Shiming Huang
CSCI360 Project 2
Report:
Some of the things I've learned from the video was mainly openssl based on linux such as encryption with any of the listed encryption methods as well as decryption of it. Along with how to create text files, copying/sharing or linking a public or private key with 2 directories(folders) and allowing them to encrypt and decrypt messages that are created and then sent to the other. The video also showed how to sign/refine signatures in files. I also learned more and have an indelph understanding of how to use command prompt better since I haven't done that before. Since this video was using linux while I was using windows, I had to learn the equivalent or alternative similar methods that were used that can work on my end. All in all it was a very interesting and fun experience.

Part 1: listing commands
Openssl list all commands, including ciphers

```
C:\OpenSSL-Win64\bin>openssl help
Standard commands
asn1parse         ca                ciphers           cms
crl               crl2pkcs7         dgst              dhparam
dsa               dsaparam          ec                ecparam
enc               engine            errstr            gendsa
genpkey           genrsa            help              list
nseq              ocsp              passwd            pkcs12
pkcs7             pkcs8             pkey              pkeyparam
pkeyutl           prime             rand              rehash
req               rsa               rsautl            s_client
s_server          s_time            sess_id           smime
speed             spkac             srp               storeutl
ts                verify            version           x509

Message Digest commands (see the `dgst' command for more details)
blake2b512        blake2s256        gost              md4
md5               mdc2              rmd160            sha1
sha224            sha256            sha3-224          sha3-256
sha3-384          sha3-512          sha384            sha512
sha512-224        sha512-256        shake128          shake256
sm3

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       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      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          sm4-cbc
sm4-cfb           sm4-ctr           sm4-ecb           sm4-ofb
```

Part 2: speed test

Sample of speed on my computer with openssl, there is a lot more but it'll be too much to fit them all in here

```
C:\Users\Shiming>openssl speed
Doing mdc2 for 3s on 16 size blocks: 2653777 mdc2's in 2.97s
Doing mdc2 for 3s on 64 size blocks: 805521 mdc2's in 3.02s
Doing mdc2 for 3s on 256 size blocks: 211136 mdc2's in 3.02s
Doing mdc2 for 3s on 1024 size blocks: 53620 mdc2's in 3.02s
Doing mdc2 for 3s on 8192 size blocks: 6745 mdc2's in 3.00s
Doing mdc2 for 3s on 16384 size blocks: 3373 mdc2's in 2.98s
Doing md4 for 3s on 16 size blocks: 9582843 md4's in 3.00s
Doing md4 for 3s on 64 size blocks: 7389491 md4's in 3.02s
Doing md4 for 3s on 256 size blocks: 4752115 md4's in 3.02s
Doing md4 for 3s on 1024 size blocks: 1893060 md4's in 2.98s
Doing md4 for 3s on 8192 size blocks: 285352 md4's in 3.02s
Doing md4 for 3s on 16384 size blocks: 143554 md4's in 3.00s
Doing md5 for 3s on 16 size blocks: 22003870 md5's in 3.00s
Doing md5 for 3s on 64 size blocks: 13258613 md5's in 3.02s
Doing md5 for 3s on 256 size blocks: 5901723 md5's in 3.02s
Doing md5 for 3s on 1024 size blocks: 1833673 md5's in 3.02s
Doing md5 for 3s on 8192 size blocks: 245964 md5's in 2.95s
Doing md5 for 3s on 16384 size blocks: 123839 md5's in 2.95s
Doing hmac(md5) for 3s on 16 size blocks: 7506139 hmac(md5)'s in 2.98s
Doing hmac(md5) for 3s on 64 size blocks: 6242281 hmac(md5)'s in 3.02s
Doing hmac(md5) for 3s on 256 size blocks: 3926154 hmac(md5)'s in 3.02s
Doing hmac(md5) for 3s on 1024 size blocks: 1586899 hmac(md5)'s in 3.02s
Doing hmac(md5) for 3s on 8192 size blocks: 240704 hmac(md5)'s in 3.02s
Doing hmac(md5) for 3s on 16384 size blocks: 122435 hmac(md5)'s in 3.02s
Doing sha1 for 3s on 16 size blocks: 23108418 sha1's in 3.00s
Doing sha1 for 3s on 64 size blocks: 13839982 sha1's in 3.00s
Doing sha1 for 3s on 256 size blocks: 7188271 sha1's in 3.02s
Doing sha1 for 3s on 1024 size blocks: 2449438 sha1's in 3.00s
Doing sha1 for 3s on 8192 size blocks: 341503 sha1's in 3.02s
Doing sha1 for 3s on 16384 size blocks: 173456 sha1's in 3.02s
```

