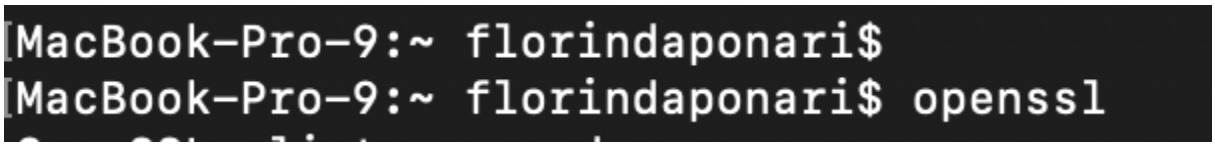
Florinda Ponari

For project 2, I delved into OpenSSL for encrypting and decrypting messages. What is OpenSSL? OpenSSL is “an open-source command line tool commonly used to generate CSRs and private keys, install the SSL files on your server, merge files, convert your certificate into various SSL formats, verify certificate information, and troubleshoot any potential issues”. OpenSSL has SSL/TLS protocols, symmetric/asymmetric encryption, digital signatures, and cryptographic standards like DES, AES, RC4, and RSA. In today's digital world, encryption is crucial for protecting data and protecting said data against cyber threats. It has been proven that OpenSSL is great for encryption and decryption. Initially, I lacked an understanding of what OpenSSL was; however, after exploring it firsthand, I now recognize its importance in cryptography and security, providing essential functionalities for data protection.

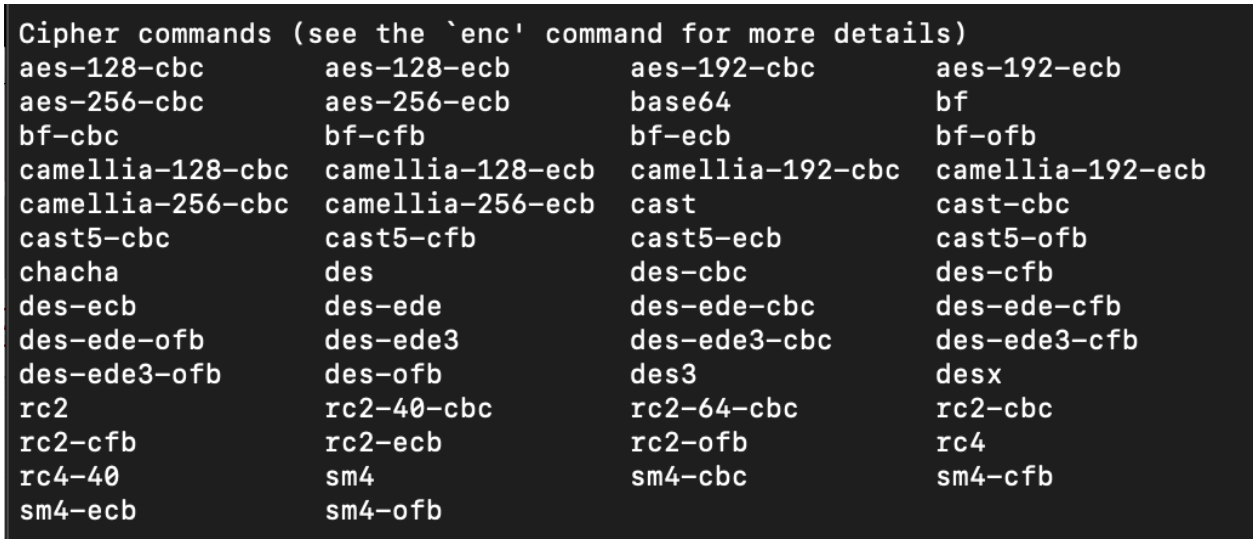
Task 1 was simple; I already had OpenSSL pre-installed on my Mac, so I was able to dive straight into using it:

Open SSL:



For Part B, I entered the cipher, digest, and speed commands. Here is what was shown on the terminal:

Cipher commands:



Digest commands:

```
Message Digest commands (see the `dgst' command for more details)
gost-mac          md4               md5               md_gost94
ripemd160         sha1              sha224            sha256
sha384            sha512            sm3               sm3WithRSAEncryption
streebog256       streebog512       whirlpool
```

