Network Security Homework4

109064518 高聖哲(Kao Sheng-Che)

Develop Environment

Operation System	Ubuntu 16.04
Compiler	GNU C/C++

The architecture of EC-ElGamal encryption/decryption

Enter the following command to install gmp library in Linux terminal.

```
├── BigNumber           ## BigNumber Operation
├── FiniteFieldElement  ## FiniteFieldElement Operation based on BigNumber
├── Point               ## Point element of Elliptic Curve
└── EllipticCurve       ## Main Cryptography method
```

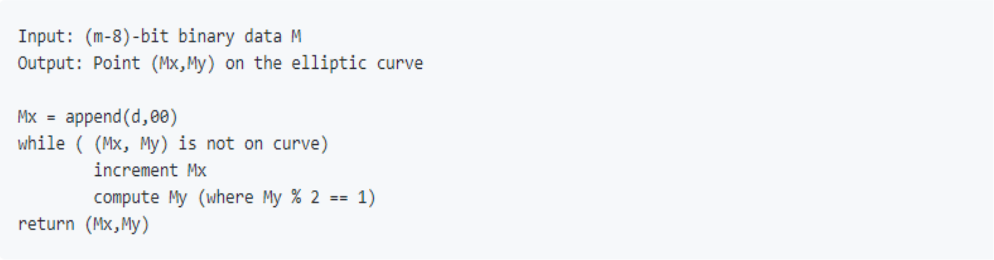
Compile and Run the Program(Makefile)

```
$ make
$ ./homework4.out
```

Elliptic Curve Cryptography Explanation

1. Step1:Data Embedded Method

- ◆ 當我們得到 plaintext 後，我們將其轉換成 Point Type，因此在以下動作我們將 plaintext 假設為 38-byte long,比我們的 Bignumber(40-Byte long)小一點



```
Input: (m-8)-bit binary data M
Output: Point (Mx,My) on the elliptic curve

Mx = append(d,00)
while ( (Mx, My) is not on curve)
        increment Mx
        compute My (where My % 2 == 1)
return (Mx,My)
```

2. Step2:Data Types and Conversions

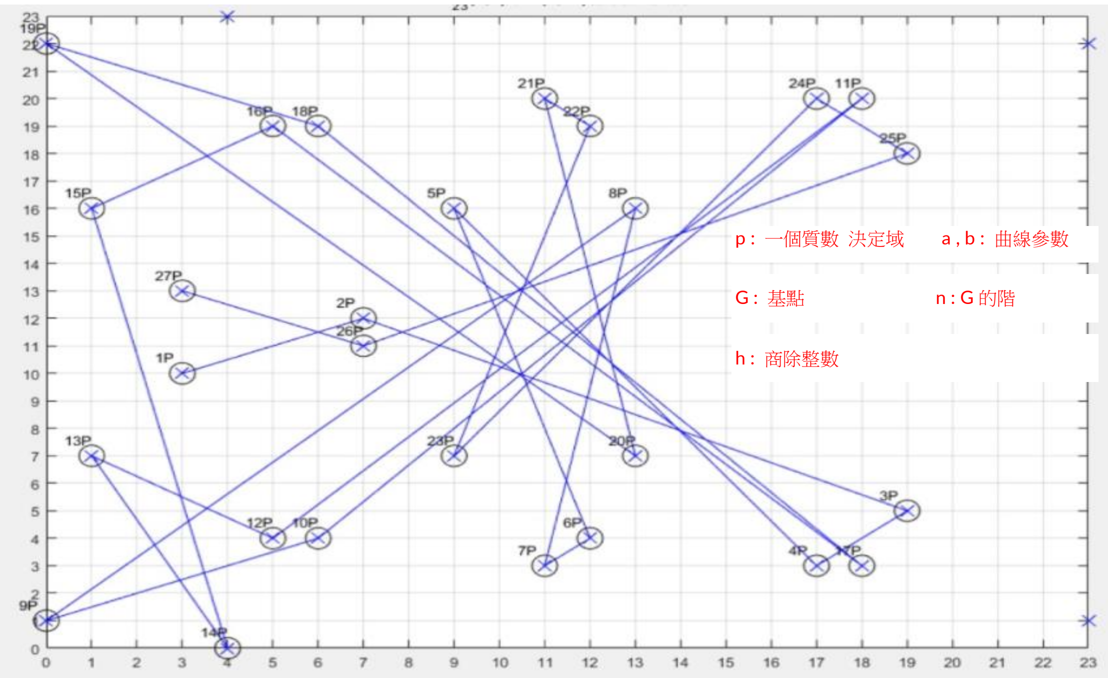
- ◆ Point Type 有時是多餘的，一旦我們擁有 x coordinate of a Point，同樣也能得到 y coordinate 在 Elliptic Curve Equation: y^2 = x^3 + ax + b