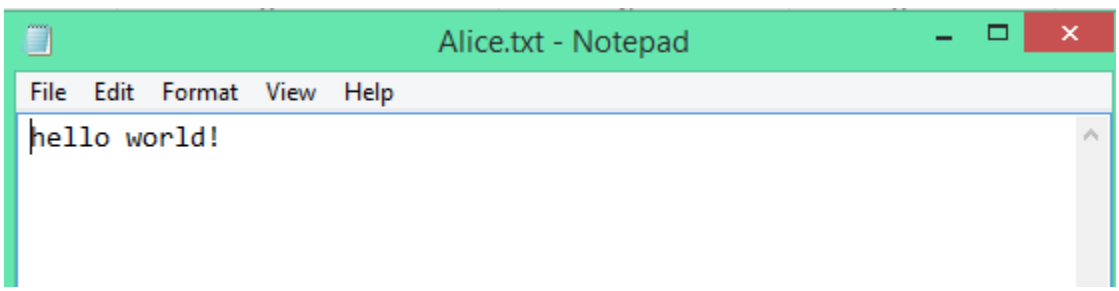
Speed test comparing rsa and aes. Couldn't specifically pick out just 1024 bits for aes so I had it to print all the aes out.

```
C:\Users\Shiming>openssl speed rsa1024
Doing 1024 bits private rsa's for 10s: 71268 1024 bits private RSA's in 9.78s
Doing 1024 bits public rsa's for 10s: 1029700 1024 bits public RSA's in 9.78s
```

```
C:\Users\Shiming>openssl speed aes
Doing aes-128 cbc for 3s on 16 size blocks: 43510402 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 64 size blocks: 11469021 aes-128 cbc's in 2.98s
Doing aes-128 cbc for 3s on 256 size blocks: 2893036 aes-128 cbc's in 3.02s
Doing aes-128 cbc for 3s on 1024 size blocks: 731169 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 8192 size blocks: 92160 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 16384 size blocks: 45777 aes-128 cbc's in 3.00s
Doing aes-192 cbc for 3s on 16 size blocks: 38146898 aes-192 cbc's in 3.00s
Doing aes-192 cbc for 3s on 64 size blocks: 9824258 aes-192 cbc's in 2.95s
Doing aes-192 cbc for 3s on 256 size blocks: 2519527 aes-192 cbc's in 2.98s
Doing aes-192 cbc for 3s on 1024 size blocks: 631862 aes-192 cbc's in 2.98s
Doing aes-192 cbc for 3s on 8192 size blocks: 79412 aes-192 cbc's in 3.00s
Doing aes-192 cbc for 3s on 16384 size blocks: 39662 aes-192 cbc's in 2.98s
Doing aes-256 cbc for 3s on 16 size blocks: 33498230 aes-256 cbc's in 2.98s
Doing aes-256 cbc for 3s on 64 size blocks: 8779380 aes-256 cbc's in 3.02s
Doing aes-256 cbc for 3s on 256 size blocks: 2223147 aes-256 cbc's in 2.97s
Doing aes-256 cbc for 3s on 1024 size blocks: 558385 aes-256 cbc's in 2.98s
Doing aes-256 cbc for 3s on 8192 size blocks: 69593 aes-256 cbc's in 3.02s
Doing aes-256 cbc for 3s on 16384 size blocks: 34913 aes-256 cbc's in 2.98s
```

