whirlpool's in 2.61s
Doing whirlpool for 3s on 8192 size blocks: 27740 whirlpool's in 2.59s
Doing rmd160 for 3s on 16 size blocks: 2147640 rmd160's in 2.66s
Doing rmd160 for 3s on 64 size blocks: 2079940 rmd160's in 2.39s
Doing rmd160 for 3s on 256 size blocks: 948811 rmd160's in 2.14s
Doing rmd160 for 3s on 1024 size blocks: 287034 rmd160's in 2.58s
Doing rmd160 for 3s on 8192 size blocks: 41681 rmd160's in 2.33s
Doing rc4 for 3s on 16 size blocks: 53109780 rc4's in 2.53s
Doing rc4 for 3s on 64 size blocks: 24245654 rc4's in 2.47s
Doing rc4 for 3s on 256 size blocks: 4385058 rc4's in 2.34s
Doing rc4 for 3s on 1024 size blocks: ^C
C:\Users\Sammy>openssl speed rsa2048
Doing 2048 bit private rsa's for 10s: 12737 2048 bit private RSA's in 8.45s
Doing 2048 bit public rsa's for 10s: 305344 2048 bit public RSA's in 8.14s
OpenSSL 1.0.2j-fips  26 Sep 2016
built on: reproducible build, date unspecified
options:bn(64,64) rc4(16x,int) des(idx,cisc,2,long) aes(partial) idea(int) blowfish(idx)
compiler: /mingw/bin/gcc.exe -I. -I.. -I../include  -DZLIB -DOPENSSL_THREADS -D_MT -DDSO_WIN32 -DOPENSSL_SSL_CLIENT_ENGINE_AUTO=capi -DOPENSSL_CAPIENG_DIALOG -DWIN
VER=0x0501 -D_WIN32_WINNT=0x0501 -D_WIN32_IE=0x0501 -DL_ENDIAN -Wall -DWIN32_LEAN_AND_MEAN -DUNICODE -D_UNICODE -O2 -pipe -mms-bitfields -fno-builtin -march=core2
-mtune=core2 -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -I/mingw/include -DRC4_ASM -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM
 -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM -DGHASH_ASM -DECP_NISTZ256_ASM
