Applied Cryptography (UE20CS314)

Lab 4

Name: Vishwa Mehul Mehta

SRN: PES2UG20CS389

Section: F

Task 1:

Screenshot:

```
[10/25/22]seed@VM:~/.../lab4$ vim task1.c
[10/25/22]seed@VM:~/.../lab4$ gcc task1.c -o task1 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task1
a * b =  BD442D07C45A6708B52C365E89CA2D469EC29191E7E576B76D26264944DE293869B488E
969F7B63178471CBB4098DD3C
a^b mod n= 55E47FE848E4B4B83A608CE3BDB143119659D893BABD73406300A62B47B6B4C5
[10/25/22]seed@VM:~/.../lab4$
```

Observation:

We use the <openssl/bn.h> library for BIGNUM calculations. Here, we see the multiplication and modular exponentiation functions.

Task 2:

Screenshot:

```
[10/25/22]seed@VM:~/.../lab4$ vim task2.c
[10/25/22]seed@VM:~/.../lab4$ gcc task2.c -o task2 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task2
d =  3587A24598E5F2A21DB007D89D18CC50ABA5075BA19A33890FE7C28A9B496AEB
[10/25/22]seed@VM:~/.../lab4$
```

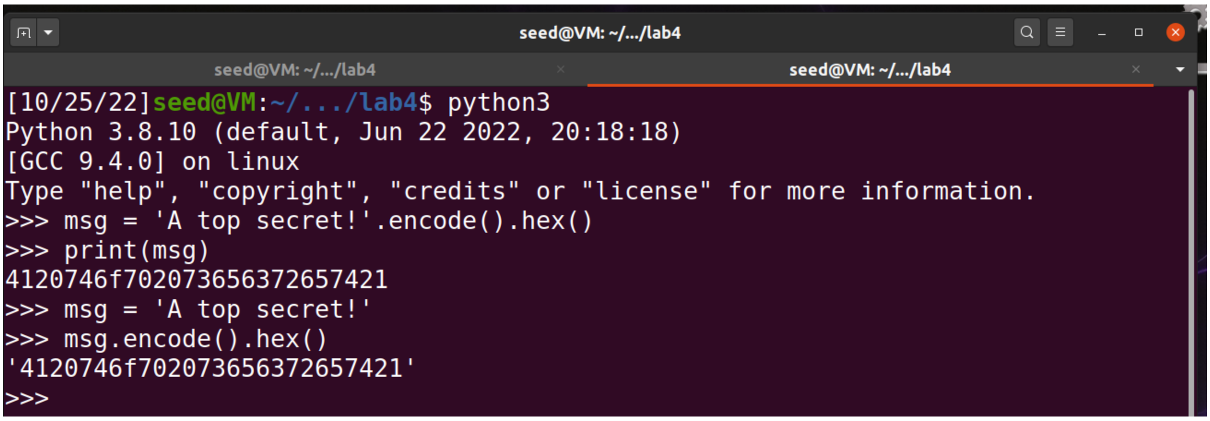
Observation:

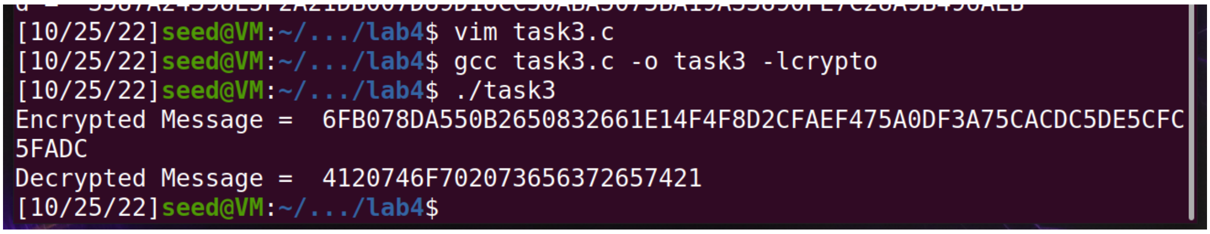
We now calculate the private key using the same library functions given 2 hexadecimal prime numbers p, q, and the public key (e, n)