Part 3: encryption and decryption

Created a message inside a .txt file

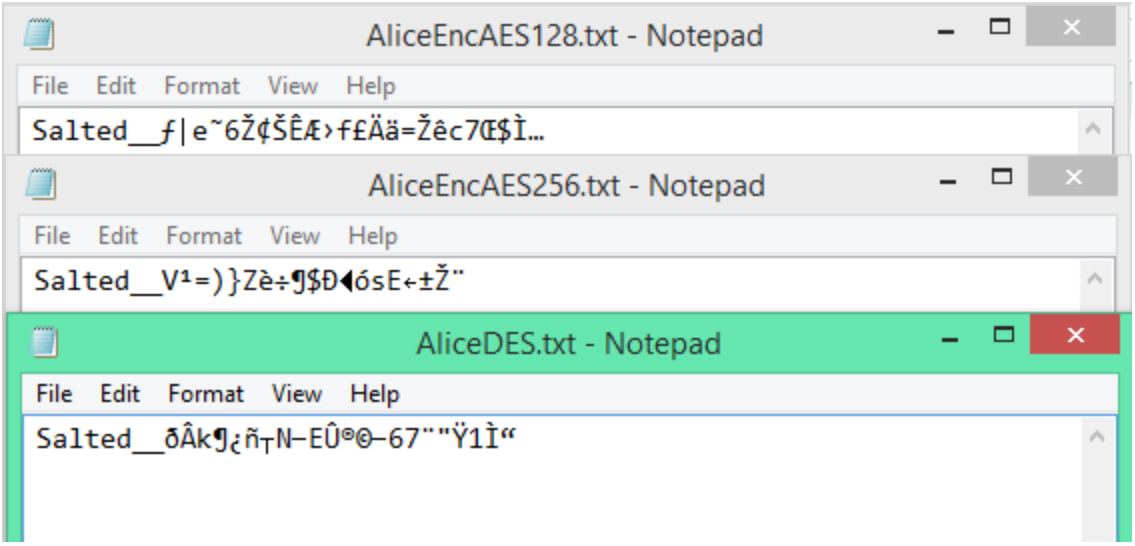


Encryption using aes-128 cbc, aes-256 ctr and des

```
C:\OpenSSL-Win64\bin>openssl aes-128-cbc -in Alice.txt -out AliceEncAES128.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\OpenSSL-Win64\bin>openssl aes-256-ctr -in Alice.txt -out AliceEncAES256.txt
enter aes-256-ctr encryption password:
Verifying - enter aes-256-ctr encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\OpenSSL-Win64\bin>openssl des -in Alice.txt -out AliceDES.txt
enter des-cbc encryption password:
Verifying - enter des-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

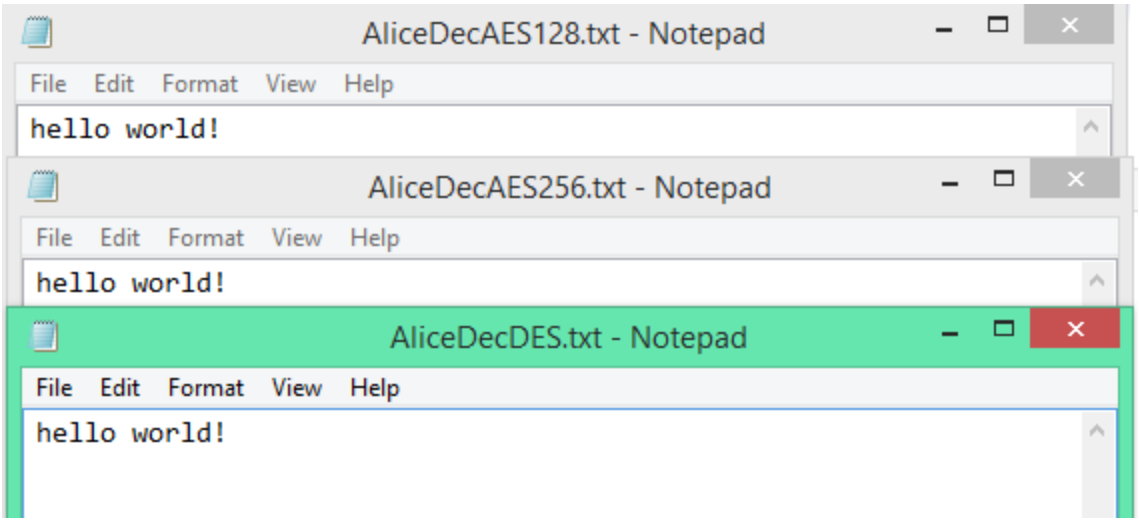


Decryption

