**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт: №8 «Информационные технологии   
и прикладная математика»   
Кафедра: 806 «Вычислительная математика   
и программирование»**

Лабораторная работа № 4   
по курсу «Криптография»

Группа: М8О-308Б-22

Студент(ка): А. Д. Галкин

Преподаватель: А. В. Борисов

Оценка:

Дата: 19.03.2025

Москва, 2025

**ОГЛАВЛЕНИЕ**

[**1** **Тема** 3](#_Toc197028801)

[Исследование эллиптических кривых над конечным простым полем 3](#_Toc197028802)

[**2** **Задание** 3](#_Toc197028803)

[**3** **Теория** 3](#_Toc197028804)

[Эллиптические кривые 3](#_Toc197028805)

[Операция сложения 3](#_Toc197028806)

[Кривые над конечным полем 4](#_Toc197028807)

[Особенности сложения 4](#_Toc197028808)

[Порядок 4](#_Toc197028809)

[Задача дискретного логарифмирования 5](#_Toc197028810)

[**4** **Ход лабораторной работы** 6](#_Toc197028811)

[Результат работы программы 6](#_Toc197028812)

[**5** **Выводы** 8](#_Toc197028813)

[**6** **Список используемой литературы** 9](#_Toc197028814)

# **Тема**

# Исследование эллиптических кривых над конечным простым полем

# **Задание**

1. Подобрать такую эллиптическую кривую, порядок точки которой полным перебором находится за 10 минут на ПК.
2. Упомянуть в отчёте результаты замеров работы программы, характеристики вычислителя.
3. Указать какие алгоритмы и/или теоремы существуют для облегчения и ускорения решения задачи полного перебора.

Рассмотреть для случая конечного простого поля Z\_p.

# **Теория**

## Эллиптические кривые

Эллиптические кривые представляют собой специальный вид кривых, которые описываются уравнением Вейерштрасса:

В криптографии также накладывается дополнительное ограничение на эллиптические кривые: . Кривые, удовлетворяющие этому условию, называются гладкими.

## Операция сложения

Мы можем также определить на эллиптической кривой алгебру.

**Сложение трех точек** на эллиптической кривой обычно определяется следующим образом: если мы имеем три точки , и на кривой, и они лежат на одной прямой, то их сумма равна нулю (то есть ). Если мы хотим сложить только две точки, скажем и , мы просто берем точку , которая является третьей точкой на прямой через и , и определяем как обратную к . Если же , то точка является точкой пересечения касательной в точке с кривой.

**Обратной точкой** точке будем называть точку , которая является зеркальной версией точки относительно оси .

**Нулевая точка** на эллиптической кривой представляет собой "точку в бесконечности". Она играет роль нейтрального элемента в группе, определенной на кривой. Это означает, что для любой точки на кривой выполняется равенство . Кроме того, если точка находится на кривой, то .

## Кривые над конечным полем

В криптографии используется специальный вид кривых, которые определены на конечном поле. Это поле представляет собой кольцо вычетов по модулю простого числа. Такое поле обычно обозначается как и содержит целые положительные числа в диапазоне . Точки на такой кривой должны удовлетворять следующему равенству:

То есть, все операции проводятся в контексте модуля .

На таком кольце определены операции сложения, вычитания, умножения по модулю. Также, над кольцом определена операция дискретного логарифма, который на самом деле представляет собой аналог операции "деления" над полем .

## Особенности сложения

Операция сложения в таком поле имеет свои особенности. Если мы складываем две точки и на эллиптической кривой, результатом будет точка , где:

здесь представляет наклон прямой, проходящей через точки и .

## Порядок

**Порядок кривой** соответствует количеству точек на кривой, включая точку в бесконечности. В контексте эллиптической кривой, порядок точки означает такое наименьшее положительное число , при котором , где - это точка в бесконечности, а - это результат сложения точки самой с собой раз.

## Задача дискретного логарифмирования

В контексте эллиптических кривых, задача дискретного логарифмирования формулируется следующим образом. Пусть даны точки , на эллиптической кривой, и требуется найти такое целое число , которое будет удовлетворять следующему равенству

где означает результат сложения точки с самой собой раз.

Такая задача является на текущей момент "сложной". То есть, не было придумано алгоритма, который смог бы решить поставленную задачу за приемлемое время. Если бы был найден эффективный алгоритм для решения этой задачи, то большинство современных криптосистем стали бы небезопасными.

Сложность задачи дискретного логарифмирования в контексте эллиптических кривых обусловлена тем, что операции на эллиптических кривых отличаются от операций в обычных группах. В частности, операция сложения на эллиптической кривой не является коммутативной, что затрудняет применение многих известных алгоритмов. Кроме того, эффективные алгоритмы для решения задачи дискретного логарифмирования в других группах, таких как группа целых чисел по модулю $p$, не могут быть применены к эллиптическим кривым, что делает задачу дискретного логарифмирования на них особенно сложной.