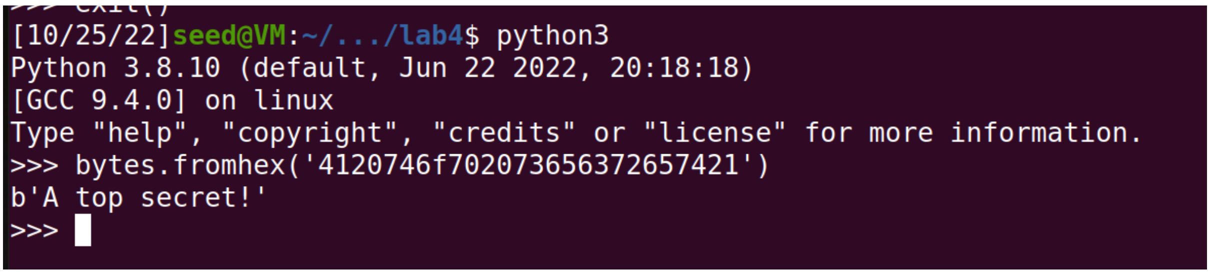
The **res1** variable takes the value **p-1** and **res2** takes the value **q-1**. These values are used to calculate f(n) (Totient Function) which is **res3 = res1 * res2.** Then we find the value of private key **d = e^{-1} mod res3.**

Task 3:

Screenshot:







Observation:

Task 1 is generating the hex code for the secret message that is used as the message in the encryption and decryption for Task 2.

In Task 2, we encrypt using the formula **enc = m^e mod n**, where **m** is the message, e and n are part of the public key. Decryption is done using the formula **dec = enc^d mod n**, where d is the private key and enc is the encrypted message.

Task 4:

Screenshot:

```
[10/25/22]seed@VM:~/.../lab4$ vim task4.c
[10/25/22]seed@VM:~/.../lab4$ gcc task4.c -o task4 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task4
Decrypted Message =  50617373776F72642069732064656573
[10/25/22]seed@VM:~/.../lab4$
```

```
[10/25/22]seed@VM:~/.../lab4$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bytes.fromhex('50617373776F72642069732064656573')
b'Password is dees'
>>>
```

Observation:

We decrypt a cipher text **c** and find the message using the decryption half of the code in Task 3.

Task 5:

Screenshot:

```
>>> msg = 'I owe you $2000'
>>> msg.encode().hex()
'49206f776520796f75202432303030'
>>>
```

```
[10/25/22]seed@VM:~/.../lab4$ vim task5.c
[10/25/22]seed@VM:~/.../lab4$ gcc task5.c -o task5 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task5
Sign =  80A55421D72345AC199836F60D51DC9594E2BDB4AE20C804823FB71660DE7B82
[10/25/22]seed@VM:~/.../lab4$
```

```
>>> msg = 'I owe you $3000'
>>> msg.encode().hex()
'49206f776520796f75202433303030'
>>>
```

```
[10/25/22]seed@VM:~/.../lab4$ vim task5.c
[10/25/22]seed@VM:~/.../lab4$ gcc task5.c -o task5 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task5
Sign =  04FC9C53ED7BBE4ED4BE2C24B0BDF7184B96290B4ED4E3959F58E94B1ECEA2EB
[10/25/22]seed@VM:~/.../lab4$
```

Observation:

We sign a message using the formula **sign = m^d mod n**, which gives a unique value for every message, a change in a single bit causes a change in multiple bits in the sign, as seen in the example where we change 2000 to 3000 and the signs are completely different.

Task 6:

Screenshot:

```
[10/25/22]seed@VM:~/.../lab4$ vim task6.c
[10/25/22]seed@VM:~/.../lab4$ gcc task6.c -o task6 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task6
Message =  4C61756E63682061206D697373696C652E
[10/25/22]seed@VM:~/.../lab4$
```

```
>>> bytes.fromhex('4C61756E63682061206D697373696C652E')
b'Launch a missile.'
>>>
```

Observation:

We now verify the sign by getting the message using the formula **message = s^e mod n**, and convert to byte form from hex.