3. Step3:橢圓曲線計算流程

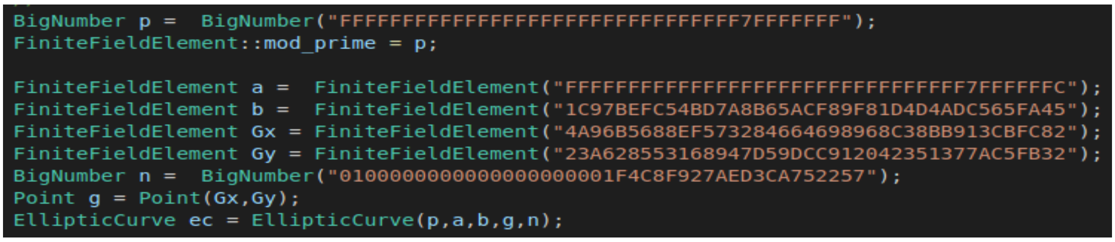
I. 設定出一個有限域 Fp，如果橢圓曲線上一點 P，存在最小的正整數 n 使得數乘 nP=O∞ ,則將 n 稱為 P 的階，若 n 不存在，則 P 是無限階的。

II. 因此選定 n 後即可 計算出可得 27P=-P，所以 28P=O ∞ P 的階為 28，這些點做成了一個循環阿貝爾群，其中生成元為 P，階數為 29 並從裡面選取基點，並開始計算。

III. 考慮 K=kG ，其中 K、G 為橢圓曲線 Ep(a,b)上的點，n 為 G 的階（nG=O∞），k 為小於 n 的整數。則給定 k 和 G，根據加法法則，計算 K 很容易，但反過來，給定 K 和 G，求 k 就非常困難，其中 k、K 分別為私鑰、公鑰。



IV. The quintuple (p,a,b,G,n) parameters of general elliptic curve group



```
BigNumber p =  BigNumber("FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7FFFFFFF");
FiniteFieldElement::mod_prime = p;

FiniteFieldElement a =  FiniteFieldElement("FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7FFFFFFC");
FiniteFieldElement b =  FiniteFieldElement("1C97BEFC54BD7A8B65ACF89F81D4D4ADC565FA45");
FiniteFieldElement Gx = FiniteFieldElement("4A96B5688EF573284664698968C38BB913CBFC82");
FiniteFieldElement Gy = FiniteFieldElement("23A628553168947D59DCC912042351377AC5FB32");
BigNumber n =  BigNumber("0100000000000000000001F4C8F927AED3CA752257");
Point g = Point(Gx,Gy);
EllipticCurve ec = EllipticCurve(p,a,b,g,n);
```

