Grenoble INP – Ensimag, UGA

M2 CySec – Advanced cryptography

# Elliptic curves lab session

*Contributors :*
Anthony Martinez

*Supervisors :*
Emmanuel Peyre

December 13, 2021

# Contents

# 1 Part 1 : function over E(F)

We consider the following equation for the curve E :

$$Y^2 = X^3 + aX + b$$

And F denote the field Z/pZ. For the compilation and execution I used python3.8.

## 1.1 Question 1

The function to cumpute the inverse element over P is **inverse_element**.

## 1.2 Question 2

The function to test if an element is on the curve is **is_on_curve**. We have defined the InfinityPoint by the point of coordinates (0,0).

## 1.3 Question 3

To compute the map P -> 2P we have to use the function **doubling**.

## 1.4 Question 4

To compute the addition over the elliptic curve we used **__add__**. For example if P and Q are two point the program instruction P + Q will compute the addition of the two points over E(F).

## 1.5 Question 5

To compute the multiplication over the elliptic curve we used **__rmul__**. For example if P is a point and n a scaler the program instruction n * P will compute the multiplication over E(F).

## 1.6 Quesiton 6

The small program that perform the Diffie-Hellman key exchange works as follow : With A and B the two party who want to exchange a secret :
Given two scalars a and b (choose randomly for A and B with a only known by A and b only known by B) and a public point G.
A send X = aG to B (over E(F))
B send Y = bG to A (over E(F))
A compute aX = secret_key
B compute aY = secret_key
The algorithm is compute by the function **diffie_hellman**.

# 2 Part 2

For this section we take the following value :
a = 0
b = 7
x = 0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798
y = 0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8
p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F

## 2.1 Quesiton 7

We compute the following point P = (x,y) and we see that is belong to the curve. Test with the assertion **assert G.is_on_curve() == True**.

## 2.2 Quesiton 8

With o = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141
We compute oP and we get the InfinityPoint.

## 2.3 Quesiton 9

We have used this website `https://www.dcode.fr/primality-test` to check if o is prime. And we get that o is prime.

## 2.4 Quesiton 10

With the last question we can conclude that the order of P is o. Because a point on the curve multiplied by the scalar o give the neutral point Infinity and o divide p-1.

## 2.5 Quesiton 11

For this question we have change the Diffie Helllman protocol to choose the secret keys of A and B randomly. The new function is **diffie_hellman_subgroup**.