Task 7:

Screenshot:

```
[10/25/22]seed@VM:~/.../lab4$ cat c1.pem
-----BEGIN CERTIFICATE-----
```

```
MIIHRzCCBi+gAwIBAgIQD6pjEJMHvD1BSJJkDM1NmjANBgkqhkiG9w0BAQsFADBP
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMSkwJwYDVQQDEyBE
aWdpQ2VydCBUTFMgUlNBIFNIQTI1NiAyMDIwIENBMTAeFw0yMjAzMTQwMDAwMDBa
Fw0yMzAzMTQyMzU5NTlaMIGWMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZv
cm5pYTEUMBIGA1UEBxMLTG9zIEFuZ2VsZXMxQjBABgNVBAoMOUludGVybmV0wqBD
b3Jwb3JhdGlvbsKgZm9ywqBBc3NpZ25lZMKgTmFtZXPCoGFuZMKgTnVtYmVyczEY
MBYGA1UEAxMPd3d3LmV4YW1wbGUub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAlV2WY5rlGn1fpwvuBhj0nVBcNxCxkHUG/pJG4HvaJen7YIZ1mLc7
/P4snOJZiEfwWFTikHNbcUCcYiKG8JkFebZOYMc1U9PiEtVWGU4kuYuxiXpD8oMP
in1B0SgrF7gKfO1//I2weJdAUjgZuXBCPAlhz2EnHddzXUtwm9XuOLO/Y6LATVMs
bp8/lXnfo/bX0UgJ7C0aVqOu07A0Vr6OkPxwWmOvF3cRKhVCM7U4B51KK+IsWRLm
8cVW1IaXjwhGzW7BR6EI3sxCQ4Wnc6HVPSgmomLWWWkIGFPAwcWUB4NC12yhCO5i
W/dxNMWNLMRVtnZAyq6FpZ8wFK6j4OMwMwIDAQABo4ID1TCCA9EwHwYDVR0jBBgw
FoAUt2ui6qiqhIx56rTaD5iyxZV2ufQwHQYDVR0OBBYEFPcqCdAkWxFx7rq+9D4c
PVYSiBa7MIGBBgNVHREEejB4gg93d3cuZXhhbXBsZS5vcmeCC2V4YW1wbGUubmV0
ggtleGFtcGxlLmVkdYILZXhhbXBsZS5jb22CC2V4YW1wbGUub3Jngg93d3cuZXhh
bXBsZS5jb22CD3d3dy5leGFtcGxlLmVkdYIPd3d3LmV4YW1wbGUubmV0MA4GA1Ud
DwEB/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwgY8GA1Ud
HwSBhzCBhDBAoD6gPIY6aHR0cDovL2NybDMuZGlnaWNlcnQuY29tL0RpZ2lDZXJ0
VExTUlNBU0hBMjU2MjAyMENBMS00LmNybDBAoD6gPIY6aHR0cDovL2NybDQuZGln
aWNlcnQuY29tL0RpZ2lDZXJ0VExTUlNBU0hBMjU2MjAyMENBMS00LmNybDA+BgNV
```

```
[10/25/22]seed@VM:~/.../lab4$ cat c0.pem
-----BEGIN CERTIFICATE-----
```

```
MIIEvjCCA6agAwIBAgIQBtjZBNVYQ0b2ii+nVCJ+xDANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0yMTA0MTQwMDAwMDBaFw0zMTA0MTMyMzU5NTlaME8xCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxKTAnBgNVBAMTIERpZ2lDZXJ0IFRMUyBS
U0EgU0hBMjU2IDIwMjAgQ0ExMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAwUuzZUdwvN1PWNvsnO3DZuUfMRNUrUpmRh8sCuxkB+Uu3Ny5CiDt3+PE0J6a
qXodgojlEVbbHp9YwlHnLDQNLtKS4VbL8Xlfs7uHyiUDe5pSQWYQYE9XE0nw6Ddn
g9/n00tnTCJRpt8OmRDtV1F0JuJ9x8piLhMbfyOIJVNvwTRYAIuE//i+p1hJInuW
raKImxW8oHzf6VGo1bDtN+I2tIJLYrVJmuzHZ9bjPvXj1hJeRPG/cUJ9WIQDgLGB
Afr5yjK7tI4nhyfFK3TUqNaX3sNk+crOU6JWvHgXjkkDKa77SU+kFbnO8lwZV21r
eacroicgE7XQPUDTITAHk+qZ9QIDAQABo4IBgjCCAX4wEgYDVR0TAQH/BAgwBgEB
