Week 4: Real-World Cryptography Labs

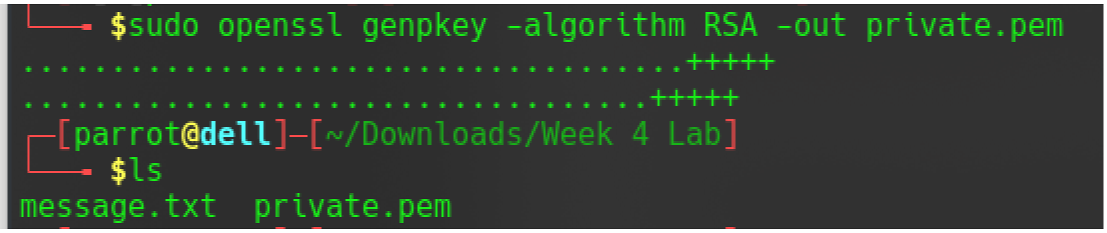
TASK 1: File Encryption with OpenSSL

RSA

1) Use the openssl command to generate the RSA private and public keys.

Generate the private key:

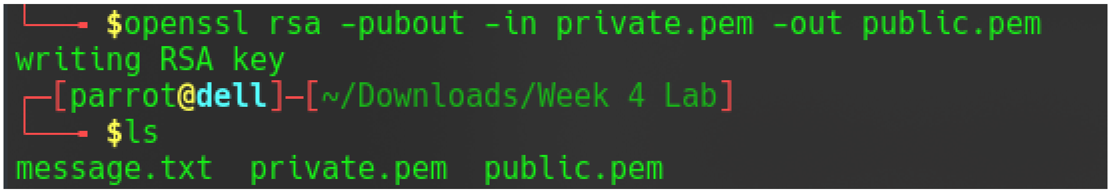
openssl genpkey -algorithm RSA -out private.pem



```
 └──╼ $sudo openssl genpkey -algorithm RSA -out private.pem
.................................+++++
...............................+++++
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $ls
message.txt  private.pem
```

2) Extract the public key from the private key:

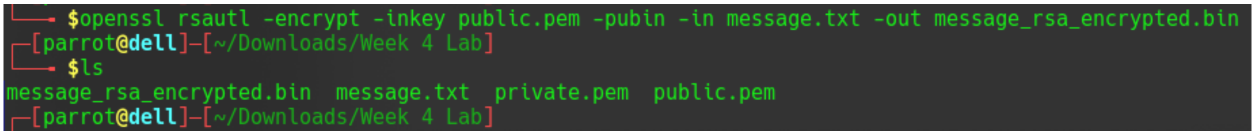
openssl rsa -pubout -in private.pem -out public.pem



```
└──╼ $openssl rsa -pubout -in private.pem -out public.pem
writing RSA key
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $ls
message.txt  private.pem  public.pem
```

3) Encrypt the File Using RSA

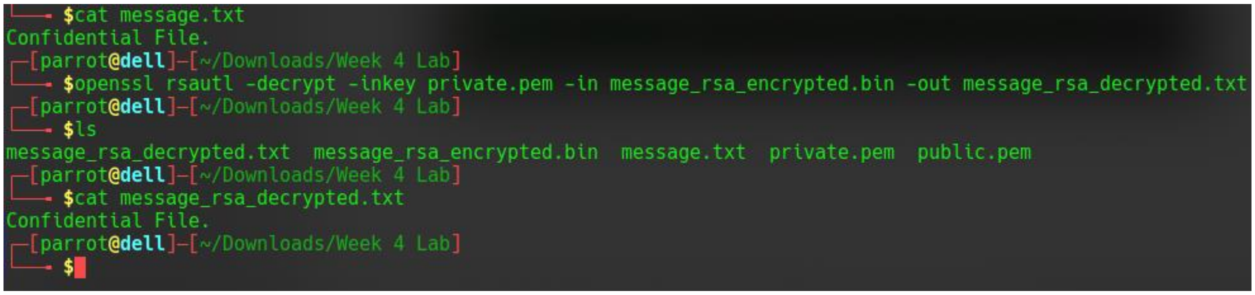
openssl rsautl -encrypt -inkey public.pem -pubin -in message.txt -out message_rsa_encrypted.bin



```
└──╼ $openssl rsautl -encrypt -inkey public.pem -pubin -in message.txt -out message_rsa_encrypted.bin
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $ls
message_rsa_encrypted.bin  message.txt  private.pem  public.pem
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
```

