Лабораторная работа №7

Дискретное логарифмирование

Выполнил: Хамза Хуссен

Цели работы

Изучить задачу дискретного логарифмирования.

Познакомиться с теоретическими основами решения в конечных группах.

Реализовать и протестировать один из алгоритмов решения — ρ-алгоритм Полларда.

Теоретическая основа

Задача дискретного логарифмирования:
Найти x в уравнении:

$$g^x \equiv a \ (mod p)$$

где:

- G — конечная мультипликативная абелева группа,
- g — образующий элемент,
- $a \in G$.

Условия разрешимости:
Если группа циклическая и порождается g, решение существует для любого a.

Сложность задачи

•**Полный перебор**: требует до | G | шагов.

•**Алгоритмы эффективного решения**:

- ρ-алгоритм Полларда,
- Baby-step Giant-step,
- Алгоритм Полига — Хеллмана.

Область применения: криптография, защита данных, электронная подпись.

ρ-алгоритм Полларда

Идея метода: использование "парадокса дней рождений" и сжимающей функции.

Входные данные:

- p — простое число,
- a, b — элементы группы,
- f — сжимающее отображение.

Выход: показатель x или сообщение об отсутствии решения.

Этапы алгоритма

1.Инициализация:
Выбрать u, v, вычислить $c \equiv a^u b^v \ (mod p)$,$d \equiv c$.

2.Итерация:
Повторять:

$$c \leftarrow f(c), d \leftarrow f\big(f(d)\big)$$

1.до совпадения $c \equiv d \ (mod p)$.

2.Решение уравнения:
Найти x из линейного сравнения по модулю порядка r.

Реализация на Python

Основные функции:

ext_euclid — расширенный алгоритм Евклида,

inverse — вычисление обратного элемента,

xab — шаг преобразования,

pollrad — реализация алгоритма Полларда,

verify — проверка результата.

Пример выполнения

Входные данные:

$$g = 5,\ h = 64,\ p = 107$$

Код вызова:

python

```
res = pollrad(5, 64, 107)

print(res)
```

Результат:

Найденный x, проверка: pow(g, x, p) == h.

Результаты тестирования

Контрольный пример:

(5, 64, 107) : 52

Validates: True

Вывод:

Программа корректно находит дискретный логарифм для заданных входных данных.

Выводы

Изучена задача дискретного логарифмирования.

Реализован ρ-алгоритм Полларда на языке Python.

Проверена корректность работы на контрольном примере.

Алгоритм эффективен для решения задачи в циклических группах умеренного размера.