ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

---------***---------

BÁO CÁO BÀI TẬP LAB1

CÔNG NGHỆ CHUỖI KHỐI

Giảng viên hướng dẫn : PGS.TS. Nguyễn Đình Thúc

Học viên thực hiện : Nguyễn Thiện Dương - 21C12004

Môn học : Công nghệ chuỗi khối

Khóa : 31

Ngành : Hệ thống thông tin

TP. HỒ CHÍ MINH, tháng 01 năm 2023

1. Hệ thống mật mã khóa công khai ECC

Mật mã đường cong Elliptic (ECC) là họ hiện đại của các hệ thống mật mã khóa công khai , dựa trên cấu trúc đại số của các đường cong elliptic trên các trường hữu hạn và độ khó của Bài toán logarit rời rạc trên đường cong Elliptic (ECDLP) .

ECC thực hiện tất cả các khả năng chính của hệ thống mật mã bất đối xứng: mã hóa , chữ ký và trao đổi khóa .

Mật mã ECC được coi là sự kế thừa tự nhiên hiện đại của hệ thống mật mã RSA , bởi vì ECC sử dụng các khóa và chữ ký nhỏ hơn RSA cho cùng một mức độ bảo mật và cung cấp khả năng tạo khóa rất nhanh , thỏa thuận khóa nhanh và chữ ký nhanh .

Mật mã học sử dụng các đường cong elip ở dạng đơn giản hóa (dạng Weierstras), được định nghĩa là : $y^2 = x^3 + ax + b$

Khóa riêng (Private Key): Việc tạo khóa riêng của mật mã ECC cũng đơn giản như việc tạo một số nguyên ngẫu nhiên một cách an toàn trong một phạm vi cụ thể, giúp quá trình này diễn ra rất nhanh chóng. Bất kỳ số nguyên nào trong trường đại diện cho khóa riêng ECC hợp lệ.

Khóa công khai (Public key): Khóa công khai trong ECC là các điểm EC, là các cặp tọa độ nguyên x và y nằm trên một đường cong. Do các tính năng độc đáo của nó, các điểm EC có thể được nén thành một tọa độ + 1 bit (lẻ hoặc chẵn). Kết quả là khóa công khai được nén tương ứng với ECC 256

2. Khóa riêng tư , khóa công khai và khởi tạo điểm trong ECC

Trong ECC , khi chúng ta nhân một điểm EC cố định G (điểm tạo) với số nguyên k nhất định (k có thể được coi là khóa riêng), chúng ta thu được điểm EC là P (khóa chung tương ứng của nó).

Do đó, trong ECC, ta có:

- Đường cong Eelliptic (EC) trên trường hữu hạn $\mathbb{F}p$
- G == điểm tạo (hằng số cố định, một điểm cơ sở trên EC)
- k == khóa riêng (số nguyên)
- P == khóa công khai (điểm)

3. Demo quá trình tạo khóa riêng và khóa công khai trong ECC

Ví dụ với đường cong ECC thỏa : $y^2 \equiv x^3 + 7 \ (mod\ 17)$ với điểm tạo G(15 ; 13) có bậc thứ tự n = 18. Gọi tên là đường cong **p1707**

Đoạn mã dưới đây minh họa **phép nhân EC** . Nó nhân điểm tạo **G** với 0, 1, 2, ..., 24.

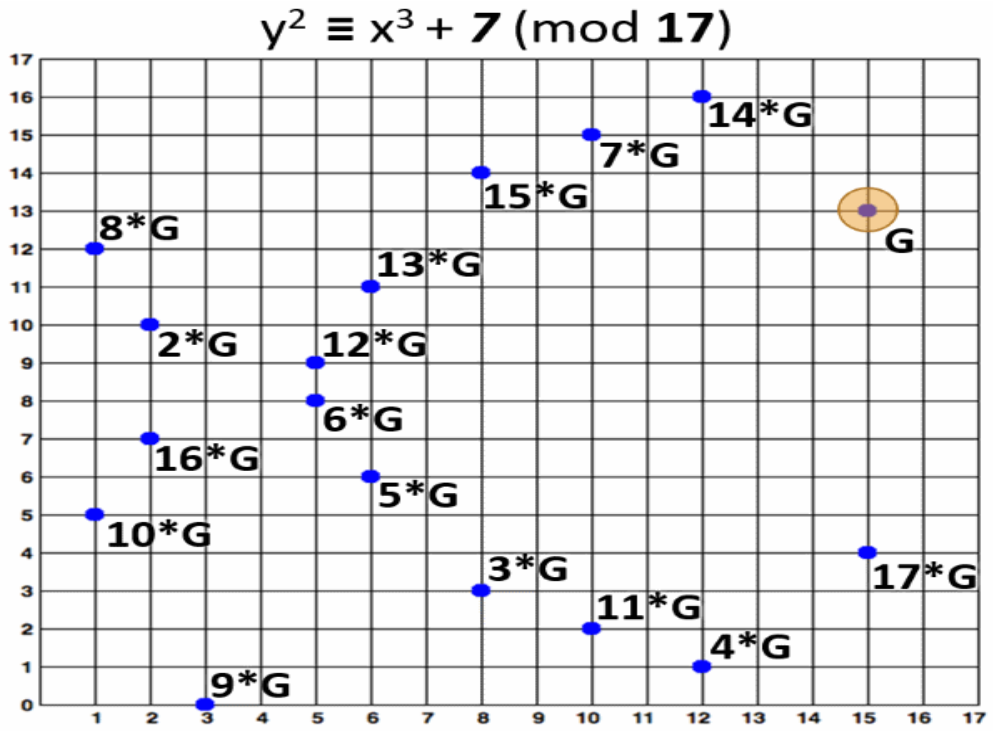
demo.py 5

```python
from tinyec.ec import SubGroup, Curve
from tinyec.ec import SubGroup, Curve
from tinyec import registry
from tinyec import registry
import secrets
from nummaster.basic import sqrtmod

field = SubGroup(p=17, g=(15, 13), n=18, h=1)
curve = Curve(a=0, b=7, field=field, name='p1707')
print('curve:', curve)

for k in range(0, 25):
    p = k * curve.g
    print(f"{k} * G = ({p.x}, {p.y})")

print("-----------------------------------")
```

