Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «Криптография»

Студент:	А. В. Тимофеев
Преподаватель:	А. В. Борисов
Группа:	М8О-307Б-19
Дата:	
Оценка:	
Подпись:	

Москва, 2022

Лабораторная №3

Задача:

1. 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

1 Характеристики ПК студента

Процессор Intel Core i5-10600K @ 4.10GHz, память: 16 Gb, разрядность системы: 64.

2 Описание

Подобрать такую эллиптическую кривую над конечным простым полем порядка p, порядок точки которой полным перебором находится за 10 минут на ПК.

[1]: «Эллиптическая кривая — это просто множество точек, описываемое уравнением:

$$y2 = x3 + ax + b$$

»

Общий метод и алгоритм решения Можно использовать каноническую форму эллиптической кривой: $y2 = x3 + ax + b$

a и b пришлось подбирать случайно, а вот модуль кривой p – уже вручную, пока подсчёт порядка точки не стал удовлетворять условию.

Спустя несколько итераций, было найдено значение p, удовлетворяющее условию – 29959.

Алгоритм довольно простой: сначала полным перебором находятся находятся все точки, принадлежащие кривой, затем выбирается случайная точка и находится её порядок через сложение её самой с собой, пока сумма не станет точкой (0,0). И, разумеется, это работает за O(p^2)

Можно использовать алгоритм Шуфа (с теоремой Хассе) для ускорения решения задачи полного перебора. Сложность будет равняться O(log q), где q – число элементов поля.

Также есть метод комплексного умножения, он позволяет эффективнее находить кривые с заданным количеством точек. Но плох тем, что работает лишь при определённых условиях.

3 Исходный код

```
import time
import random


A = 46566389548546536960066742301497483930834559
B = 50102988429491433759123793207594191096335


def elliptic_curve(x, y, p):
    return (y ** 2) \% p == (x ** 3 + (A \% p) * x + (B \% p)) \% p


def print_curve(p):
    print("y^2 = x^3 + {0} * x + {1} (mod {2})".format(A \% p, B \% p, p))


def extended_euclidean_algorithm(a, b):
    s, old_s = 0, 1
    t, old_t = 1, 0
    r, old_r = b, a

    while r != 0:
        quotient = old_r // r
        old_r, r = r, old_r - quotient * r
        old_s, s = s, old_s - quotient * s
        old_t, t = t, old_t - quotient * t

    return old_r, old_s, old_t


def inverse_of(n, p):
    gcd, x, y = extended_euclidean_algorithm(n, p)
    assert (n * x + p * y) \% p == gcd

    if gcd != 1:
        raise ValueError(
            '{} has no multiplicative inverse '
            'modulo {}'.format(n, p))
    else:
        return x \% p


def add_points(p1, p2, p):
    if p1 == (0, 0):
        return p2
    elif p2 == (0, 0):
        return p1
```

```
    elif p1[0] == p2[0] and p1[1] != p2[1]:
        return (0, 0)

    if p1 == p2:
        s = ((3 * p1[0] ** 2 + (A \% p)) * inverse_of(2 * p1[1], p)) \% p
    else:
        s = ((p1[1] - p2[1]) * inverse_of(p1[0] - p2[0], p)) \% p
    x = (s ** 2 - 2 * p1[0]) \% p
    y = (p1[1] + s * (x - p1[0])) \% p
    return (x, -y \% p)


def order_point(point, p):
    i = 1
    check = add_points(point, point, p)
    while check != (0, 0):
        check = add_points(check, point, p)
        i += 1
    return i


if __name__ == '__main__':
    p = 29959

    print_curve(p)
    points = []
    start = time.time()
    for x in range(0, p):
        for y in range(0, p):
            if elliptic_curve(x, y, p):
                points.append((x, y))

    cnt_points = len(points)
    print("\n")
    print("Order curve = {0}".format(cnt_points))
    point = random.choice(points)

    print("Order point {0}: {1}".format(point, order_point(point, p)))
    print("Time: {} min.".format((time.time() - start)/60))
```

4 Консоль

```
y^2 = x^3 + 18375 * x + 2871 (mod 29959)

Order curve = 30119
Order point (17186,2225): 51476
Time: 10.278407212098439 min.
```

5 Выводы

Выполнив третью лабораторную работу по курсу «Криптография», я познакомился с новой для себя темой "эллиптические кривые"

Я научился искать порядок группы, которая образуется точками кривой и порядок подгруппы, для каждой точки эллиптической кривой. Также я узнал про алгоритмы и теоремы, использующиеся для облегчения и ускорения процесса полного перебора.

Список литературы

[1] *Доступно о криптографии на эллиптических кривых.*
URL: https://habr.com/ru/post/335906/ (дата обращения: 27.04.2022).