```
C:\OpenSSL-Win64\bin>openssl aes-128-cbc -d -in AliceEncAES128.txt -out AliceDec
AES128.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\OpenSSL-Win64\bin>openssl aes-256-ctr -d -in AliceEncAES256.txt -out AliceDec
AES256.txt
enter aes-256-ctr decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\OpenSSL-Win64\bin>openssl des -d -in AliceDES.txt -out AliceDecDES.txt
enter des-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```



Generating private key(windows) and seeing all of its content such as modulus, 2prime, etc.

```
C:\OpenSSL-Win64\bin>openssl genrsa -out privatekeyA.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....................................................................
.....+++++
e is 65537 (0x010001)

C:\OpenSSL-Win64\bin>type privatekeyA.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAypSdKuqhixOJzI0PesuwOfZ2a/hnGf3CPRcQYu2pclSMZaSi
mwH9X1TJAMoBzJro3eY8FaL1EKRq9Fy0yRaTYXixnzH3ZHiXNyZaF8gFvC+AERWS
OFTmPIz/S1SQupW/wrZHXGz2Q4G3rA6hpUJD5bydIvnLmSNIKMWFTgV89WHIqkm5
Ydz9bgHsX82oLQdS11TDGGN+h88pPbTD8okjTnCcZJ4d4h4VmcjYaY0oYJOraFQb
UMCeKIXjIztIV6UvE39bRDl25aisgA+Zj8VyuEfZCZfAGTqncIgB3+gCxvxuhx3C
Lvin0PPsNG55odzM1G2XuFGEW49/OH4EeElYqwIDAQABAoIBAQCinamMScxjV8vF
53YFnHxaCv4EreIVex3w7BK7VwwuOrB45y+gPniljenMOFw5YiCL/8U7mRQfx9wT
TFfTE8w68iejIvj0Ldk07BRmAqqX5YloLWiyz3QCJyzHy5piLNNB+hBOoSkOB5Lp
CtW3Uvot8Loy2VIWG7xQ/j2D4+PrFKPSiBxYQwOn1Q1Q4JzYSrwzlCTFKGcLJWEX
q7bnzhqMAdwDIRrbQuslmVZGWBePRU/+L8vi1HagVjhhGAhAMTFKo4NBhHdLwRwB
H40Roqf8cl5y6sLHUcNk8CKvKd36+e0OND/4zt/jRYavJPt6KnEKRptiDxrgVqB9
p4A4u0ABAoGBAP1B7BPvaOV7UV3dLPvT6BMrG+r/vZWWbHpBtJ1GyNFzGzcSatwl
tBfImuLCqZjuxh47cB7C7YkZ8qWCk3hHdRIZlBYTEMaqxsECE0GYBbDBViuDtQzG
0WO88dps0Uc1ovjRniWxV0yMKvj4o1KY/nian5EGr3yTQO3dmjt3kDlrAoGBAMzG
NJ6/2SybDl780VibQ+X7zec7xrT+Z8M99mz12TkgVYMx0zBQM3WFylH+fPPYv2J9
3WW6g4rVhRsZuV0GkLqVOV973arbTwpQOfBU8o19+pQdgoRc5mjOvfsSy00j61QI
zRb6Idv1f48Bp3EBNaHUAYqLbljWFKMCF35psu3BAoGAeSyahStOuEYZt1yFEnTg
nJNKx1TX5APWIWTgw0PEABsgEWiZgn3zvlF4cjkLpqXQW1S7H6zKtIeaxsBESQ9/
n0pnxWwX4caCVITmq8LOLjVDQozBCrzyINOdsJHSTE0Ijh7VmVlR2MlUIk3s7y9O
ymvJHz3gpy1ZqrYV/pAFjMMCgYA/AbmNIxCqpecNGRM9BUEmEY1wWbcxV5oLvBQH
wLme1zEYlgsaMYMz5P+0rO47hXSV1I57z6DL5hj31rmvQubjUij+FTnGaYZvcPHy