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.000664s 0.000027s   1506.8  37508.7
```

**For these three encryptions, I created a text file(msg) and I called it from the right directory and I typed the name of the text file for the encryption.**

**Encryption using the -aes-128-cbc command**

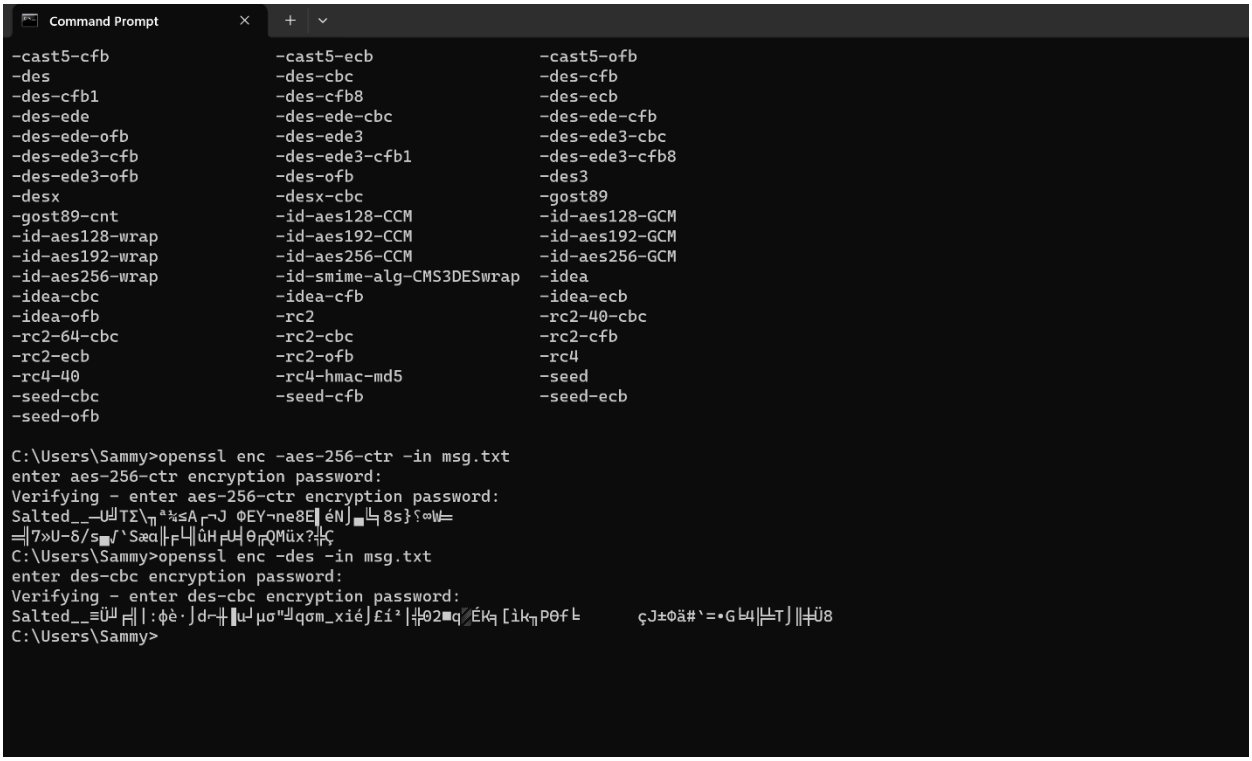
```
Command Prompt
Doing rc4 for 3s on 16 size blocks: 53109780 rc4's in 2.53s
Doing rc4 for 3s on 64 size blocks: 24245654 rc4's in 2.47s
Doing rc4 for 3s on 256 size blocks: 4385058 rc4's in 2.34s
Doing rc4 for 3s on 1024 size blocks: ^C
C:\Users\Sammy>openssl speed rsa2048
Doing 2048 bit private rsa's for 10s: 12737 2048 bit private RSA's in 8.45s
Doing 2048 bit public rsa's for 10s: 305344 2048 bit public RSA's in 8.14s
OpenSSL 1.0.2j-fips  26 Sep 2016
built on: reproducible build, date unspecified
options:bn(64,64) rc4(16x,int) des(idx,cisc,2,long) aes(partial) idea(int) blowfish(idx)
compiler: /mingw/bin/gcc.exe -I. -I.. -I../include  -DZLIB -DOPENSSL_THREADS -D_MT -DDSO_WIN32 -DOPENSSL_SSL_CLIENT_ENGINE_AUTO=capi -DOPENSSL_CAPIENG_DIALOG -DWI
VER=0x0501 -D_WIN32_WINNT=0x0501 -D_WIN32_IE=0x0501 -DL_ENDIAN -Wall -DWIN32_LEAN_AND_MEAN -DUNICODE -D_UNICODE -O2 -pipe -mms-bitfields -fno-builtin -march=core2
-mtune=core2 -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -I/mingw/include -DRC4_ASM -DSHA1_ASM -DSHA256_ASM -DSHA512_AS
 -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM -DGHASH_ASM -DECP_NISTZ256_ASM
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.000664s 0.000027s   1506.8  37508.7