4) Decrypt the RSA Encrypted File Using the Private Key

openssl rsautl -decrypt -inkey private.pem -in message_rsa_encrypted.bin -out message_rsa_decrypted.txt

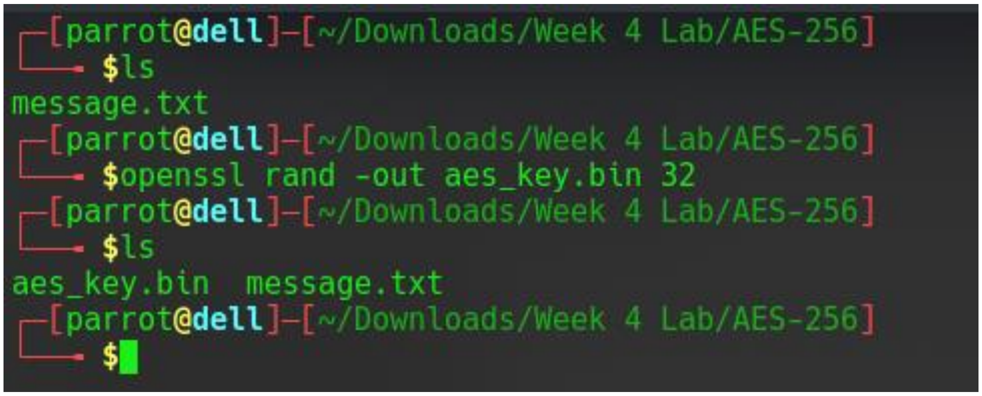


```
└──╼ $cat message.txt
Confidential File.
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $openssl rsautl -decrypt -inkey private.pem -in message_rsa_encrypted.bin -out message_rsa_decrypted.txt
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $ls
message_rsa_decrypted.txt  message_rsa_encrypted.bin  message.txt  private.pem  public.pem
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $cat message_rsa_decrypted.txt
Confidential File.
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $
```

A E S

1) Let’s generate a random symmetric key (AES-256)

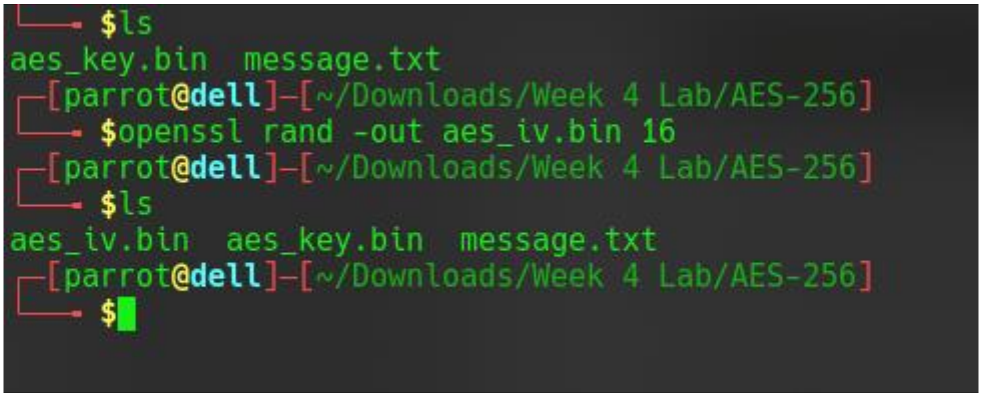
openssl rand -out aes_key.bin 32



```
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $ls
message.txt
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $openssl rand -out aes_key.bin 32
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $ls
aes_key.bin  message.txt
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $
```

2) Generate the AES IV (initialization vector)