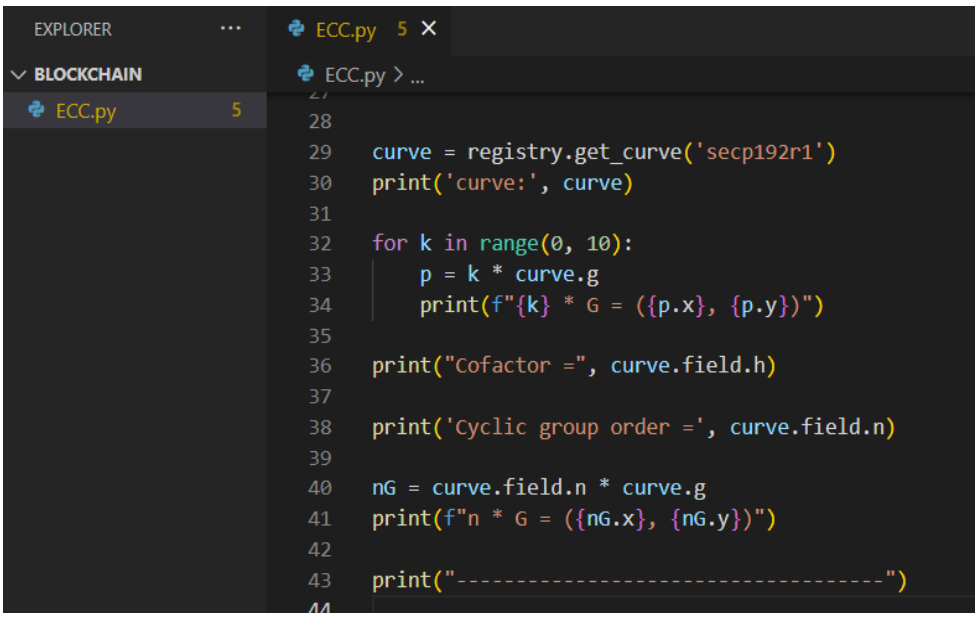
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
PS C:\Users\thien\OneDrive\Máy tính\Blockchain> py demo.py
curve: "p1707" => y^2 = x^3 + 0x + 7 (mod 17)
0 * G = (None, None)
1 * G = (15, 13)
2 * G = (2, 10)
3 * G = (8, 3)
4 * G = (12, 1)
5 * G = (6, 6)
6 * G = (5, 8)
7 * G = (10, 15)
8 * G = (1, 12)
9 * G = (3, 0)
10 * G = (1, 5)
11 * G = (10, 2)
12 * G = (5, 9)
13 * G = (6, 11)
14 * G = (12, 16)
15 * G = (8, 14)
16 * G = (2, 7)
17 * G = (15, 4)
18 * G = (None, None)
19 * G = (15, 13)
20 * G = (2, 10)
21 * G = (8, 3)
22 * G = (12, 1)
23 * G = (6, 6)
24 * G = (5, 8)
```

Các điểm EC, được tạo bằng cách nhân điểm tạo **G** với 2, 3, 4, ..., 17 được hiển thị trên hình bên dưới:



Giả sử ta sử dụng một đường cong khác chứa 192 bit có tên là **secp192r1 .** (192-bit, p = 6277101735386680763835789423207666416083908700 390324961279)



Kết quả đầu ra cũng tương tự như ví dụ trước:

```
curve: "secp192r1" => y^2 = x^3 + 6277101735386680763835789423207666416083908700390324961276x + 245515554600894381774029391519745178476910805816119123806 5 (mod 6277101735386680763835789423207666416083908700390324961279)
0 * G = (None, None)
1 * G = (602046282375688656758213480587526111916698976636884684818, 174050332293622031404857552280219410364023488927386650641)
2 * G = (5369744403678710563432458361254544170966096384586764429448, 5429234379789071039750654906915254128254326554272718558123)
3 * G = (2915109630280678890720206779706963455590627465886103135194, 2946626711558792003980654088990112021985937607003425539581)
4 * G = (1305994880430903997305943738697779408316929565234787837114, 3981863977451150342116987835776121688410789618551673306674)
5 * G = (4102832511167848740189935621365668701106767069367626602 40, 1206654674899825246688205669651974202006189255452737318561)
6 * G = (4008504146453526025173196900303594155799995627910231899946, 3263759301305176906990806636587838100022690095020155627760)
7 * G = (3473339081378406123852871299395262476289672479707038350589, 2152713176906603604200842901176476029776543437891569565621)
8 * G = (1167950611014894512313033362696697441497340081390841490910, 4002177906111215127148483369584652296488769677804145538752)
9 * G = (3176317450453705650283775811228493626776489433309636475023, 4460189377466938476679380385498011517961211807501706 2201)
Cofactor = 1
Cyclic group order = 6277101735386680763835789423176059013767194773182842284081
n * G = (None, None)
-----------------------------------
```

Tạo một khóa riêng ngẫu nhiên `privKey` (số nguyên trong phạm vi [0...n-1]) và khóa chung tương ứng của nó : `pubKey = privKey * G`

Kết quả tạo khóa PublicKey và PrivateKey như hình dưới với thời gian thực thị khoảng 0.04s.

```
-----------------------------------
private key: 1042153625119078852931123103993673019629999446592240033797
public key: (156732397188999796702038590490984158028509960938915992676 1, 1182591487081491758839459226001778188332069612380023485407) on "secp192r1" => y^2 = x^3 + 627710173538668076383578942320766641608390870039032496127 6x + 245515554600894381774029391519745178476910805816119123806 5 (mod 6277101735386680763835789423207666416083908700390324961279)
-----------------------------------
Duration: 0:00:00.040073
```

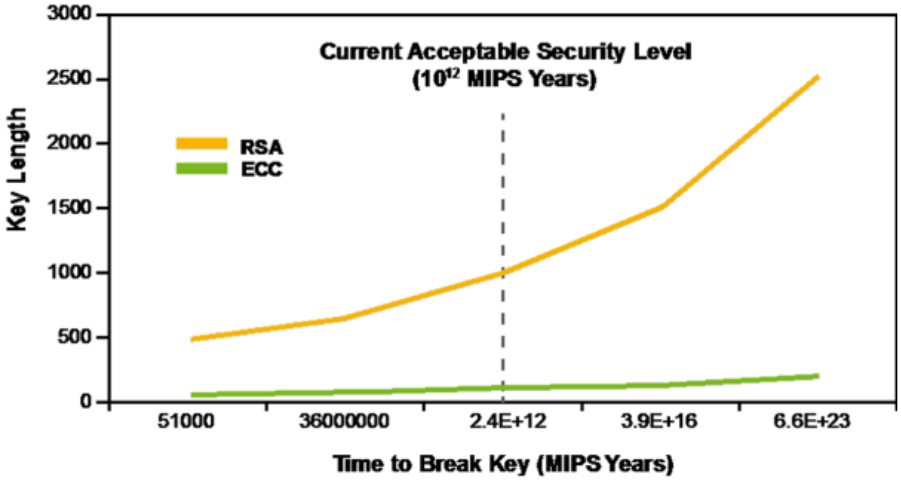
4. Đánh giá hệ thống mật mã hóa đường cong ECC

Ưu điểm :

- Tạo khóa nhanh: Quá trình tạo khóa của mật mã ECC cũng đơn giản như tạo một số nguyên ngẫu nhiên một cách an toàn trong một phạm vi cụ thể, khiến quá trình này diễn ra rất nhanh chóng.
- Kích thước khóa nhỏ hơn: So với mã hóa không phải EC (dựa trên các trường Galois thông thường), ECC cho phép ít khóa hơn để đảm bảo tính bảo mật như nhau .
- Độ trễ thấp : Bằng cách tính toán chữ ký trong hai giai đoạn, ECC đạt được độ trễ thấp hơn so với nghịch đảo trong suốt. ECC có các giao thức mạnh mẽ để trao đổi khóa được ủy quyền và công nghệ này đã được áp dụng rộng rãi.
- Tính bảo mật cao : ECC cung cấp khả năng bảo mật mạnh mẽ trong một thế giới mà điện thoại di động ngày càng phải thực hiện nhiều thao tác mã hóa hơn với ít tài nguyên máy tính hơn.

Nhược điểm : Kích thước mã hóa lớn , Thuật toán Phức tạp hơn

So sánh mức độ bảo mật theo thời gian do khóa tạo ra giữa RSA và ECC. Từ biểu đồ có thể thấy rằng khóa do ECC tạo ra có hiệu suất bảo mật tốt hơn.



Link github source code : https://github.com/DuongLuke/LAB1_ECC_Blockchain