3eBn4txoGJBhM8h9dSeFimndiXDOQYsBR+OFL4uZL/QaVDnxWi9DNtW0JeLhqTIb
ZmB0wQKBgG69eJr+xHK6gvNdoN7CaQXTmX1rqK3FgAfQI47e/D53xJkcAHr0SV9g
6gwlJkAdgGf/yl/I3FNzTQzSGMBABiS0onqpZvtDzpvJZbO5VIITEAIPfaPwEZyV
N8z9v0o04scrkRpRJbrwcidH6k0qMvppFg4LIUcjKgF6Fc7x3D6q
-----END RSA PRIVATE KEY-----
```

```
C:\OpenSSL-Win64\bin>openssl rsa -in privatekeyA.pem -text
RSA Private-Key: (2048 bit, 2 primes)
modulus:
    00:ca:94:9d:2a:ea:a1:8b:13:89:cc:8d:0f:7a:cb:
    b0:39:f6:76:6b:f8:67:19:fd:c2:3d:17:10:62:ed:
    a9:72:54:8c:65:a4:a2:9b:01:fd:5f:54:c9:00:ca:
    01:cc:9a:e8:dd:e6:3c:15:a2:f5:10:a4:6a:f4:5c:
    b4:c9:16:93:61:78:b1:9f:31:f7:64:78:97:37:26:
    5a:17:c8:05:bc:2f:80:11:15:92:38:54:e6:3c:8c:
    ff:4b:54:90:ba:95:bf:c2:b6:47:5c:6c:f6:43:81:
    b7:ac:0e:a1:a5:42:43:e5:bc:9d:22:f9:cb:99:23:
    48:28:c5:85:4e:05:7c:f5:61:c8:aa:49:b9:61:dc:
    fd:6e:01:ec:5f:cd:a8:2d:07:52:d6:54:c3:18:63:
    7e:87:cf:29:3d:b4:c3:f2:89:23:4e:70:9c:64:9e:
    1d:e2:1e:15:99:c8:d8:69:8d:28:60:93:ab:68:54:
    1b:50:c0:9e:28:85:e3:23:3b:48:57:a5:2f:13:7f:
    5b:44:39:76:e5:a8:ac:80:0f:99:8f:c5:72:b8:47:
    d9:09:97:c0:19:3a:a7:70:88:01:df:e8:02:c6:fc:
    6e:87:1d:c2:2e:f8:a7:d0:f3:ec:34:6e:79:a1:dc:
    cc:d4:6d:97:b8:51:84:5b:8f:7f:38:7e:04:78:49:
    58:ab
publicExponent: 65537 (0x10001)
privateExponent:
    00:a2:9d:a9:8c:49:cc:63:57:cb:c5:e7:76:05:9c:
    7c:5a:0a:fe:04:ad:e2:15:7b:1d:f0:ec:12:bb:57:
    0c:2e:3a:b0:78:e7:2f:a0:3e:78:a5:8d:e9:cc:38:
    5c:39:62:20:8b:ff:c5:3b:99:14:1f:c7:dc:13:4c:
    57:d3:13:cc:3a:f2:27:a3:22:f8:f4:2d:d9:34:ec:
    14:66:02:aa:97:e5:89:68:2d:68:b2:cf:74:02:27:
    2c:c7:cb:9a:62:2c:d3:41:fa:10:4e:a1:29:0e:07:
    92:e9:0a:d5:b7:52:fa:2d:f0:ba:32:d9:52:16:1b:
    bc:50:fe:3d:83:e3:e3:eb:14:a3:d2:88:1c:58:43:
    03:a7:d5:09:50:e0:9c:d8:4a:bc:33:94:24:c5:28:
    67:0b:25:61:17:ab:b6:e7:ce:1a:8c:01:dc:03:21:
    1a:db:42:eb:25:99:56:46:58:17:8f:45:4f:fe:2f:
    cb:e2:d4:76:a0:56:38:61:18:08:40:31:31:4a:a3:
    83:41:84:77:4b:c1:1c:01:1f:8d:11:a2:a7:fc:72:
    5e:72:ea:c2:c7:51:c3:64:f0:22:af:29:dd:fa:f9:
    ed:0e:34:3f:f8:ce:df:e3:45:86:af:24:fb:7a:2a:
    71:0a:46:9b:62:0f:1a:e0:56:a0:7d:a7:80:38:bb:
    40:01
prime1:
    00:fd:41:ec:13:ef:68:e5:7b:51:5d:dd:2c:fb:d3:
    e8:13:2b:1b:ea:ff:bd:95:96:6c:7a:41:b4:9d:46:
    c8:d1:73:1b:37:12:6a:dc:25:b4:17:c8:9a:e2:c2:
    a9:98:ee:c6:1e:3b:70:1e:c2:ed:89:19:f2:a5:82:
    93:78:47:75:12:19:94:16:13:10:c6:aa:c6:c1:02:
    13:41:98:05:b0:c1:56:2b:83:b5:0c:c6:d1:63:bc:
    f1:da:6c:d1:47:35:a2:f8:d1:9e:25:b1:57:4c:8c:
    2a:f8:f8:a3:52:98:fe:78:9a:9f:91:06:af:7c:93:
    40:ed:dd:9a:3b:77:90:39:6b
```

```
prime2:
    00:cc:c6:34:9e:bf:d9:2c:9b:0e:5e:fc:d1:58:9b:
    43:e5:fb:cd:e7:3b:c6:b4:fe:67:c3:3d:f6:6c:f5:
    d9:39:20:55:83:31:d3:30:50:33:75:85:ca:51:fe:
    7c:f3:d8:bf:62:7d:dd:65:ba:83:8a:d5:85:1b:19:
    b9:5d:06:90:ba:95:39:5f:7b:dd:aa:db:4f:0a:50:
    39:f0:54:f2:8d:7d:fa:94:1d:82:84:5c:e6:68:ce:
    bd:fb:12:cb:4d:23:eb:54:08:cd:16:fa:21:db:f5:
    7f:8f:01:a7:71:01:35:a1:d4:01:8a:8b:6e:58:d6:
    14:a3:02:17:7e:69:b2:ed:c1
exponent1:
    79:2c:9a:85:2b:4e:b8:46:19:b7:5c:85:12:74:e0:
    9c:93:4a:c7:54:d7:e4:03:d6:21:64:e0:c3:43:c4:
    00:1b:20:11:68:99:82:7d:f3:be:51:78:72:39:0b:
    a6:a5:d0:5b:54:bb:1f:ac:ca:b4:87:9a:c6:c0:44:
    49:0f:7f:9f:4a:67:c5:6c:17:e1:c6:82:54:84:e6:
    ab:c2:ce:2e:35:43:42:8c:c1:0a:bc:f2:20:d3:9d:
    b0:91:d2:4c:4d:08:8e:1e:d5:99:59:51:d8:c9:54:
    22:4d:ec:ef:2f:4e:ca:6b:c9:1f:3d:e0:a7:2d:59:
    aa:b6:15:fe:90:05:8c:c3
exponent2:
    3f:01:b9:8d:23:10:aa:a5:e7:0d:19:13:3d:05:41:
    26:11:8d:70:59:b7:31:57:9a:0b:bc:14:07:c0:b9:
    9e:d7:31:18:96:0b:1a:31:83:33:e4:ff:b4:ac:ee:
    3b:85:74:95:d4:8e:7b:cf:a0:cb:e6:18:f7:96:b9:
    af:42:e6:e3:52:28:fe:15:39:c6:69:86:6f:70:f1:
    f2:dd:e0:67:e2:dc:68:18:90:61:33:c8:7d:75:27:
    85:8a:69:dd:89:70:ce:41:8b:01:47:e3:85:2f:8b:
    99:2f:f4:1a:54:39:f1:5a:2f:43:36:d5:b4:25:e2:
    e1:a9:32:1b:66:60:74:c1
coefficient:
    6e:bd:78:9a:fe:c4:72:ba:82:f3:5d:a0:de:c2:69:
    05:d3:99:7d:6b:a8:ad:c5:80:07:d0:23:8e:de:fc:
    3e:77:c4:99:1c:00:7a:f4:49:5f:60:ea:0c:25:26:
    40:1d:80:67:ff:ca:5f:c8:dc:53:73:4d:0c:d2:18:
    c0:40:06:24:b4:a2:7a:a9:66:fb:43:ce:9b:c9:65:
    b3:b9:54:82:13:10:02:0f:7d:a3:f0:11:9c:95:37:
    cc:fd:bf:4a:34:e2:c7:2b:91:1a:51:25:ba:f0:72:
    27:47:ea:4d:2a:32:fa:69:16:0e:0b:21:47:23:2a:
    01:7a:15:ce:f1:dc:3e:aa
```