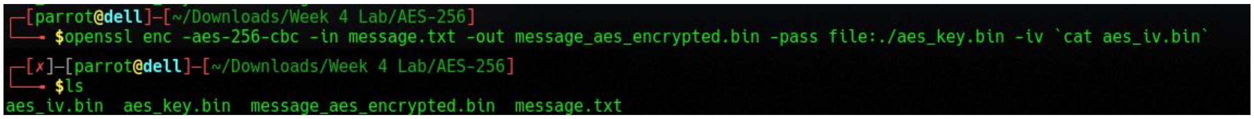
openssl rand -out aes_iv.bin 16



```
└──╼ $ls
aes_key.bin  message.txt
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $openssl rand -out aes_iv.bin 16
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $ls
aes_iv.bin  aes_key.bin  message.txt
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $
```

3) Encrypt the file using AES-256

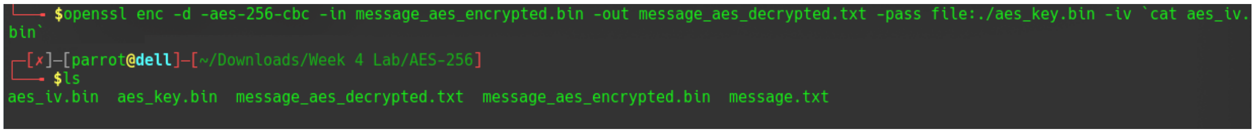
openssl enc -aes-256-cbc -in message.txt -out message_aes_encrypted.bin -pass file:./aes_key.bin -iv `cat aes_iv.bin`



```
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $openssl enc -aes-256-cbc -in message.txt -out message_aes_encrypted.bin -pass file:./aes_key.bin -iv `cat aes_iv.bin`
┌─[✗]─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $ls
aes_iv.bin  aes_key.bin  message_aes_encrypted.bin  message.txt
```

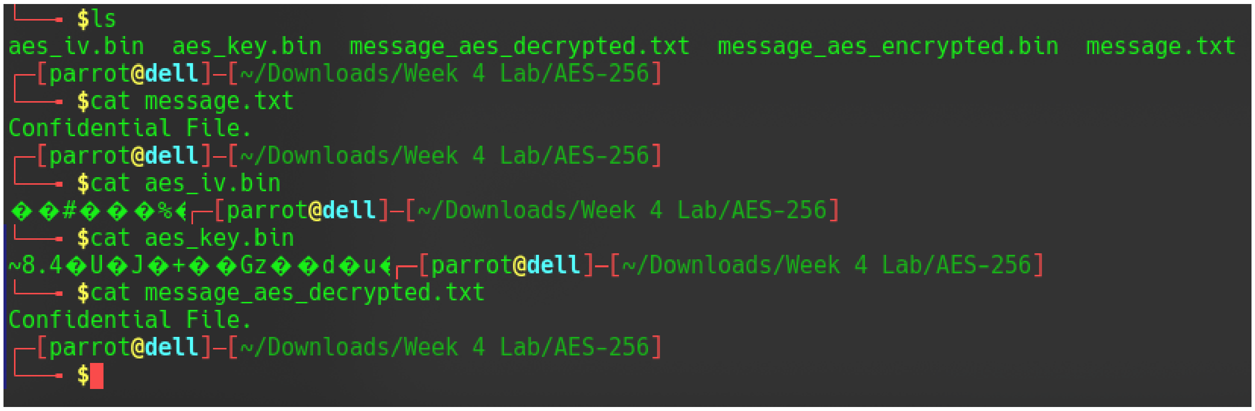
4) To decrypt the AES-encrypted file, let's execute the following command:

openssl enc -d -aes-256-cbc -in message_aes_encrypted.bin -out message_aes_decrypted.txt -pass file:./aes_key.bin -iv `cat aes_iv.bin`



```
└──╼ $openssl enc -d -aes-256-cbc -in message_aes_encrypted.bin -out message_aes_decrypted.txt -pass file:./aes_key.bin -iv `cat aes_iv.
bin`
┌─[✗]─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $ls
aes_iv.bin  aes_key.bin  message_aes_decrypted.txt  message_aes_encrypted.bin  message.txt
```