C:\Users\Sammy>openssl enc -aes-128-cbc -in msg
msg: No such file or directory
7864:error:02001002:system library:fopen:No such file or directory:bss_file.c:402:fopen('msg','rb')
7864:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:404:

C:\Users\Sammy>openssl enc -aes-128-cbc -in msg.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
```

## Encryption with -aes-256-ctr with the text file msg.



```
Command Prompt

-camellia-256-cbc          -camellia-256-cfb          -camellia-256-cfb1
-camellia-256-cfb8         -camellia-256-ecb          -camellia-256-ofb
-camellia128               -camellia192               -camellia256
-cast                      -cast-cbc                  -cast5-cbc
-cast5-cfb                 -cast5-ecb                 -cast5-ofb
-des                       -des-cbc                   -des-cfb
-des-cfb1                  -des-cfb8                  -des-ecb
-des-ede                   -des-ede-cbc               -des-ede-cfb
-des-ede-ofb               -des-ede3                  -des-ede3-cbc
-des-ede3-cfb              -des-ede3-cfb1             -des-ede3-cfb8
-des-ede3-ofb              -des-ofb                   -des3
-desx                      -desx-cbc                  -gost89
-gost89-cnt                -id-aes128-CCM             -id-aes128-GCM
-id-aes128-wrap            -id-aes192-CCM             -id-aes192-GCM
-id-aes192-wrap            -id-aes256-CCM             -id-aes256-GCM
-id-aes256-wrap            -id-smime-alg-CMS3DESwrap  -idea
-idea-cbc                  -idea-cfb                  -idea-ecb
-idea-ofb                  -rc2                       -rc2-40-cbc
-rc2-64-cbc                -rc2-cbc                   -rc2-cfb
-rc2-ecb                   -rc2-ofb                   -rc4
-rc4-40                    -rc4-hmac-md5              -seed
-seed-cbc                  -seed-cfb                  -seed-ecb
-seed-ofb

C:\Users\Sammy>openssl enc -aes-256-ctr -in msg.txt
enter aes-256-ctr encryption password:
Verifying - enter aes-256-ctr encryption password:
Salted__╩TΣ\╥ª¾≤A┌¬J ΦEY¬ne8E▌éN⌡▄╚╕8s}ˢ∞W╞═
═╢7»U-δ/s▄ᶠ`Sæα╟┌F╝╢ûH┌U╢θ┌ᴿQMüx?╝ᶜC
C:\Users\Sammy>
```

**DES encryption. The command line is at the bottom part of the prompt.**



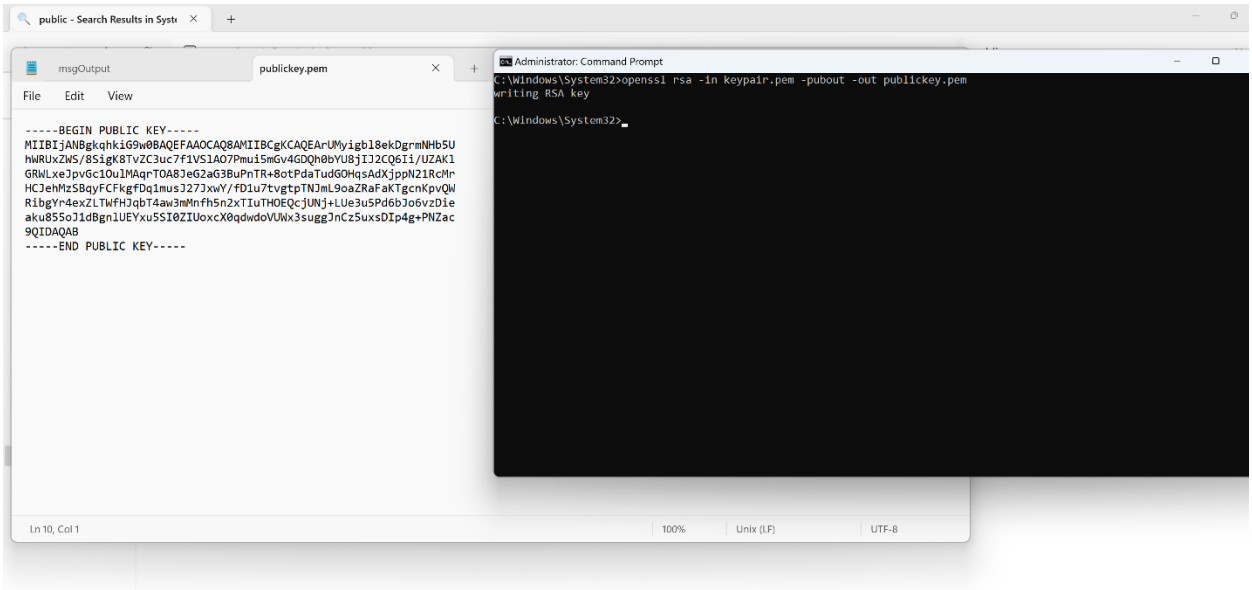
```
Command Prompt