# **Ход лабораторной работы**

Я использовал каноническую форму эллиптической кривой:

Коэффициенты a и b были выбраны случайно. Модуль кривой p подбирался вручную, пока подсчёт порядка точки не стал удовлетворять условию.

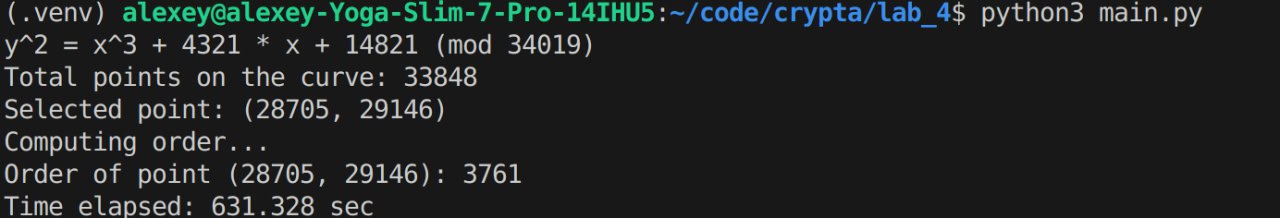
Через пару итераций был подобран p = 34019, удовлетворяющий условию задания.

Алгоритм работы: полным перебором нахожу все точки, принадлежащие кривой. Выбираю случайную точку, нахожу её порядок путём сложения самой с собой до тех пор, пока эта сумма не станет точкой (0, 0). Полный перебор, как несложно догадаться, работает за O()

Все дальнейшие вычисления производились на ПК со следующими характеристиками:

* Процессор**:** Intel Core i7-11370H, 4 ядра / 8 потоков, частота до 4.80 GHz
* ОЗУ: 16 ГБ DDR4

## Результат работы программы



**Листинг**

import time

import random

A = 1860348749492490789823288813930625381760

B = 2001637506671384833171818673149062805974

def elliptic\_curve(x, y, p):

return (y \*\* 2) % p == (x \*\* 3 + (A % p) \* x + (B % p)) % p

def print\_curve(p):

print(f"y^2 = x^3 + {A % p} \* x + {B % p} (mod {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):

n = n % p

gcd, x, y = extended\_euclidean\_algorithm(n % p, p)

if gcd != 1:

raise ValueError(f"{n} has no inverse mod {p}")

return x % p

def add\_points(p1, p2, p):

if p1 == (0, 0):

return p2

if p2 == (0, 0):

return p1

if p1[0] == p2[0] and (p1[1] + p2[1]) % p == 0:

return (0, 0)

if p1 == p2:

s = ((3 \* p1[0] \*\* 2 + A % p) \* inverse\_of(2 \* p1[1], p)) % p

else:

dx = (p2[0] - p1[0]) % p

dy = (p2[1] - p1[1]) % p

s = (dy \* inverse\_of(dx, p)) % p

x3 = (s \*\* 2 - p1[0] - p2[0]) % p

y3 = (s \* (p1[0] - x3) - p1[1]) % p

return (x3, y3)

def order\_point(point, p):

result = point

i = 1

while result != (0, 0):

result = add\_points(result, point, p)

i += 1

return i

if \_\_name\_\_ == '\_\_main\_\_':

p = 34019

print\_curve(p)

points = []

start = time.time()

for x in range(p):

for y in range(p):

if elliptic\_curve(x, y, p):

points.append((x, y))

print("Total points on the curve:", len(points))

point = random.choice(points)

print(f"Selected point: {point}")

print("Computing order...")

order = order\_point(point, p)

print(f"Order of point {point}: {order}")

print("Time elapsed: {:.3f} sec".format(time.time() - start))

# **Выводы**

Для ускорения решения задачи полного перебора используются, например, алгоритм Шуфа, использующий теорему Хассе. Его сложность , где q — число элементов поля. Есть алгоритм Baby-step Giant-step, который находит порядок отдельной точки. Сложность О(). Еще существует метод комплексного умножения, который позволяет более эффективно находить кривые с заданным количеством точек. Однако в отличие от алгоритма Шуфа, который является универсальным, метод комплексного умножения работает только при выполнении определенных условий.

# **Список используемой литературы**

* + - 1. "Доступно о криптографии на эллиптических кривых" Хабр: <https://habr.com/ru/articles/335906/>
      2. "Что такое шифрование на основе эллиптических кривых" keepersecurity: <https://www.keepersecurity.com/blog/ru/2023/06/07/what-is-elliptic-curve-cryptography/>