Лабораторная работа №6. Разложение чисел на множители.

Предмет: Математические основы защиты информации и информационной безопасности

Александр Сергеевич Баклашов

Содержание

List of Figures

1 Цель работы

Рассмотреть и реализовать алгоритм, реализующий p-метод Полларда и реализовать метод квадратов.

2 Задание

Реализовать следующие алгоритмы:

- p-метод Полларда;
- Метод квадратов.

3 Теоретическое введение

p-Метод Полларда:

p-Метод Полларда — это один из методов факторизации составных чисел, который был разработан Джоном Поллардом. Метод основан на свойствах мультипликативной группы вычетов по модулю простого числа p.

Метод Полларда работает основываясь на том, что при достаточном количестве итераций два значения в последовательности x_i будут находиться в одном цикле и могут быть использованы для вычисления делителя.

Применение параметра c в функции $f(x)$ позволяет разнообразить последовательность значений $f(x)$ и улучшить эффективность метода в некоторых случаях.

Метод Полларда является одним из алгоритмов факторизации, используемых для разложения больших составных чисел на их простые множители.

Метод квадратов:

Метод квадратов (Quadratic Sieve) — это алгоритм факторизации целых чисел, разработанный Карлом Померанцем и Джоном Поллелем. Этот метод основан на поиске целых чисел, которые представляют собой разность двух квадратов. Метод квадратов является одним из эффективных методов для факторизации больших составных чисел и часто применяется в криптографии. Вместе с методом факторизации ро (алгоритмом Полларда), он используется для атаки на криптографические системы, основанные на сложности факторизации больших чисел.

4 Выполнение лабораторной работы

4.1 p-Метод Полларда

4.1.1 Задача

Реализовать p-Метод Полларда

4.1.1.1 Решение

Реализуем p-Метод Полларда (рис. 4.1)

In [1]:
```python
from math import gcd

def f(x, n):
    # Функция для вычисления f(x) (mod n)
    return (x**2 + 5) % n

def find_divisor(n,c):
    a, b = c, c
    while True:
        # Шаг 2: Вычислить a <- f(a) (mod n), b <- f(b) (mod n)
        a = f(a, n)
        b = f(f(b, n), n)

        # Шаг 3: Найти НОД(a - b, n)
        d = gcd(abs(a - b), n)

        # Шаг 4: Проверить результат
        if 1 < d < n:
            # Делитель найден
            return d
        elif d == n:
            # Делитель не найден
            return "Делитель не найден"


c = 1
n = 1359331
result = find_divisor(n,c)
print("Нетривиальный делитель числа", n,"=",result)
```

```
Нетривиальный делитель числа 1359331 = 1181
```

Figure 4.1: p-Метод Полларда

4.2 Метод квадратов

4.2.1 Задача

Реализовать метод квадратов

4.2.1.1 Решение

Реализуем метод квадратов (рис. 4.2)

In [2]:
```python
from math import isqrt

def square_factorization(n):
    a = isqrt(n) + 1  # ближайшее целое квадратному корню из n
    b2 = a**2 - n

    while not is_square(b2):
        a += 1
        b2 = a**2 - n

    b = isqrt(b2)
    return a + b, a - b  # разложение числа на множители

def is_square(x):
    return isqrt(x)**2 == x


n = 1359331
factors = square_factorization(n)
print(f"Разложение числа {n} на множители: {factors}")
```

```
Разложение числа 1359331 на множители: (1181, 1151)
```

Figure 4.2: Метод квадратов

5 Выводы

В ходе данной лабораторной работы я рассмотрел и реализовал следующие алгоритмы:

- p-метод Полларда;
- Метод квадратов.

6 Библиография

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

2. Лабораторная работа №6. Разложение чисел на множители. - 3 с. [Электронный ресурс]. М. URL: Лабораторная работа №6. Разложение чисел на множители. (Дата обращения: 22.11.2023).