-cast5-cfb                  -cast5-ecb                  -cast5-ofb
-des                        -des-cbc                    -des-cfb
-des-cfb1                   -des-cfb8                   -des-ecb
-des-ede                    -des-ede-cbc                -des-ede-cfb
-des-ede-ofb                -des-ede3                   -des-ede3-cbc
-des-ede3-cfb               -des-ede3-cfb1              -des-ede3-cfb8
-des-ede3-ofb               -des-ofb                    -des3
-desx                       -desx-cbc                   -gost89
-gost89-cnt                 -id-aes128-CCM              -id-aes128-GCM
-id-aes128-wrap             -id-aes192-CCM              -id-aes192-GCM
-id-aes192-wrap             -id-aes256-CCM              -id-aes256-GCM
-id-aes256-wrap             -id-smime-alg-CMS3DESwrap   -idea
-idea-cbc                   -idea-cfb                   -idea-ecb
-idea-ofb                   -rc2                        -rc2-40-cbc
-rc2-64-cbc                 -rc2-cbc                    -rc2-cfb
-rc2-ecb                    -rc2-ofb                    -rc4
-rc4-40                     -rc4-hmac-md5               -seed
-seed-cbc                   -seed-cfb                   -seed-ecb
-seed-ofb

C:\Users\Sammy>openssl enc -aes-256-ctr -in msg.txt
enter aes-256-ctr encryption password:
Verifying - enter aes-256-ctr encryption password:
Salted__╩TΣ\╥ª¾≤A┌¬J ΦEY¬ne8E▌éN⌡▄╙8s}⁾∞W╞
═╢7»U-δ/s▄∫`Sæα╟┌╙╢ùH╓U╝θ╓QMüx?╬Ç
C:\Users\Sammy>openssl enc -des -in msg.txt
enter des-cbc encryption password:
Verifying - enter des-cbc encryption password:
Salted__≡ÜJ╒╢│:φè·⌡d-╪▐u┘μσ"╝qσm_xié⌡£í²│╬02■q░ÉK╕[ìk╥PΘf╘      çJ±Φä#`=•G╘╢╩T⌡│╬Ü8
C:\Users\Sammy>
```

**This is the part of the creation for the private and public key. The private was created successfully when the message “BEGIN RSA PRIVATE KEY” showed after I typed -check command.**

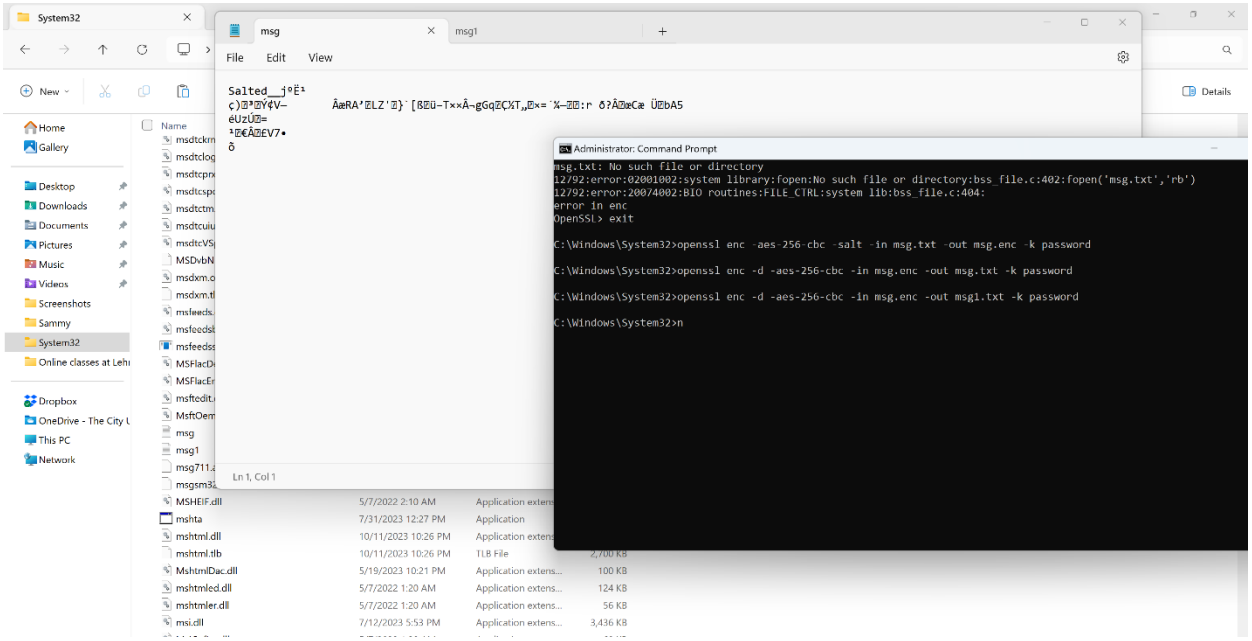
Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>openssl genrsa -out keypair.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................................+++
.....................+++
e is 65537 (0x10001)

C:\Windows\System32>openssl rsa -in keypair.pem -check
RSA key ok
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEArUMyigbl8ekDgrmNHb5UhWRUxZWS/8SigK8TvZC3uc7f1VSl
AO7Pmui5mGv4GDQh0bYU8jIJ2CQ6Ii/UZAKlGRWLxeJpvGc1OulMAqrTOA8JeG2a
G3BuPnTR+8otPdaTudGOHqsAdXjppN21RcMrHCJehMzSBqyFCFkgfDq1musJ27Jx
wY/fD1u7tvgtpTNJmL9oaZRaFaKTgcnKpvQWRibgYr4exZLTWfHJqbT4aw3mMnfh
5n2xTIuTHOEQcjUNj+lUe3u5Pd6bJo6vzDieaku855oJ1dBgnlUEYxu5SI0ZIUox
cX0qdwdoVUWx3suggJnCz5uxsDIp4g+PNZac9QIDAQABAoIBAE3MFX5amJdlzFi/
XjjTxHnc0h+iCM5liZ4EfvWobIa2+FR+3FQt5vQkuWlSWaTsi57xhjg19auioKPH
HDQH5//6Z7dj8ZxSL1mTdbjg4VSTk+oKNJ9LfYT3yW2om2Xf2pceHBVzLRKSRz7q
+kEn4EyStiQlH801W8qi0m5VokO7tnrOolFDjFrD+8HcktSKFZPnj8W+9M7wVWit
L1qjAnIk5OgLV65cmW/s3J6u7x322W9k9kJTzlyWm/Bi+7tjQhXlZapA3oSz21/9
t0f/fosB9c0pJzJljAnGd+2N0Hf8yNckytwy4ktthnq0f8nRfUNHWaOH7j+/f4Qw
1NHEOWUCgYEA2w/O4O6mUyQ/kXz/+s1BNf8vfxSftUmYj+cs4Vamlkh29lNAEJhx
CFM7TKm3r5ES3SPhHw1/LMnHqj+91VbaOXoBnPKkktry3bhoq8P0U9RQ6IV1JJEN
DOs+7SoMWdHOc6GjIxb+/UnuBtg0YjBWwC/8wpg6poLZDphJIMMus9MCgYEAynph
49XYXk3hkb45nN86UyiQLxcl6bBxbj4E4Pa2Ucclpx5Sp+8IVxDsZysORsO1z15X
qHNrGUTKU0Rx3VTv1Er8aBtaR4bJHYR6s/S4kGdUKF4DI3BpHALkGyMxF4nEr3gI
9ct+Kdmp8yae9fgzIinTWyLBnmAYK9iVPpGBlxcCgYEA1sDQl0JeW/OVPhPE/yJp
Y34gGf0l9JuJrMl8/gvslDfGKAXFMH/Bj9/IqodXRA58575c3AxnOIk37HdAqvzS
QraRDkPxguaSYTp/oqQfym7CQrFS/d4VZSzaK7alFQjR1BoJzESnBbiMWLlr4TZy
VUa2DC8gkEyGbgBs8GloR3ECgYEAsFDj309/S5eMHqrEZptoobOH4snIhPfVvscY
3oFYAFxnqGBaX8xhudeN1MUEeJDCSGKB+LkRAP8LJHlMogztjzMBGvL+qZxwk1ii
tQQcl/qyqTg8iWbA0NifpBYAENwyPw5K1yctpOna+VafQ9evR0eeuhTI/830HnJK
fCKe818CgYEAheKOUSE5Vyxamh7ET2ZtImnqNnDKR3+M4rxZhOcAeg0Fvatp3uai
H6IJVdMAhsZN4ctOW53nXZrmAk49tnyFKpkghhvmv6kT7D3B2w1gviPtOdUVQ1JR
W1MRqyVIgV64w0vwUxYqxt5acPXwrjI0ecOtt+3OFR4vbh3zqrJLqqc=
-----END RSA PRIVATE KEY-----

C:\Windows\System32>
```