5) Let's check the content of the message_aes_decrypted txt file:

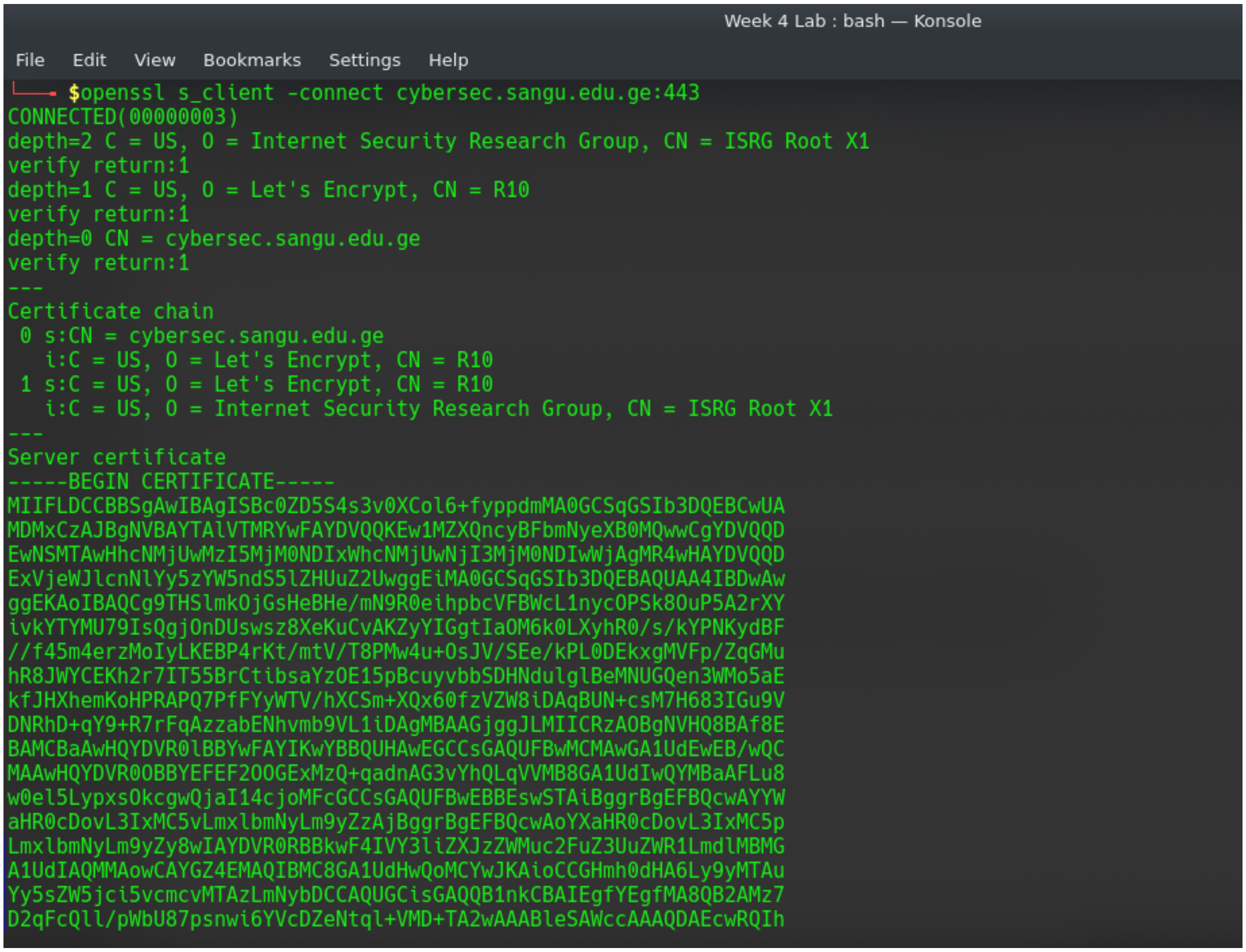


```
└──╼ $ls
aes_iv.bin  aes_key.bin  message_aes_decrypted.txt  message_aes_encrypted.bin  message.txt
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $cat message.txt
Confidential File.
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $cat aes_iv.bin
��#���%�┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $cat aes_key.bin
~8.4�U�J�+��Gz��d�u�┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $cat message_aes_decrypted.txt
Confidential File.
┌─[parrot@dell]─[~/Downloads/Week 4 Lab/AES-256]
└──╼ $
```