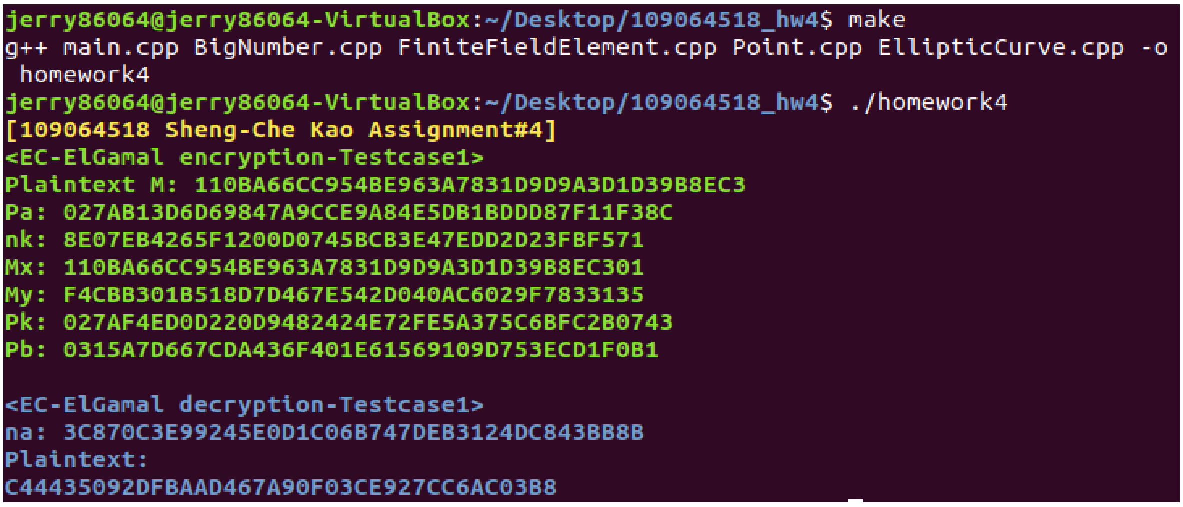
4. 橢圓曲線加解密演算法原理 ECIES

設私鑰、公鑰分別為 k、K，即 K = kG，其中 G 為 G 點。

I. 公鑰加密：選擇隨機數 r，將訊息 M 生成密文 C，該密文是一個點對，即：C = {rG, M rK}，其中 K 為公鑰

II. 私鑰解密：M rK – k(rG) = M r(kG) – k(rG) = M，其中 k、K 分別為私鑰、公鑰。

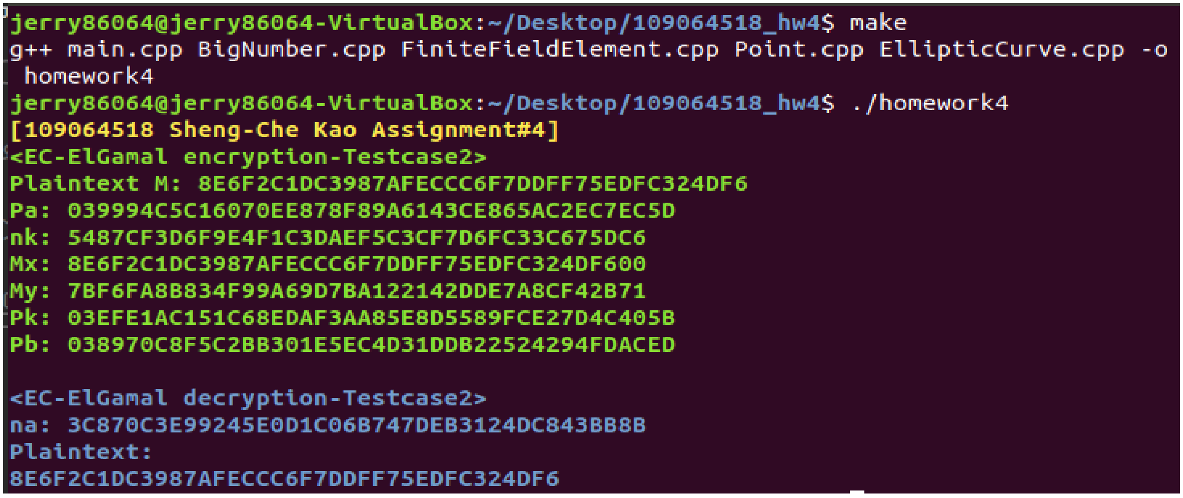
- The compile result of the Ubuntu terminal



```
jerry86064@jerry86064-VirtualBox:~/Desktop/109064518_hw4$ make
g++ main.cpp BigNumber.cpp FiniteFieldElement.cpp Point.cpp EllipticCurve.cpp -o
 homework4
jerry86064@jerry86064-VirtualBox:~/Desktop/109064518_hw4$ ./homework4
[109064518 Sheng-Che Kao Assignment#4]
<EC-ElGamal encryption-Testcase1>
Plaintext M: 110BA66CC954BE963A7831D9D9A3D1D39B8EC3
Pa: 027AB13D6D69847A9CCE9A84E5DB1BDDD87F11F38C
nk: 8E07EB4265F1200D0745BCB3E47EDD2D23FBF571
Mx: 110BA66CC954BE963A7831D9D9A3D1D39B8EC301
My: F4CBB301B518D7D467E542D040AC6029F7833135
Pk: 027AF4ED0D220D9482424E72FE5A375C6BFC2B0743
Pb: 0315A7D667CDA436F401E61569109D753ECD1F0B1

<EC-ElGamal decryption-Testcase1>
na: 3C870C3E99245E0D1C06B747DEB3124DC843BB8B
Plaintext:
C44435092DFBAAD467A90F03CE927CC6AC03B8
```