Below are the Speed and Speed RSA commands. It was time-consuming due to the extensive output on the screen, so I have provided a screenshot of the initial lines:

Speed command:

```
OpenSSL> speed
Doing md4 for 3s on 16 size blocks: 44864180 md4's in 3.00s
Doing md4 for 3s on 64 size blocks: 28394733 md4's in 3.00s
Doing md4 for 3s on 256 size blocks: 12278351 md4's in 3.00s
Doing md4 for 3s on 1024 size blocks: 3752587 md4's in 3.00s
Doing md4 for 3s on 8192 size blocks: 500896 md4's in 3.00s
Doing md5 for 3s on 16 size blocks: 23907839 md5's in 3.00s
Doing md5 for 3s on 64 size blocks: 14163765 md5's in 3.00s
Doing md5 for 3s on 256 size blocks: 6164402 md5's in 3.00s
Doing md5 for 3s on 1024 size blocks: 1889574 md5's in 3.00s
```

Speed rsa2048 command:

```
OpenSSL> speed rsa2048
Doing 2048 bit private rsa's for 10s: 6763 2048 bit private RSA's in 9.99s
Doing 2048 bit public rsa's for 10s: 466733 2048 bit public RSA's in 9.99s
LibreSSL 3.3.6
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.001478s 0.000021s    676.7  46698.1
```

3a: Encrypted password CBC:

```
OpenSSL> enc -aes-128-cbc -base64 -in myfile.txt
[enter aes-128-cbc encryption password:
[Verifying - enter aes-128-cbc encryption password:
U2FsdGVkX18+BQ7ly4h+d+6nB5GxH2tAwWNkhbWLbvCjnul0l2Xp9vKUZZlzqnOC
OpenSSL>
```

3b: encrypted password ctr

```
OpenSSL> enc -aes-256-ctr -base64 -in myfile.txt
[enter aes-256-ctr encryption password:
[Verifying - enter aes-256-ctr encryption password:
U2FsdGVkX1/fKRzY+ddnLbO6I+TR7pSz9FFuawWxptFRtyqIGf3oRxA=
OpenSSL>
```

3c: encrypted password DES:

```
OpenSSL> enc -des -in myfile.txt -out enc
[enter des-cbc encryption password:
[Verifying - enter des-cbc encryption password:
OpenSSL>
OpenSSL> ^Z
[1]+  Stopped                 openssl
[MacBook-Pro-9:~ florindaponari$ ls
Applications            Pictures                list-commands
Desktop                 Public                  list-digest-commands
Documents               Sites                   main.dSYM
Downloads               enc                     myfile.txt
Library                 heartshapedsunglasses   speed
Movies                  help
Music                   list-cipher-commands
[MacBook-Pro-9:~ florindaponari$ cat myfile.txt
I will encrypt something
[MacBook-Pro-9:~ florindaponari$ cat enc
Salted__?'??e0<6"."?
                  ??ru???C?kşgj?n????
```