/wIBADAdBgNVHQ4EFgQUt2ui6qiqhIx56rTaD5iyxZV2ufQwHwYDVR0jBBgwFoAU
A95QNVbRTLtm8KPiGxvDl7I90VUwDgYDVR0PAQH/BAQDAgGGMB0GA1UdJQQWMBQG
CCsGAQUFBwMBBggrBgEFBQcDAjB2BggrBgEFBQcBAQRqMGgwJAYIKwYBBQUHMAGG
GGh0dHA6Ly9vY3NwLmRpZ2ljZXJ0LmNvbTBABggrBgEFBQcwAoY0aHR0cDovL2Nh
Y2VydHMuZGlnaWNlcnQuY29tL0RpZ2lDZXJ0R2xvYmFsUm9vdENBLmNydDBCBgNV
HR8EOzA5MDegNaAzhjFodHRwOi8vY3JsMy5kaWdpY2VydC5jb20vRGlnaUNlcnRH
bG9iYWxSb290Q0EuY3JsMD0GA1UdIAQ2MDQwCwYJYIZIAYb9bAIBMAcGBWeBDAEB
MAgGBmeBDAECATAIBgZngQwBAgIwCAYGZ4EMAQIDMA0GCSqGSIb3DQEBCwUAA4IB
AQCAMs5eC91uWg0Kr+HWhMvAjvqFcO3aXbMM9yt1QP6FCvrzMXi3cEsaiVi6gL3z
ax3pfs8LulicWdSQ0/1s/dCYbbdxglvPbQtaCdB73sRD2Cqk3p5BJl+7j5nL3a7h
```

```
[10/25/22]seed@VM:~/.../lab4$ vim c1.pem
[10/25/22]seed@VM:~/.../lab4$ vim c0.pem
[10/25/22]seed@VM:~/.../lab4$ openssl x509 -in c1.pem -noout -modulus
Modulus=955D96639AE51A7D5FA70BEE0618F49D505C3710B1907506FE9246E07BDA25E9FB608675
98B73BFCFE2C9CE2598847F05854E290735B71409C622286F0990579B64E60C73553D3E212D55619
4E24B98BB1897A43F2830F8A7D41D1282B17B80A7CED7FFC8DB0789740523819B970423C0961CF61
271DD7735D4B709BD5EE38B3BF63A2C04D532C6E9F3F9579DFA3F6D7D14809EC2D1A56A3AED3B034
56BE8E90FC705A63AF1777112A154233B538079D4A2BE22C5912E6F1C556D486978F0846CD6EC147
A108DECC424385A773A1D53D2826A262D65969081853C0C1C594078342D76CA108EE625BF77134C5
8D2CC455B67640CAAE85A59F3014AEA3E0E33033
[10/25/22]seed@VM:~/.../lab4$ openssl x509 -in c1.pem -text -noout |grep "Expone
nt"
                Exponent: 65537 (0x10001)
[10/25/22]seed@VM:~/.../lab4$
```

```
[10/25/22]seed@VM:~/.../lab4$ openssl x509 -in c0.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            06:d8:d9:04:d5:58:43:46:f6:8a:2f:a7:54:22:7e:c4
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert G
lobal Root CA
        Validity
            Not Before: Apr 14 00:00:00 2021 GMT
            Not After : Apr 13 23:59:59 2031 GMT
        Subject: C = US, O = DigiCert Inc, CN = DigiCert TLS RSA SHA256 2020 CA1
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:c1:4b:b3:65:47:70:bc:dd:4f:58:db:ec:9c:ed:
                    c3:66:e5:1f:31:13:54:ad:4a:66:46:1f:2c:0a:ec:
```

```
[10/25/22]seed@VM:~/.../lab4$ cat signature.txt
80:32:ce:5e:0b:dd:6e:5a:0d:0a:af:e1:d6:84:cb:c0:8e:fa:
         85:70:ed:da:5d:b3:0c:f7:2b:75:40:fe:85:0a:fa:f3:31:78:
         b7:70:4b:1a:89:58:ba:80:bd:f3:6b:1d:e9:7e:cf:0b:ba:58:
         9c:59:d4:90:d3:fd:6c:fd:d0:98:6d:b7:71:82:5b:cf:6d:0b:
         5a:09:d0:7b:de:c4:43:d8:2a:a4:de:9e:41:26:5f:bb:8f:99:
         cb:dd:ae:e1:a8:6f:9f:87:fe:74:b7:1f:1b:20:ab:b1:4f:c6:
         f5:67:5d:5d:9b:3c:e9:ff:69:f7:61:6c:d6:d9:f3:fd:36:c6:
         ab:03:88:76:d2:4b:2e:75:86:e3:fc:d8:55:7d:26:c2:11:77:
         df:3e:02:b6:7c:f3:ab:7b:7a:86:36:6f:b8:f7:d8:93:71:cf:
         86:df:73:30:fa:7b:ab:ed:2a:59:c8:42:84:3b:11:17:1a:52:
         f3:c9:0e:14:7d:a2:5b:72:67:ba:71:ed:57:47:66:c5:b8:02:
         4a:65:34:5e:8b:d0:2a:3c:20:9c:51:99:4c:e7:52:9e:f7:6b:
         11:2b:0d:92:7e:1d:e8:8a:eb:36:16:43:87:ea:2a:63:bf:75:
         3f:eb:de:c4:03:bb:0a:3c:f7:30:ef:eb:af:4c:fc:8b:36:10:
         73:3e:f3:a4
[10/25/22]seed@VM:~/.../lab4$
```

```
cat: signature: No such file or directory
[10/25/22]seed@VM:~/.../lab4$ cat signature.txt | tr -d '[:space:]:'
8032ce5e0bdd6e5a0d0aafe1d684cbc08efa8570edda5db30cf72b7540fe850afaf33178b7704b1a
8958ba80bdf36b1de97ecf0bba589c59d490d3fd6cfdd0986db771825bcf6d0b5a09d07bdec443d8
2aa4de9e41265fbb8f99cbddaee1a86f9f87fe74b71f1b20abb14fc6f5675d5d9b3ce9ff69f7616c
d6d9f3fd36c6ab038876d24b2e7586e3fcd8557d26c21177df3e02b67cf3ab7b7a86366fb8f7d893
71cf86df7330fa7babed2a59c842843b11171a52f3c90e147da25b7267ba71ed574766c5b8024a65
345e8bd02a3c209c51994ce7529ef76b112b0d927e1de88aeb36164387ea2a63bf753febdec403bb
0a3cf730efebaf4cfc8b3610733ef3a4[10/25/22]seed@VM:~/.../lab4$
```

```
[10/25/22]seed@VM:~/.../lab4$ vim task7.c
[10/25/22]seed@VM:~/.../lab4$ gcc task7.c -o task7 -lcrypto
[10/25/22]seed@VM:~/.../lab4$ ./task7
Message =  48BD70DEDF511CD84432604CF75529666F001CDBBA36CD067797691DAB3667055424F
7632C1E2A8F3E3E366CAFF2F5DC69B54686A580B9C838259A77EF6EAA9924AF3D098FAEF863B3940
6BF306E76D9861BFD3D6EF0E416943D62956D049405C7C35363FA2E8F403850715B098BDDC28DDA2
5716D7204F7E21E3AA2C3EBFEEC1FD1A386F08EC93E8CC035CB701181895E65757ED26D3F793AC80
2104FF0A5B3A5E1C5358A8731A9D4999A585C38EA66B6E556899E3BED18E611A8EE546E8CB96D048
38C6BB0547E5589C00A4DD10F4E1135524DC7CD75C7357FC3D4A36DC3628902F93A0010B805DFEC9
AD65263542070175A6757CF9D0611AE1A0A6AF0B6C7
```

Observation:

We verify a certificate manually by first getting the user and the server certificates first along with the **modulus, exponent, and signature** to get back a message.