NSS Assignment

SAHIL BONDRE U18CO021

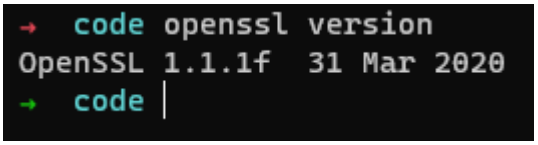
Solution

In this assignment, you have to demonstrate the working of Elliptic Curve key generation. You have to use the openssl library to demonstrate. In the end, show that the same key is shared between two users i.e. User A and User B. You have to generate the following files and upload the files on the assignment upload link:

1. Global Parameter file
2. Public key Private key files for User A and User B

You can use this resource to write this assignment or any material available on the Internet. Follow the below steps for writing this assignment.

1. Check if openssl is installed in your linux system or not. If not, go to this link to install openssl in your system.



2. OpenSSL provides two command line tools for working with keys suitable for Elliptic Curve (EC) algorithms:
    - openssl ecparam
    - openssl ec

```
→  code openssl ecparam -help
Usage: ecparam [options]
Valid options are:
 -help               Display this summary
 -inform PEM|DER     Input format - default PEM (DER or PEM)
 -outform PEM|DER    Output format - default PEM
 -in infile          Input file  - default stdin
 -out outfile        Output file - default stdout
 -text               Print the ec parameters in text form
 -C                  Print a 'C' function creating the parameters
 -check              Validate the ec parameters
 -list_curves        Prints a list of all curve 'short names'
 -no_seed            If 'explicit' parameters are chosen do not use the seed
 -noout              Do not print the ec parameter
 -name val           Use the ec parameters with specified 'short name'
 -conv_form val      Specifies the point conversion form
 -param_enc val      Specifies the way the ec parameters are encoded
 -genkey             Generate ec key
 -rand val           Load the file(s) into the random number generator
 -writerand outfile  Write random data to the specified file
 -engine val         Use engine, possibly a hardware device
→  code openssl ec -help
Usage: ec [options]
Valid options are:
 -help              Display this summary
 -in val            Input file
 -inform format     Input format - DER or PEM
 -out outfile       Output file
 -outform PEM|DER   Output format - DER or PEM
 -noout             Don't print key out
 -text              Print the key
 -param_out         Print the elliptic curve parameters
 -pubin             Expect a public key in input file
 -pubout            Output public key, not private
 -no_public         exclude public key from private key
 -check             check key consistency
 -passin val        Input file pass phrase source
 -passout val       Output file pass phrase source
 -param_enc val     Specifies the way the ec parameters are encoded
 -conv_form val     Specifies the point conversion form
 -*                 Any supported cipher
 -engine val        Use engine, possibly a hardware device
→  code
```

3. You can use secp256k1 curve for this demonstration. However, you can choose any of the curves of your choice. The list of the curves can be seen using ecparam command with list curves option

```
→  code openssl ecparam -list_curves
  secp112r1 : SECG/WTLS curve over a 112 bit prime field
  secp112r2 : SECG curve over a 112 bit prime field
  secp128r1 : SECG curve over a 128 bit prime field
  secp128r2 : SECG curve over a 128 bit prime field
  secp160k1 : SECG curve over a 160 bit prime field
  secp160r1 : SECG curve over a 160 bit prime field
  secp160r2 : SECG/WTLS curve over a 160 bit prime field
  secp192k1 : SECG curve over a 192 bit prime field
  secp224k1 : SECG curve over a 224 bit prime field
  secp224r1 : NIST/SECG curve over a 224 bit prime field
  secp256k1 : SECG curve over a 256 bit prime field
  secp384r1 : NIST/SECG curve over a 384 bit prime field
  secp521r1 : NIST/SECG curve over a 521 bit prime field
  prime192v1: NIST/X9.62/SECG curve over a 192 bit prime field
  prime192v2: X9.62 curve over a 192 bit prime field
  prime192v3: X9.62 curve over a 192 bit prime field
  prime239v1: X9.62 curve over a 239 bit prime field
  prime239v2: X9.62 curve over a 239 bit prime field
  prime239v3: X9.62 curve over a 239 bit prime field
  prime256v1: X9.62/SECG curve over a 256 bit prime field
  sect113r1 : SECG curve over a 113 bit binary field
  sect113r2 : SECG curve over a 113 bit binary field
  sect131r1 : SECG/WTLS curve over a 131 bit binary field
  sect131r2 : SECG curve over a 131 bit binary field
  sect163k1 : NIST/SECG/WTLS curve over a 163 bit binary field
  sect163r1 : SECG curve over a 163 bit binary field
  sect163r2 : NIST/SECG curve over a 163 bit binary field
  sect193r1 : SECG curve over a 193 bit binary field
  sect193r2 : SECG curve over a 193 bit binary field
  sect233k1 : NIST/SECG/WTLS curve over a 233 bit binary field
  sect233r1 : NIST/SECG/WTLS curve over a 233 bit binary field
  sect239k1 : SECG curve over a 239 bit binary field
  sect283k1 : NIST/SECG curve over a 283 bit binary field
  sect283r1 : NIST/SECG curve over a 283 bit binary field
  sect409k1 : NIST/SECG curve over a 409 bit binary field
  sect409r1 : NIST/SECG curve over a 409 bit binary field
  sect571k1 : NIST/SECG curve over a 571 bit binary field
  sect571r1 : NIST/SECG curve over a 571 bit binary field
```

4. Generate a Elliptic Curve Diffie-Hellman global domain parameters and save it in a file ECDHparam.pem

Use the following command:

openssl ecparam -name secp256k1 -out ECDHparam.pem

5. Display the generated global public parameters, using suitable command.

openssl ecparam -in ECDHparam.pem -text -param_enc explicit -noout

```
→  code openssl ecparam -in ECDHparam.pem -text -param_enc explicit -noout
Field Type: prime-field
Prime:
    00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
    ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:fe:ff:
    ff:fc:2f
A:    0
B:    7 (0x7)
Generator (uncompressed):
    04:79:be:66:7e:f9:dc:bb:ac:55:a0:62:95:ce:87:
    0b:07:02:9b:fc:db:2d:ce:28:d9:59:f2:81:5b:16:
    f8:17:98:48:3a:da:77:26:a3:c4:65:5d:a4:fb:fc:
    0e:11:08:a8:fd:17:b4:48:a6:85:54:19:9c:47:d0:
    8f:fb:10:d4:b8
Order:
    00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
    ff:fe:ba:ae:dc:e6:af:48:a0:3b:bf:d2:5e:8c:d0:
    36:41:41
Cofactor:  1 (0x1)
→  code cat ECDHparam.pem
-----BEGIN EC PARAMETERS-----
BgUrgQQACg==
-----END EC PARAMETERS-----
→  code
```

6. Generate the key using ecparam command with -genkey option and display the keys.

openssl ecparam -genkey -name prime256v1 -noout -out ec256-key-pair.pem

```
→  code openssl ecparam -genkey -name prime256v1 -noout -out ec256-key-pair.pem
→  code cat ec256-key-pair.pem
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIEB74EGyrTqwhGfCgDOjxjGVtlMdjcQihMt7sVp8voKLoAoGCCqGSM49
AwEHoUQDQgAE4Cwj9ft389SUxTAwC5/hq1hBrqMVyXvhP0yXJujcjQL/crAT7N8O
YDTq4fgCrfMjwVuRmaneXdcs3m80Z+QXIg==
-----END EC PRIVATE KEY-----
→  code
```