**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 №5   
по курсу «Криптография»

Группа: М8О-306Б-21

Студент: В.Р. Орусский

Преподаватель: А. В. Борисов

Оценка:

Дата: 18.05.2024

Москва, 2024

**ОГЛАВЛЕНИЕ**

[1 Тема 3](#_Toc158983147)

[2 Задание 3](#_Toc158983148)

[3 Теория 4](#_Toc158983149)

[4 Ход лабораторной работы 5](#_Toc158983150)

[5 Выводы 6](#_Toc158983151)

# **Тема**

Эллиптическая кривая. Определение порядка точки кривой за определённое время.

# **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 по модулю p (Z/p).

# **Теория**

Эллиптическая кривая над полем K - неособая кубическая кривая на проективной плоскости над алгебраическим замыканием поля 𝐾, задаваемая уравнением 3-й степени с коэффициентами из поля 𝐾 и «точкой на бесконечности». Так как, нас интересует конечное простое поле целых чисел по модулю, то можно рассмотреть только каноническую форму эллиптических кривых:

Конечное поле по модулю – поле, состоящее из конечного числа элементов, в данном случае это количество ограничено модулем. То есть, поле Z по модулю N имеет значения от 0 до N - 1.

Теперь, мы рассматриваем не целостную кривую, а набор точек, симметричный, относительно

Порядок точки p на эллиптической кривой — это наименьшее положительное число n, такое что n\*p=O, где O — "бесконечно удаленная" точка, служащая нейтральным элементом группы.

Мультипликативная инверсия (модульная арифметика) для числа a это такое целое число x, что a\*x сравнимо с 1 по модулю m:

Для нахождения порядка точки, необходимо знать, как производить деление эллиптических кривых внутри поля по модулю, для этого используется умножение на скаляр, а также нахождение обратной величины (для этого будет использоваться расширенный алгоритм Евклида).

Расширенный алгоритм Евклида — модификация алгоритма Евклида, вычисляющая, кроме наибольшего общего делителя (НОД) целых чисел 𝑎 и 𝑏, ещё и коэффициенты соотношения Безу, то есть такие x, y, что:

Нахождение порядка точки «в лоб» довольно ресурсоёмко и является переборным алгоритмом, поэтому ниже представлены алгоритмы, работающие быстрее.

Теоремы Хассе (сужает диапазон поиска)

Для эллиптической кривой над полем Z/p, порядок кривой (количество точек на кривой) ограничен теоремой Хассе:

.

Алгоритм Полларда (Полига-Хеллмана)

Алгоритм Полларда для ро-метода факторизации, а также его модификация для нахождения порядка элемента в группе, используют идею случайных прогулок для определения циклов и, соответственно, факторов порядка группы. Выбираются случайные точки и выполняются операции группы (например, сложение точек на эллиптической кривой), формируя "случайную прогулку" по элементам группы. Используется идея Флойда для обнаружения циклов в последовательности точек. Когда цикл найден, можно вычислить порядок (или фактор порядка) элемента.

Алгоритм Бейбиджа-Шэнкса

Алгоритм Бейбиджа-Шэнкса предназначен для нахождения логарифма в группе (в нашем контексте — порядка точки на эллиптической кривой), используя метод "встречи посередине". Этот алгоритм эффективен, когда размер группы известен и невелик. Проблема нахождения порядка разбивается на две меньшие задачи, которые решаются независимо, обычно через создание двух списков: один для "прямых" операций, другой — для "обратных". Ищется совпадение между значениями в двух списках, что позволяет вычислить искомый порядок (или логарифм) "по середине" изначальной задачи.

# **Ход лабораторной работы**

Работу я выполнял на языке Python. Основная идея алгоритма: выбор параметров кривой, генерация точек на ней и вычисление порядка случайной точки с последующим увеличением параметра p для поиска подходящей кривой. Программа запрашивает у пользователя параметры a, b и p для эллиптической кривой и время, в течение которого должен выполняться поиск. Затем, используя экземпляр класса EllipticCurve, программа в цикле ищет такую кривую, порядок точки которой можно вычислить в указанное время. Для этого программа увеличивает параметр p на фиксированное значение (3000) на каждой итерации, пытаясь найти подходящую кривую.

Класс EllipticCurve:

Инициализация: принимает коэффициенты a, b и p, проверяя условие несингулярности кривой.

Проверка принадлежности точки кривой: метод is\_elliptic\_curve используется для проверки, удовлетворяет ли точка уравнению эллиптической кривой.

Вычисление обратного элемента: метод inverse\_of вычисляет обратный элемент для заданного числа в поле по модулю p, используя расширенный алгоритм Евклида.

Сложение точек на кривой: метод add\_points реализует операцию сложения двух точек на эллиптической кривой.

Вычисление порядка точки: метод order\_point находит порядок заданной точки путём повторного сложения точки с самой собой до тех пор, пока не будет достигнута нейтральная точка.

Шаг итерации: метод step выполняет один шаг итерации, включая генерацию точек на кривой и вычисление порядка случайно выбранной точки.

Проверка на простоту и поиск следующего простого числа: методы is\_prime\_number и get\_next\_prime\_number используются для проверки чисел на простоту и поиска следующего простого числа, начиная с заданного значения.

Но из-за того, что я использовал Python, программа работала сильно дольше заданного времени. Вот ряд тестов:

# **Выводы**

В процессе выполнения данной ЛР, я узнал много нового, а также повторил, что было известно о модели OSI / ISO, почитал побольше про протоколы, используемые для передачи и защиты данных в интернете. Изучил процесс работы с WireShark, научился ориентироваться в нём, применять фильтры, отслеживать пакеты, находить нужные. Косвенно изучил некоторые стандарты, по которым существует интернет (RFC).

# **Список используемой литературы**

1. Расширенный алгоритм Евклида - [https://ru.wikipedia.org/wiki/](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%B0)
2. Мультипликативная инверсия - [https://ru.wikipedia.org/wiki/](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E_%D1%87%D0%B8%D1%81%D0%BB%D0%BE)
3. Эллиптическая кривая - [https://ru.wikipedia.org/wiki](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%BB%D0%B8%D0%BF%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BA%D1%80%D0%B8%D0%B2%D0%B0%D1%8F)
4. Методичка по работе с эллиптическими кривыми - <https://habr.com/ru/articles/335906/>