RSA, generating private and public keys:

```
OpenSSL> rsa -in keya -pubout -out publica
writing RSA key
```

```
MacBook-Pro-9:~ florindaponari$ openssl
OpenSSL> genrsa -out keya
Generating RSA private key, 2048 bit long modulus
.....................................................
.....................................................
e is 65537 (0x10001)
```

```
MacBook-Pro-9:~ florindaponari$ cat keya
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAy++2ryh/LcJm97hLnt1p7NbRoGw790f3If43AfB/p5peDNbF
ERzofz+yP0md3KdFyhIAHh0u4xpepaxQOfWxnVLCkfVIQa8K+44IVFCAlQTIiMHR
el13fXYUQ15xTHyUr5aqHelCa0Zo0aAfS6AxDOUHqtjQV5EvGTN/EAIFEcHfg7e3
uiSCsjIclaGCNqeIoXPHeLqem5x8ID/gGz1Bn1r4ltQA9tCyG1iCyxHueie/PLNt
NRCwvCD68zJILVB61lwG4nhghBFyJAtE3PKcrrrP4WyssFHTd/5wtzpG6WwnZyUy
OoaRKsVUmecwGCoj8ToMm9SyDGpbITba6yC0cwIDAQABAoIBADEc9sryHsa/ZZrM
6HFG82aVSYCv8DUaQLsQBOFTsPqFI5dF9cmsQ5W5mkzBhuIH6rfCZALcllkdFaQi
fmOm9vJjZvZNsuwWord4OitMA9tffofo3XDxp4AweqrKyNBZxFkRGMEBVvyMxVuO
i1rkJkvrsFAFzQ75qPURwMzzH5NpxfByC8FTFRHxn/tPo/CaS88Y93lh77MER+39
zuWGNKURV+JYfVhi06hPURKvk/E3f5gNBdIcjCeRLPjoquZkDK5+CroVo8sg1tSx
RecW7aymxx8omAoT/s2HMkTBMCaJlJH061wNbQRv+AMSzxCq74FR/Fqrfov2sqpM
to+jw/kCgYEA9rlI2Dk0sW5n9NIh5PfUVjS4TbzjZwufh8n+aSa8mJsBEFo6ZQNK
7xhv/SlTRYcGlgPQGKO1aZ0AicaxWj4H+ONN3ZT4kmlUgim5FywWi9DcKmQtIpDU
oduzltS/VVl3Hkv9QcSf53Y4/Ill41tuLzWARxXGReL2OGm6S3kI9/0CgYEA05qZ
wWSzT30Ao7Ob6f6S5O98OXsRUNFWt3miwQKYmSIZTcM8bi40IDmzJuxcwx2q2wRK
9j8VNAM5FnRAHhMdLtAv1JmUy7Uogk5OcC/ZP1f/tNnBYJbFLhUQ/2skdc7tcWAq
AqTDmBvagDVPpexXAACF21AL1wtQHfbePUT+8S8CgYBztNtLxDBJEd0g/Lc2mV90
ek8A7he7iMDtPrbYe3kxHGh4UchW/R9UWCKVGJ9b2QImsm0SURxQBnhtJMih6Lt2
ZqtqwmV8zHb8gDK0glGkinPmUYq1TjTaH53um6GLlmR9yedgw6S2OURqcslcwdE/
xJh/Pd/5gII2XKcoo/+nxQKBgQChrkALJ8FmW/HQIx1fhjeBG4eEeAc8BCfJGvSD
m1Mztn01EJcs7LrBYLUFaZZbM2Dn5fjM/FhcHJZTFNIk6J/ktAfH3u11VN816VU/
nR/WqYtlCPkbrzfMRSZustPx/AErHroauQAGSkoCqjlIYzP6eWvNrfTOBAPcUVNv
5zcQ2wKBgQCX4zc9ZtIlF7m9xXXrhGWJi15DJKvTdnv+FX723XtljpfQYuycYJ3I
I1uDIb/kuuq8R0oZEcfeWge9pEaFod5ixl6+6Ao+l9pmTW4KQ8dfzyo3ERpJMQnC
mXlYP+2wZAiMf59QiyZK61zPwjzw265bqlj//4IQ5VbnEF2GQhRgvQ==
-----END RSA PRIVATE KEY-----
```

```
MacBook-Pro-9:~ florindaponari$ cat publica
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAy++2ryh/LcJm97hLnt1p
7NbRoGw790f3If43AfB/p5peDNbFERzofz+yP0md3KdFyhIAHh0u4xpepaxQOfWx
nVLCkfVIQa8K+44IVFCAlQTIiMHRel13fXYUQ15xTHyUr5aqHelCa0Zo0aAfS6Ax
DOUHqtjQV5EvGTN/EAIFEcHfg7e3uiSCsjIclaGCNqeIoXPHeLqem5x8ID/gGz1B
n1r4ltQA9tCyG1iCyxHueie/PLNtNRCwvCD68zJILVB61lwG4nhghBFyJAtE3PKc
rrrP4WyssFHTd/5wtzpG6WwnZyUyOoaRKsVUmecwGCoj8ToMm9SyDGpbITba6yC0
cwIDAQAB
-----END PUBLIC KEY-----
```

4: exchange of encrypted data:

This was the most challenging part for me: I followed the tutorial, and when it came to encrypting, it kept saying “file not found,” as the necessary files were present and contained the appropriate certificate. I entered the appropriate commands that should have given the decrypted message.

