

## Лабораторная работа 7

Терентьев Егор Дмитриевич, НФИмд-01-23

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ПРЕЗЕНТАЦИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

дисциплина: Математические основы защиты информации  
и информационной безопасности

Преподаватель: Кулябов Дмитрий Сергеевич

Студент: Терентьев Егор Дмитриевич

Группа: НФИмд-01-23

МОСКВА

2023 г.

# **Прагматика выполнения лабораторной работы**

# Прагматика выполнения лабораторной работы

Требуется реализовать:

1. Алгоритм, реализующий  $p$ -метод Полларда для задач дискретного логарифмирования

## Цель работы

# Цель работы

Освоить на практике дискретное логарифмирование в конечном поле.

## **Выполнение лабораторной работы**

1. Для реализации р-метода Полларда:



# 1. Для реализации р-метода Полларда:

1. Функция, реализующая р-метод Полларда
2. Функция нахождения НОД
3. Расширенный алгоритм Евклида для вычисления модульного обратного элемента

```
1  def pollard_p_method(p, a, b, f, r, u, v):
2      # Выбор произвольных чисел u, v
3      c = (a ** u * b ** v) % p
4      d = c
5
6      # Итерации метода Полларда
7      while True:
8          c = f(c) % p
9          d = f(f(d)) % p
10
11         # Если c = d, вычисляем логарифмы для c и d
12         if c == d:
13             # Вычисляем логарифм x решением сравнения по модулю r
14             # Решения нет, если r и p не взаимно просты
15             if gcd(r, p-1) != 1:
16                 return "Решения нет"
17
18             # Вычисляем логарифм x
19             x = (u - v * modinv((u - v), r)) * (c - a ** u) % r % r
20             return x
21
22     # Функция вычисления наибольшего общего делителя (GCD)
23     def gcd(a, b):
24         while b:
```

2. Основная функция запуска где получаем входные значения и шифруем слово

## 2. Основная функция запуска где получаем входные значения и шифруем слово

```
37 # Пример использования
38 p = 107
39 a = 10
40 b = 64
41 r = 53
42 u = 2
43 v = 2
44
45 # Определение функции f
46 def f(c):
47     if c < r:
48         return (10 * c) % p
49     else:
50         return (64 * c) % p
51
52 result = pollard_p_method(p, a, b, f, r, u, v)
53 print("Решение:", result)
54
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS F:\учеба 5 курс\информационная безопасность> python .\Lab7_log\lab7_logs.py
Решение: Решения нет
○ PS F:\учеба 5 курс\информационная безопасность> |
```

Figure 2: output

## Выводы

## Выводы

В результате выполнения работы я освоил на практике дискретное логарифмирование в конечном поле.