TASK 2: SSL/TLS in HTTPS

1) Inspecting HTTPS Website with OpenSSL

openssl s_client -connect cybersec.sangu.edu.ge:443



```
└── $openssl s_client -connect cybersec.sangu.edu.ge:443
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = R10
verify return:1
depth=0 CN = cybersec.sangu.edu.ge
verify return:1
---
Certificate chain
 0 s:CN = cybersec.sangu.edu.ge
   i:C = US, O = Let's Encrypt, CN = R10
 1 s:C = US, O = Let's Encrypt, CN = R10
   i:C = US, O = Internet Security Research Group, CN = ISRG Root X1
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFLDCCBBSgAwIBAgISBc0ZD5S4s3v0XCol6+fyppdmMA0GCSqGSIb3DQEBCwUA
MDMxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQwwCgYDVQQD
EwNSMTAwHhcNMjUwMzI5MjM0NDIxWhcNMjUwNjI3MjM0NDIwWjAgMR4wHAYDVQQD
ExVjeWJlcnNlYy5zYW5ndS5lZHUuZ2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQCg9THSlmkOjGsHeBHe/mN9R0eihpbcVFBWcL1nycOPSk8OuP5A2rXY
ivkYTYMU79IsQgjOnDUswsz8XeKuCvAKZyYIGgtIaOM6k0LXyhR0/s/kYPNKydBF
//f45m4erzMoIyLKEBP4rKt/mtV/T8PMw4u+OsJV/SEe/kPL0DEkxgMVFp/ZqGMu
hR8JWYCEKh2r7IT55BrCtibsaYzOE15pBcuyvbbSDHNdulglBeMNUGQen3WMo5aE
kfJHXhemKoHPRAPQ7PfFYyWTV/hXCSm+XQx60fzVZW8iDAqBUN+csM7H683IGu9V
DNRhD+qY9+R7rFqAzzabENhvmb9VL1iDAgMBAAGjggJLMIICRzAOBgNVHQ8BAf8E
BAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMAwGA1UdEwEB/wQC
MAAwHQYDVR0OBBYEFEF2OOGExMzQ+qadnAG3vYhQLqVVMB8GA1UdIwQYMBaAFLu8
w0el5LypxsOkcgwQjaI14cjoMFcGCCsGAQUFBwEBBEswSTAiBggrBgEFBQcwAYYW
aHR0cDovL3IxMC5vLmxlbmNyLm9yZzAjBggrBgEFBQcwAoYXaHR0cDovL3IxMC5p
LmxlbmNyLm9yZy8wIAYDVR0RBBkwF4IVY3liZXJzZWMuc2FuZ3UuZWR1LmdlMBMG
A1UdIAQMMAowCAYGZ4EMAQIBMC8GA1UdHwQoMCYwJKAioCCGHmh0dHA6Ly9yMTAu
Yy5sZW5jci5vcmcvMTAzLmNybDCCAQUGCisGAQQB1nkCBAIEgfYEgfMA8QB2AMz7
D2qFcQll/pWbU87psnwi6YVcDZeNtql+VMD+TA2wAAABleSAWccAAAQDAEcwRQIh
```

