COM 5335 ASSIGNMENT #4

DUE BY 11:59PM 6/12/2021 (Sun)

10% penalty per day applies to 3-day late submissions. No submission will be accepted after 0:00 AM 6/16/2022.

Objective

Implement the General Elliptic Curve Group over prime fields GF(p) and use it to implement the EC-ElGamal cryptosystem.

Description

General elliptic curve group over a prime field GF(p) can be specified as E: $y^2=x^3+ax+b$ with point G. Let n=ord(G). The general elliptic curve group can be uniquely determined by the quintuple (p,a,b,G,n). In this assignment, we fix the following parameters.

```
p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFF
a = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFC
b = 1C97BEFC 54BD7A8B 65ACF89F 81D4D4AD C565FA45
Gx = 4A96B568 8EF57328 46646989 68C38BB9 13CBFC82
Gy = 23A62855 3168947D 59DCC912 04235137 7AC5FB32
n = 01000000 00000000 000001F4 C8F927AE D3CA7522 57
```

The objective of this assignment is to implement EC-ElGamal. Note that you need to represent the plaintext as a point on the curve and there is no guarantee that, given any x-coordinate, you can always find a y (as a solution) such that (x,y) is on the curve. This can be achieved by using **8 don't-care bits** in the x-coordinate, as shown in the Data Embedding Method below.

```
<Data Embedding Method>
Input: (m-8)-bit binary data M
Output: Point (Mx,My) on the elliptic curve
Mx = append(d,00)
while (Mx not on curve)
   increment Mx
compute y (s.t. y%2 == 1)
return (Mx,My)
```

You should look at the following two documents. sec1-v2.pdf and sec2-v2.pdf. Look at section 2.3 in sec1-v2.pdf to see how point at infinity is represented and how point compression is done. Look at sec2-v2.pdf for parameter samples.

3 Test Cases (Input shown in bold face)

```
<EC-ElGamal encryption>
Plaintext M = 110BA66C C954BE96 3A7831D9 D9A3D1D3 9B8EC3
Pa = 02 7AB13D6D 69847A9C CE9A84E5 DB1BDDD8 7F11F38C
nk = 8E07EB42 65F1200D 0745BCB3 E47EDD2D 23FBF571
Mx = 110BA66C C954BE96 3A7831D9 D9A3D1D3 9B8EC301
My = F4CBB301 B518D7D4 67E542D0 40AC6029 F7833135
Cm = {Pk,Pb} = {7AF4ED0D 220D9482 424E72FE 5A375C6B FC2B0743,
015A7D66 7CDA436F 401E6156 9109D753 ECD1F0B1}

<EC-ElGamal decryption>
Pk = 02 3D5A5C8A 80799494 624E741A 0119804F F707A2AB
Pb = 02 3C83F7C5 2185D5AC BE561718 80995F59 1DFE5C3C
na = 3C870C3E 99245E0D 1C06B747 DEB3124D C843BB8B
Plaintext = 2923BE84 E16CD6AE 529049F1 F1BBE9EB B3A6DB
```

```
<EC-ElGamal encryption>
Plaintext M = 8E6F2C1D C3987AFE CCC6F7DD FF75EDFC 324DF6
Pa = 03 9994C5C1 6070EE87 8F89A614 3CE865AC 2EC7EC5D
nk = 5487CF3D 6F9E4F1C 3DAEF5C3 CF7D6FC3 3C675DC6
Mx = 8E6F2C1D C3987AFE CCC6F7DD FF75EDFC 324DF600
My= 7BF6FA8B 834F99A6 9D7BA122 142DDE7A 8CF42B71
Cm = {Pk,Pb} = {EFE1AC15 1C68EDAF 3AA85E8D 5589FCE2 7D4C405B,
8970C8F5 C2BB301E 5EC4D31D DB225242 94FDACED}

<EC-ElGamal decryption>
Pk = 03 EFE1AC15 1C68EDAF 3AA85E8D 5589FCE2 7D4C405B
Pb = 03 8970C8F5 C2BB301E 5EC4D31D DB225242 94FDACED
na = 3C870C3E99245E0D1C06B747DEB3124DC843BB8B
Plaintext = 8E6F2C1D C3987AFE CCC6F7DD FF75EDFC 324DF6
```

```
<EC-ElGamal encryption>
Plaintext M = 668E9E1D 01A306A1 AB76C994 9A973248 E3AB53
Pa = 02 7AB13D6D 69847A9C CE9A84E5 DB1BDDD8 7F11F38C
nk = 8E07EB42 65F1200D 0745BCB3 E47ADD2D 23FBF573
Mx = 668E9E1D 01A306A1 AB76C994 9A973248 E3AB5300
My = 91811EB3 D1BD2F35 EC24FA10 D37312FB D6827971
Cm = {Pk,Pb} = {BDC5D14A 5BA16F67 87A050C6 CD2F4C4C 72AD2671,
A9FC4BBA 3F7B3D53 D3CEF8D 0D9F01658 82541CE2}

<EC-ElGamal decryption>
Pk = 03 BDC5D14A 5BA16F67 87A050C6 CD2F4C4C 72AD2671
Pb = 02 A9FC4BBA 3F7B3D53 D3CEF8D 0D9F01658 82541CE2
na = 246FF426 810C46F5 04EE9F2F C69BFA35 B02BA373
Plaintext = 668E9E1D 01A306A1 AB76C9949 A973248 E3AB53
```

Grading

Your program MUST BE compatible with Dev C/C++ or GNU C/C++ compilers. If you wish to use other compilers, please contact the TA. **You will get no points if your program cannot be compiled by the TA.** If your program is compilable but the result is not completely correct, you'll still get partial credits. Your program should be well-commented, well-structured, and easy to understand. You may lose up to 30% of points if you fail to do so.

Submission

Put all your source codes in a folder containing main functions, function implementations, class definitions, or compilation instructions (if any). Compress them as a single zip file. DO NOT submit executable files. Name your zip file as your student ID number (i.e. 100012345.zip). Submit your source code on eLearn.