Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»
Дисциплина «Криптография»

Лабораторная работа №3

Тема: Эллиптические кривые

Студент: Глушатов И.С.
Группа: М8О-307Б-19
Преподаватель: Борисов А. В.
Дата:
Оценка:

Москва, 2022

Цель работы: приобрести знания в области эллиптических кривых, написать программу, которая умеет складывать точки на эллиптических кривых с заданными коэффициентами над конечным полем F_p и определять порядок точек.

Задание:
Подобрать такую эллиптическую кривую, порядок точки которой полным перебором находится за 10 минут на ПК. Упомянуть в отчёте результаты замеров работы программы, характеристики вычислителя. Также указать какие алгоритмы и/или теоремы существуют для облегчения и ускорения решения задачи полного перебора. Рассмотреть для случая конечного простого поля Z_p

Оборудование: Домашний компьютер, процессор Intel® CoreTM i5-7200 CPU 3.40GHz 3.40 GHz, память 8ГБ, 64-разрядная система

Ход работы

Каноническая форма эллиптической кривой над конечным полем F_p:

$$y^2 \equiv x^3 + ax + b \ (mod\ p)$$
$$4a^3 + 27b^2 \not\equiv 0 \ (mod\ p)$$

Сложение точек:

$$P + Q + R = 0$$

Пусть $P = (x_1; y_1), Q = (x_2; y_2),\ R = (x_3; y_3), 0 = (0; 0)$

1. $P + 0 = 0 + P = P$
2. $P + (-P) = P - P = 0$, где $(-P) = (x_1; -y_1\ mod\ p)$
3. $P = Q$

$$m = (3{x_1}^2 + a)(2y_1)^{-1} \ (mod\ p)$$
$$x_3 = (m^2 - x_1 - x_2) \quad (mod\ p)$$
$$y_3 = y_1 + m(x_3 - x_1) \quad (mod\ p)$$

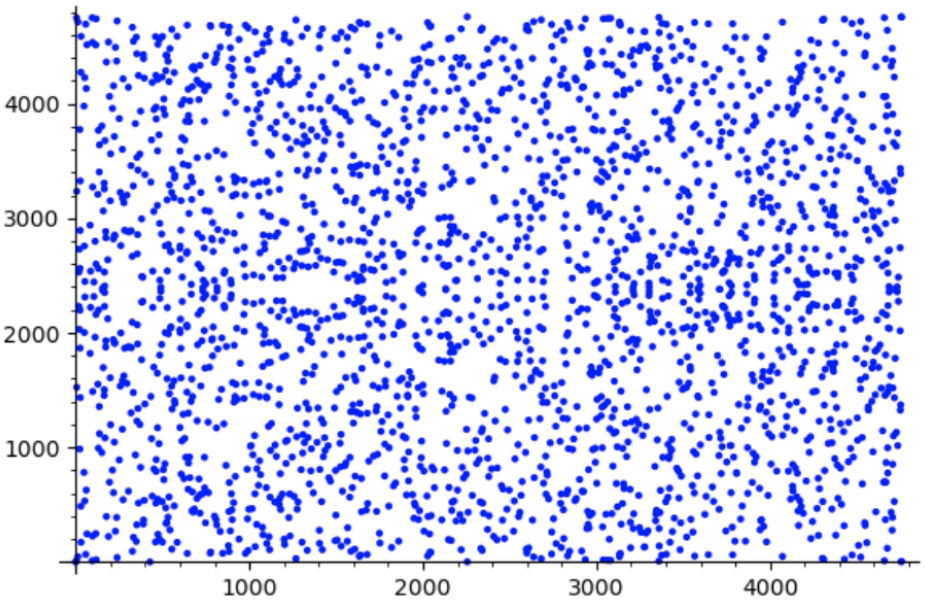
4. $P \neq Q$

$$m = (y_1 - y_2)(x_1 - x_2)^{-1} \ (mod\ p)$$
$$x_3 = (m^2 - x_1 - x_2) \quad (mod\ p)$$
$$y_3 = y_1 + m(x_3 - x_1) \quad (mod\ p)$$

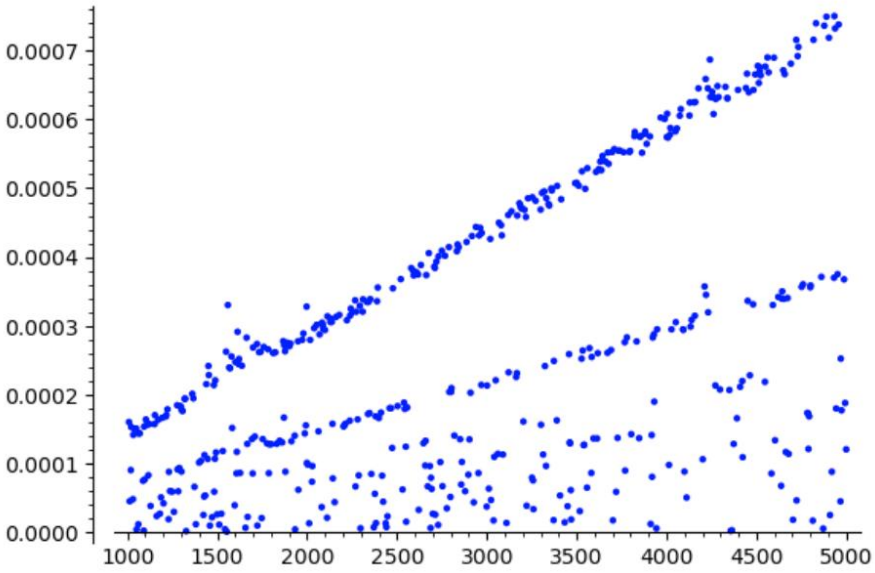
Умножение на число (возможно ускорение подобное быстрому возведению в степень):

1. $P \cdot 0 = 0$
2. $P \cdot n = \underbrace{P + P + \cdots + P}_{\text{N раз}}$

График эллиптической кривой с коэффициентами $a = -7, b = 10, p = 4759$:



Для решения поставленной задачи я выбрал коэффициенты $a = -7, b = 10$, и решил проверить, как будет зависеть порядок точки от величины характеристики поля p. График зависимости:



Видно, что порядок точки увеличивается как-то линейно. На основании этого факта можно было легко подобрать такое p, при котором поиск порядка начальной точки будет около 10 минут. На моем оборудовании подошло значение $p = 80000527$.

```
m = 80000527
m, ECFindOrderTime(-7, 10, m, *(5, 10)) # (80000527, (13.11787569920222, 80013441))
```

```
(80000527, (13.11787569920222, 80013441))
```

Время поиска порядка точки составило чуть больше 13 минут.
Алгоритм поиска подразумевает постоянное суммирование аккумулятора с т. P, пока первый не станет нулем. Код:

```
1.  def ECFindOrderTime(a, b, p, x0, y0):
2.      p1 = Point(a, b, p, x0, y0)
3.
4.      i = 1
5.      temp = p1
6.      result = [(p1.x, p1.y)]
7.
8.      start = time.time()
9.      while not(temp.x == temp.y == 0):
10.         temp += p1
11.         i+=1
12.     end = time.time()
13.
14.     return (end - start) / 60, i
```

Выводы

В ходе работы я познакомился с эллиптическими кривыми над конечными полями. Реализовал арифметические сложение точек и быстрое умножение на натуральное число, написал программу, определяющую порядок точки полным перебором. Ссылка на GitHub с реализацией. Также существует алгоритм Шуфа, определения количества точек эллиптической кривой со сложность $O(\log^6(p))$ и метод комплексного умножения.