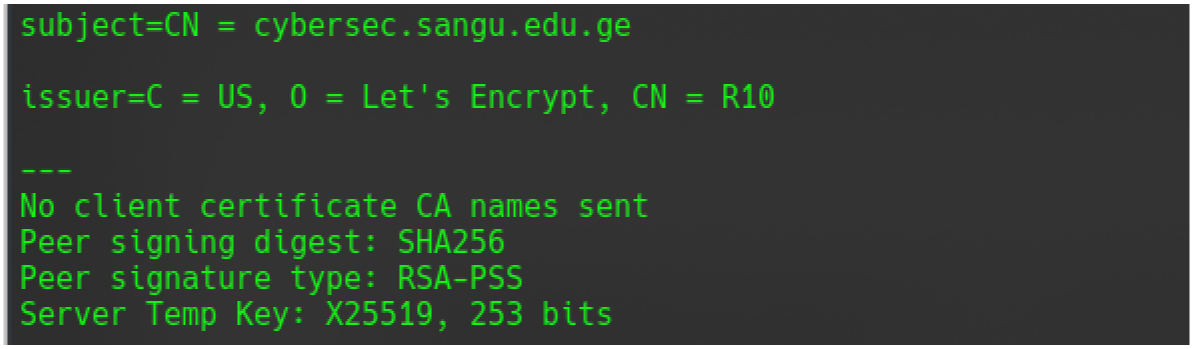
2) Server certificate is between

-----BEGIN CERTIFICATE----- and

-----END CERTIFICATE-----

```
MIIFLDCCBBSgAwIBAgISBc0ZD5S4s3v0XCol6+fyppdmMA0GCSqGSIb3DQEBCwUA

MDMxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQwwCgYDVQQD

EwNSMTAwHhcNMjUwMzI5MjM0NDIxWhcNMjUwNjI3MjM0NDIwWjAgMR4wHAYDVQQD

ExVjeWJlcnNlYy5zYW5ndS5lZHUuZ2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw

ggEKAoIBAQCg9THSlmkOjGsHeBHe/mN9R0eihpbcVFBWcL1nycOPSk8OuP5A2rXY

ivkYTYMU79IsQgjOnDUswsz8XeKuCvAKZyYIGgtIaOM6k0LXyhR0/s/kYPNKydBF

//f45m4erzMoIyLKEBP4rKt/mtV/T8PMw4u+OsJV/SEe/kPL0DEkxgMVFp/ZqGMu

hR8JWYCEKh2r7IT55BrCtibsaYzOE15pBcuyvbbSDHNdulglBeMNUGQen3WMo5aE

kfJHXhemKoHPRAPQ7PfFYyWTV/hXCSm+XQx60fzVZW8iDAqBUN+csM7H683IGu9V

DNRhD+qY9+R7rFqAzzabENhvmb9VL1iDAgMBAAGjggJLMIICRzAOBgNVHQ8BAf8E

BAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMAwGA1UdEwEB/wQC

MAAwHQYDVR0OBBYEFEF2OOGExMzQ+qadnAG3vYhQLqVVMB8GA1UdIwQYMBaAFLu8

w0el5LypxsOkcgwQjaI14cjoMFcGCCsGAQUFBwEBBEswSTAiBggrBgEFBQcwAYYW

aHR0cDovL3IxMC5vLmxlbmNyLm9yZzAjBggrBgEFBQcwAoYXaHR0cDovL3IxMC5p

LmxlbmNyLm9yZy8wIAYDVR0RBBkwF4IVY3liZXJzZWMuc2FuZ3UuZWR1LmdlMBMG

A1UdIAQMMAowCAYGZ4EMAQIBMC8GA1UdHwQoMCYwJKAioCCGHmh0dHA6Ly9yMTAu

Yy5sZW5jci5vcmcvMTAzLmNybDCCAQUGCisGAQQB1nkCBAIEgfYEgfMA8QB2AMz7

D2qFcQll/pWbU87psnwi6YVcDZeNtql+VMD+TA2wAAABleSAWccAAAQDAEcwRQIh

ALBz7/Y+YdxmsWpqRidf5DmqR1y2knMvR8QO8r2X9b9IAiAaADFE7I4h2D9Mdkjd

9CqkP8rj3N+3IR4HRmmFy4ONZgB3AM8RVu7VLnyv84db2Wkum+kacWdKsBfsrAHS

W3fOzDsIAAABleSAWfYAAAQDAEgwRgIhAMwbJRjYD2KL8fDqgg1znqo9/edhSfR2

ndFT70ji8pn2AiEAoWyu+LvtBBt0HVAcotndwEESi/OYBy39QTr6g8/zrf0wDQYJ

KoZIhvcNAQELBQADggEBAKWuRK17fPqy6f0UKNHoEjRkjtXFujXEFhbHZEVlxQ1+

xfB+zIaxfzSU51MOQCwpBKVVCdfVkvvLN/YB7s6SzoR5NBQbhZyMMzamjBH5b84Q

or3PS2nHt1s+huvIBODyGRyugCEnUhHPl2RaBHYRmytslWO/Z6gHqLAul1GVz1jS

aHPQdV4r8hLs5BmXlcGF4r6sAQ9Bu+438g+/E6Xz8Byf+93doOxrlDwwLYVqnNSX

01njQudSLnV3HhSbS7zrPLo1EDyRBUL9pr0PhGlu5div/RsFzJ/6nGv/CoSAuK64

EOqKE96KR4Dfjh6Qa+0LhktbC6MVoblV7Kqcko/Cvdw=
```