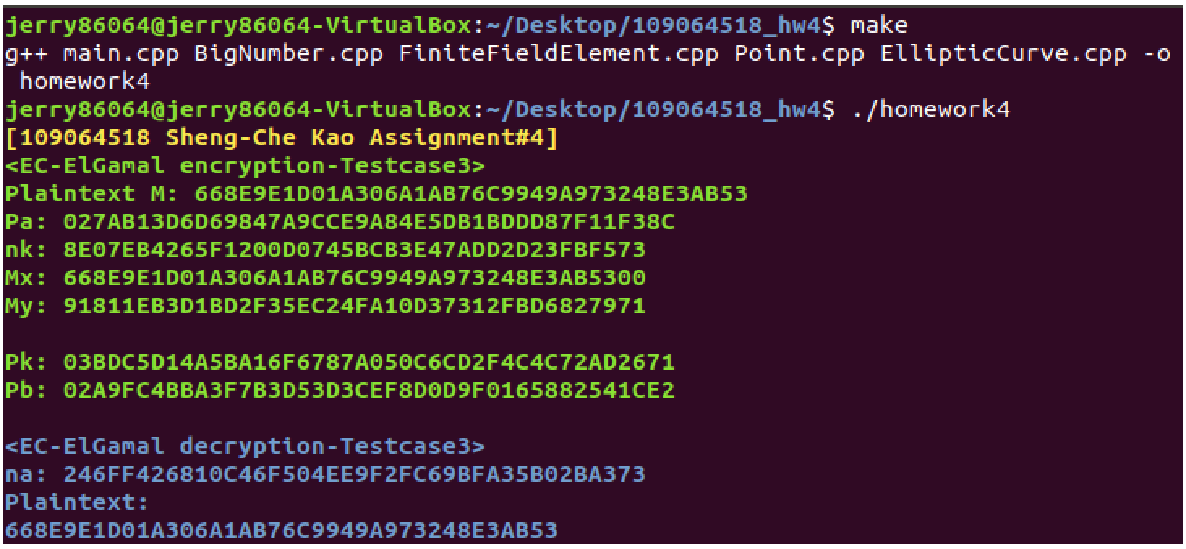
Fig.1 EC-ElGamal encryption/decryption testcase1



```
jerry86064@jerry86064-VirtualBox:~/Desktop/109064518_hw4$ make
g++ main.cpp BigNumber.cpp FiniteFieldElement.cpp Point.cpp EllipticCurve.cpp -o
 homework4
jerry86064@jerry86064-VirtualBox:~/Desktop/109064518_hw4$ ./homework4
[109064518 Sheng-Che Kao Assignment#4]
<EC-ElGamal encryption-Testcase2>
Plaintext M: 8E6F2C1DC3987AFECCC6F7DDFF75EDFC324DF6
Pa: 039994C5C16070EE878F89A6143CE865AC2EC7EC5D
nk: 5487CF3D6F9E4F1C3DAEF5C3CF7D6FC33C675DC6
Mx: 8E6F2C1DC3987AFECCC6F7DDFF75EDFC324DF600
My: 7BF6FA8B834F99A69D7BA122142DDE7A8CF42B71
Pk: 03EFE1AC151C68EDAF3AA85E8D5589FCE27D4C405B
Pb: 038970C8F5C2BB301E5EC4D31DDB22524294FDACED

<EC-ElGamal decryption-Testcase2>
na: 3C870C3E99245E0D1C06B747DEB3124DC843BB8B
Plaintext:
8E6F2C1DC3987AFECCC6F7DDFF75EDFC324DF6
```

Fig.2 EC-ElGamal encryption/decryption testcase2



```
jerry86064@jerry86064-VirtualBox:~/Desktop/109064518_hw4$ make
g++ main.cpp BigNumber.cpp FiniteFieldElement.cpp Point.cpp EllipticCurve.cpp -o
 homework4
jerry86064@jerry86064-VirtualBox:~/Desktop/109064518_hw4$ ./homework4
[109064518 Sheng-Che Kao Assignment#4]
<EC-ElGamal encryption-Testcase3>
Plaintext M: 668E9E1D01A306A1AB76C9949A973248E3AB53
Pa: 027AB13D6D69847A9CCE9A84E5DB1BDDD87F11F38C
nk: 8E07EB4265F1200D0745BCB3E47ADD2D23FBF573
Mx: 668E9E1D01A306A1AB76C9949A973248E3AB5300
My: 91811EB3D1BD2F35EC24FA10D37312FBD6827971

Pk: 03BDC5D14A5BA16F6787A050C6CD2F4C4C72AD2671
Pb: 02A9FC4BBA3F7B3D53D3CEF8D0D9F0165882541CE2

<EC-ElGamal decryption-Testcase3>
na: 246FF426810C46F504EE9F2FC69BFA35B02BA373
Plaintext:
668E9E1D01A306A1AB76C9949A973248E3AB53
```

Fig.3 EC-ElGamal encryption/decryption testcase3