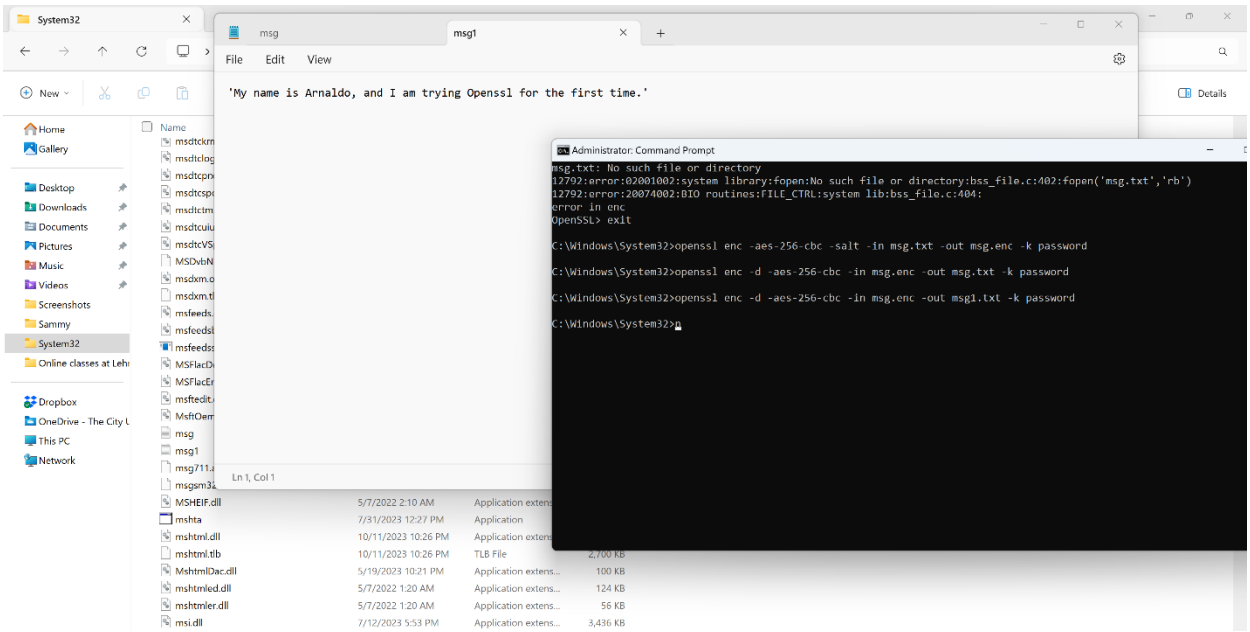
**Here is the public key where we need to use the private key name “keypair.pem”, and the command -pubout to be able to extract it from the private key. Also, I named it publickey.pem to have it saved in a text file. The message is show “BEGIN PUBLIC KEY”.**



**Here is the last part where I used the -aes-256-cbc algorithm to encrypt a message that I created. On the picture, the message is shown encrypted, and the commands can be seen as well.**



**Lastly, the decryption of the message is shown along with commands where I created a new text file "msg1" to show the decryption.**




I would say that this project had some challenges for me because since I never used OpenSSL before, and the video explained a great deal of the project but in another OS. As of result, I had to my research for the right command, and I found information about it for windows. One of the helpful websites was the OpenSSL website where sheet for commands were available.