Part 4: exchanging encrypted data and decrypting it

Generating private key for both folder A and B

```
C:\OpenSSL-Win64\bin\A>openssl genrsa -out privatekeyA.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.............................................+++++
....................................................+++++
e is 65537 (0x010001)
```

```
C:\OpenSSL-Win64\bin\B>openssl genrsa -out privatekeyB.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.............................................................+++++
.........................+++++
e is 65537 (0x010001)
```

Creating public key

>openssl rsa -in privatekeyA.pem -pubout -out publickeyA.pem

>openssl rsa -in privatekeyB.pem -pubout -out publickeyB.pem

```
C:\OpenSSL-Win64\bin\A>type publickeyA.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqbfDSZ9uxooDfTkxxAyY
ISdpoioNAVpjrXYyN1uIuH6P+XGBZi3JxA/exFzHPxpWBxffHH8qxd2vojN/acVa
WYNZaE3PyjZNLnG0QVuUfvBXe2E75t+tYr3QAkTOchprfaVDIjrV23yntyKnVkI5
W8NvC1XOluhiz+fPrKk95ffJ86JHFEjoqI5RdZwEiWj5tmxBqdoeyNyDUU7Uvjtu
3FfAPEv6xHJXhO7UjBsJSLQv2+4QF4gZoQeQeVXFYv+haXrnowyI9NLlhdyCVdLl
lsNb0+leFJw5ottGrdcsKFetNBV2tMYGCxmwJ5q+U5Y+SEzo3voVbLQSU75xmwnN
NwIDAQAB
-----END PUBLIC KEY-----
```

```
C:\OpenSSL-Win64\bin\B>type publickeyB.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAy8xmz3n8eKeSbJZcjt7Z
U9/VXU/+IXIrx8mUYHSSTNR1ZtQtuQgcNKmFwcxxLO98x+izxGRYKZpIHwFzDjmm
SrBL67YUWCHOY/z/5Op8/t1Vnz5Muc/gPhSFnlEncC0mXXo89CwSdJ2EDDsQsjlQ
cBP3eCTLilEzREkRzOps7bSJ5k6eYLLCr71WjsHbciS7wxC+6vj/v6wAuO9X6PFp
pVqH6NNy3a3afbhvv5mdRvWCULgaELX/F0FbAxMUze537mc4UenMm5mYMYmdOSXF
O+gArJ6Fc/owv/IxItcAOJYEmhCSrQArk48ZbBIDBS1i1F+GZ5zZq1TxuhHqu10/
JwIDAQAB
-----END PUBLIC KEY-----
```

Linking/sharing public key with each directories(folder) A and B

For folder A to make a link to folder B's public key, type in cmd:

>mklink publickeyB.pem \openssl-win64\bin\b\publickeyB.pem

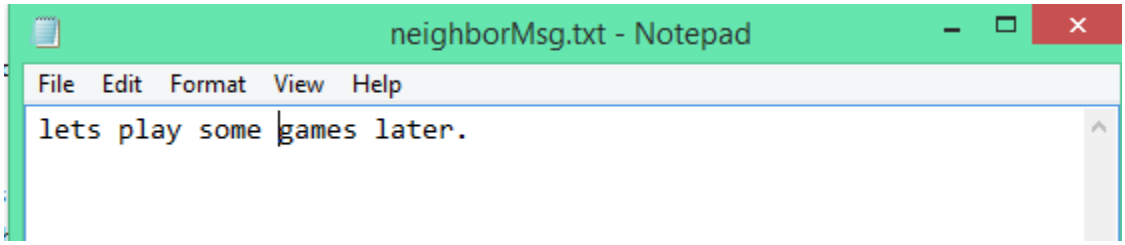
privatekeyA.pem	3/25/2021 8:13 PM	PEM File	2 KB
publickeyA.pem	3/25/2021 8:22 PM	PEM File	1 KB
publickeyB.pem	3/25/2021 9:08 PM	.symlink	0 KB

For folder B to make a link to folder A's public key:type in cmd

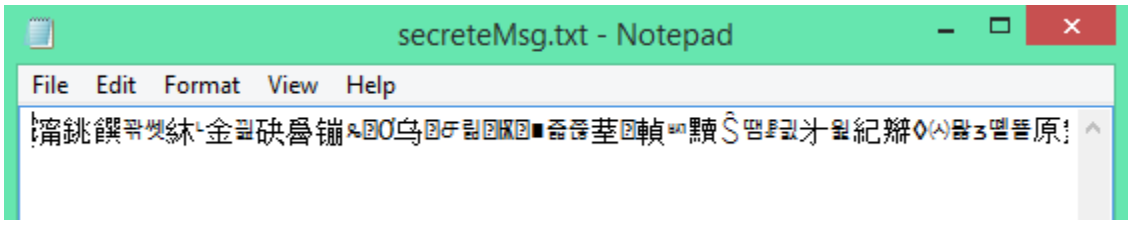
>mklink publickeyA.pem \openssl-win64\bin\b\publickeyA.pem

privatekeyB.pem	3/25/2021 8:13 PM	PEM File	2 KB
publickeyA.pem	3/25/2021 9:15 PM	.symlink	0 KB
publickeyB.pem	3/25/2021 8:23 PM	PEM File	1 KB

Now creating a message from A which will be encrypted using the public key from B so B can decrypt it.
The message:

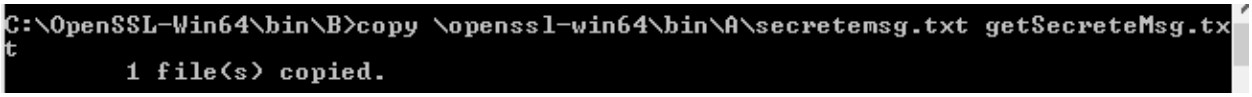


To encrypt:
>openssl rsautl -encrypt -in neighbormsg.txt -out secreteMsg.txt -inkey publickeyB.pem -pubin
Result inside the .txt file



Result in cmd

Since there isn't a "cp" command in windows, I have to get the message by using:
>copy \openssl-win64\bin\A\secretemsg.txt getSecreteMsg.txt



Now I decrypt it using the private key that is related to the public key by using:
>openssl rsautl -decrypt -in getsecretemsg.txt -out myNeighborMsg.txt -inkey privatekeyB.pem
And now the result is:

getSecreteMsg.txt	3/25/2021 9:29 PM	Text Document	1 KB
myNeighborMsg.txt	3/25/2021 9:57 PM	Text Document	1 KB
privatekeyB.pem	3/25/2021 8:13 PM	PEM File	2 KB
publickeyA.pem	3/25/2021 9:15 PM	.symlink	0 KB
publickeyB.pem	3/25/2021 8:23 PM	PEM File	1 KB