Certificate issuer is: Let's Encrypt

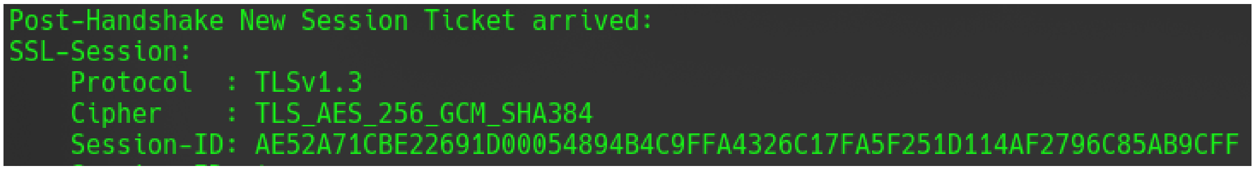


```
subject=CN = cybersec.sangu.edu.ge

issuer=C = US, O = Let's Encrypt, CN = R10

---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
```

Cipher suite used:

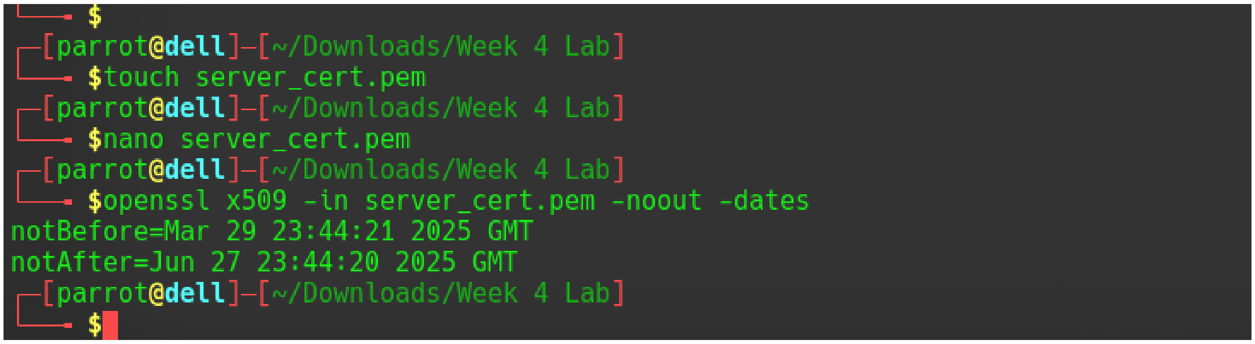
Protocol : TLSv1.3

Cipher   : TLS_AES_256_GCM_SHA384



```
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: AE52A71CBE22691D00054894B4C9FFA4326C17FA5F251D114AF2796C85AB9CFF
```

Certificate is valid until Jun 27 23:44:20 2025 GMT:



```
└──╼ $
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $touch server_cert.pem
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $nano server_cert.pem
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $openssl x509 -in server_cert.pem -noout -dates
notBefore=Mar 29 23:44:21 2025 GMT
notAfter=Jun 27 23:44:20 2025 GMT
┌─[parrot@dell]─[~/Downloads/Week 4 Lab]
└──╼ $
```