Name: Rithik Sarvesh Bharathiraja

UID: 120395246

Course: ENPM693

Task: Homework - 4

```
alex@Alex-PC:~$ openssl pkey -in privatekey.pem -text
Enter pass phrase for privatekey.pem:
-----BEGIN PRIVATE KEY-----
MIG2AgEAMBAGByqGSM49AgEGBSuBBAAiBIGeMIGbAgEBBDB8NiIpiYidZUUcJ7kZ
4ZSFiVZ+/Ri7HwIuYFkuumstLWW84FUrOES3UUXjCWd0smehZANiAARhuxy5vyqc
byPFQcB452ITPdujIQ8hAoFrbRfFM7sMV+dA0DtKeZGfJyijvFANgLuY86WWZzkF
T2c01XyuLzHElO3Mt/NIhWKM7pKDTorzi8zec/U5fKTYdu7A3sB+6K0=
-----END PRIVATE KEY-----
Private-Key: (384 bit)
priv:
    7c:36:22:29:89:88:9d:65:45:1c:27:b9:19:e1:94:
    85:89:56:7e:fd:18:bb:1f:02:2e:60:59:2e:ba:6b:
    2d:2d:65:bc:e0:55:2b:38:44:b7:51:45:e3:09:67:
    74:b2:67
pub:
    04:61:bb:1c:b9:bf:2a:9c:6f:23:c5:41:c0:78:e7:
    62:13:3d:db:a3:21:0f:21:02:81:6b:6d:17:c5:33:
    bb:0c:57:e7:40:d0:3b:4a:79:91:9f:27:28:a3:bc:
    50:0d:80:bb:98:f3:a5:96:67:39:05:4f:67:34:d5:
    7c:ae:2f:31:c4:94:ed:cc:b7:f3:48:85:62:8c:ee:
    92:83:4e:8a:f3:8b:cc:de:73:f5:39:7c:a4:d8:76:
    ee:c0:de:c0:7e:e8:ad
ASN1 OID: secp384r1
NIST CURVE: P-384
alex@Alex-PC:~$ echo "Rithik Sarvesh Bharathiraja"
Rithik Sarvesh Bharathiraja
alex@Alex-PC:~$ echo 120395246
120395246
```

1) The algorithm of the private key is Elliptic curve cryptography (SECP384R1)

2)

1. Key size : 384 bits
2. ASN1 OID: secp384r1
3. NIST CURVE: P-384
4. Curve type: 384-bit prime field Weierstrass curve
5. Also known as: P-384 ansip384r1

“openssl pkey -in privatekey.pem -text -noout” This command is used to derive the details of the private key.

openssl – That’s the main command

pkey – Used to access the private key

-in “file” – Input file to be accessed.

text – To derive the details of the key

noout – To stop the process from the printing the key again.

3) PKCS stands for Public-Key Cryptography Standards. PBKDF2 is a type of HMAC (hash-based message authentication code) in which input message/pass phrase is added along with the salt to undergo repeated process of encryption to derive the final cryptographic key. PBKDF2 with 1000 iteration was introduced in the year 2000 as it was found to be safe at that time. But, due to the present computational power available, it can be cracked instantly. In 2023, OWASP recommends to use 600,000 iteration for PBKDF2-HMAC-SHA256 and 210,000 for PBKDF2-HMAC-SHA512.