The public key for both A and B:

```
keypair.pem
MacBook-Pro-9:A florindaponari$ rm keypair.pem
MacBook-Pro-9:A florindaponari$ ls
MacBook-Pro-9:A florindaponari$ openssl genrsa -out keypairA.pem 2048
Generating RSA private key, 2048 bit long modulus
.................................+++++
.................................+++++
e is 65537 (0x10001)
MacBook-Pro-9:A florindaponari$ ls
keypairA.pem
MacBook-Pro-9:A florindaponari$ openssl rsa -in keypairA.pem -pubout -out publicA.pem
writing RSA key
MacBook-Pro-9:A florindaponari$ ls
keypairA.pem    publicA.pem
MacBook-Pro-9:A florindaponari$ cat publicA.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2jqz/n+B5x5Hzq8Gi6PB
NRxV1ottS1TXTjpWTFKvaav7j2i6NPPRLlBDyIVcszNT3a9T/yxP4pyUXgPDRJx4
UvhZNAkuzOrPeGH2UtDjdfZPDk4JAlf4gHvyN8Popa0xZWxxVAptVlMMbYzop/+5
C7LHcKDoeC5/oJ6JyPOB9SL3kGqEUXivN8xcQ8mlNx84/Z4Pxp4ZZ/8FzZf1y487
vW9oTftoLklmkV5eCaOsISrkHqzQ9DkID7TJ7vabkbZxXelptMs23HdXUml9NPaq
eQ0BEcdGxHVUMe8rOif+1VjAzwpXMETTp2Qx3v8BUGKYCKJXXNaaEQsKvlECHH2B
qwIDAQAB
-----END PUBLIC KEY-----
MacBook-Pro-9:A florindaponari$
```

```
MacBook-Pro-9:~ florindaponari$ mkdir B
MacBook-Pro-9:~ florindaponari$ cd B
MacBook-Pro-9:B florindaponari$ openssl genrsa -out keypairB.pemm 2048
Generating RSA private key, 2048 bit long modulus
..............+++++
................................+++++
e is 65537 (0x10001)
MacBook-Pro-9:B florindaponari$ ls
keypairB.pemm
MacBook-Pro-9:B florindaponari$ openssl rsa -in keypairB.pemm -pubout -out publicB.pemm
writing RSA key
MacBook-Pro-9:B florindaponari$ ls
keypairB.pemm   publicB.pemm
MacBook-Pro-9:B florindaponari$ cat publicB.pemm
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzN+534mUqrhFviIWXTcX
7/lVBVgldLkXaD6z+LCCKqUzd9OPjwidpT9FWTAm1EjNT8prVCU6OX624n1AHDcn
2niQlb9XRXfNNrN/lWiyPkNKKkQuzDc64ZW1rbXI+NY45ZuE/MlM8+uUrUcr9flv
DoTZpj9F6O0jNCRyYz/jFASSBxCQWWm/e034HFb4EfC66CQB1o4vgE5AwzgLSPVw
qyUi8dhLZPbTaSP2UkuZnDv/wJnzA8OLCqcfWTP+87LHv8TevKEa2Y9ZkqnRVwaq
txfExFA5gGzVbfCZcNxG+3C5NEx8fwWHJP8istACpxlB9YVR0gCsPDLxOTBYw/4r
BwIDAQAB
-----END PUBLIC KEY-----
MacBook-Pro-9:B florindaponari$
```

File saving:

```
MacBook-Pro-9:A florindaponari$ Hey there, I'm just playing my records on my turntable
>
MacBook-Pro-9:A florindaponari$ nano
MacBook-Pro-9:A florindaponari$ ls
keypairA.pem    msg             publibB.pemm    publicA.pem
MacBook-Pro-9:A florindaponari$ cat msg
Hey there, I'm just listening to my records on my turntable
MacBook-Pro-9:A florindaponari$
```

Encryption:

```
MacBook-Pro-9:A florindaponari$ openssl rsautl -encrypt -in msg -out enc -inkey publicB.pem
MacBook-Pro-9:A florindaponari$ openssl rsautl -encrypt -in msg -out enc -inkey publicB.pem -pubin
```

Decryption:

```
keypairA.pem    msg             publicA.pem     publicB.pem
MacBook-Pro-9:A florindaponari$ openssl rsatul -decrypt -inkey keypairA.pem -in msg_encrypted -out msg_decrypted
```

Sources:

"What Is OpenSSL?" *SSL Dragon*, www.ssldragon.com/blog/what-is-openssl/. Accessed 24 Mar. 2024.