Лабораторная работа №7. Дискретное логарифмирование в конечном поле.

Предмет: Математические основы защиты информации и информационной безопасности

Александр Сергеевич Баклашов

Содержание

1 Цель работы **4**

2 Задание **5**

3 Теоретическое введение **6**
3.1 ρ-Метод Полларда 6
3.1.1 Задача 6

4 Выводы **8**

5 Библиография **9**

List of Figures

3.1 p-Метод Полларда 7

1 Цель работы

Рассмотреть и реализовать алгоритм, реализующий ρ-метод Полларда для задач дискретного логарифмирования.

2 Задание

Реализовать алгоритм, реализующий ρ-метод Полларда для задач дискретного логарифмирования.

3 Теоретическое введение

p-Метод Полларда:

ρ-Метод Полларда для дискретного логарифмирования (ρ-метод) — алгоритм дискретного логарифмирования в кольце вычетов по простому модулю, имеющий экспоненциальную сложность. Предложен британским математиком Джоном Поллардом (англ. John Pollard) в 1978 году, основные идеи алгоритма очень похожи на идеи ро-алгоритма Полларда для факторизации чисел. Данный метод рассматривается для группы ненулевых вычетов по модулю ρ, где ρ — простое число, большее 3. # Выполнение лабораторной работы

3.1 ρ-Метод Полларда

3.1.1 Задача

Реализовать ρ-Метод Полларда для задач дискретного логарифмирования.

3.1.1.1 Решение

Реализуем ρ-Метод Полларда (рис. 3.1)

In [1]:

```
import time
start_time = time.time()
Flag = False

def f(c, u, v):
    if c < 53:
        return (10 * c) % 107, u + 1, v
    elif (c >= 53):
        return (64 * c) % 107, u, v + 1

p = 107  # Простое число p
a = 10   # Число a
b = 64   # Число b
r = 53   # Порядок числа a по модулю p
u = 2    # Произвольное число
v = 2    # Произвольное число

# Инициализация переменных для чисел c и d, а также их параметров u и v
uc = 2
vc = 2
ud = 2
vd = 2

# Вычисление начальных значений c и d
c = ((a ** u) * (b ** v)) % p
d = c

# Применение функции отображения f к числам c и d и их параметрам u и v
c, uc, vc = f(c, uc, vc)
d, ud, vd = f(f(d, ud, vd)[0], f(d, ud, vd)[1], f(d, ud, vd)[2])

# Цикл, выполняющий алгоритм p-метода Полларда до совпадения чисел c и d
while c % p != d % p:
    c, uc, vc = f(c, uc, vc)
    d, ud, vd = f(f(d, ud, vd)[0], f(d, ud, vd)[1], f(d, ud, vd)[2])

# Нахождение искомой степени числа a
x = 1
while (uc + vc * x) % r != (ud + vd * x) % r:
    x += 1
    elapsed_time = time.time() - start_time
    if elapsed_time > 5:
        print("Решений нет")
        Flag = True
        break

# Вывод степени числа a
if (Flag != True):
    print("x =", x)
```

```
x = 20
```

Figure 3.1: p-Метод Полларда

4 Выводы

В ходе данной лабораторной работы я рассмотрел и реализовал алгоритм, реализующий ρ-метод Полларда для задач дискретного логарифмирования.

5 Библиография

1. Python documentation. [Электронный ресурс]. М. URL: Python documentation (Дата обращения: 28.09.2023).

2. Лабораторная работа №7. Дискретное логарифмирование в конечном поле. - 4 с. [Электронный ресурс]. М. URL: Лабораторная работа №7. Дискретное логарифмирование в конечном поле. (Дата обращения: 30.11.2023).