Лабораторная работа №6

Математические основы защиты информации и информационной безопасности

Леонтьева Ксения Андреевна | НПМмд-02-23

Содержание

Список иллюстраций

1 Цель работы

Реализовать на языке программирования p-метод Полларда.

2 Теоретическое введение

Задача разложения составного числа на множители формулируется следующим образом: для данного положительного целого числа n найти его каноническое разложение $n = p_1^{\alpha_1} p_2^{\alpha_2} ... p_s^{\alpha_s}$, где p_i - попарно различные простые числа, $\alpha_i \geq 1$.

На практике необязательно находить каноническое разложение числа n. Достаточно найти его разложение на два *нетривиальных сомножителя:* $n = pq, 1 \leq p \leq q < n$.

p-метод Полларда. Пусть n - нечетное составное число, $S = \{0, 1, ..., n-1\}$ и $f : S \to S$ - случайное отображение, обладающее сжимающими свойствами, например, $f(x) \equiv (x^2 + 1)(mod\ n)$. Основная идея метода состоит в следующем. Выбираем случайный элемент $x_0 \in S$ и строим последовательность $x_0, x_1, x_2, ...$, определяемую рекуррентным соотношением

$$x_{i+1} = f(x_i),$$

где $i \geq 0$, до тех пор, пока не найдем такие числа i, j, что $i < j$ и $x_i = x_j$. Поскольку множество S конечно, такие индексы i, j существуют. Последовательность $\{x_i\}$ будет состоять из “хвоста” $x_0, x_1, ..., x_{i-1}$ длины $O(\sqrt{\frac{\pi n}{8}})$ и цикла $x_i = x_j, x_{i+1}, ..., x_{j-1}$ той же длины.

Более подробно см. в [1].

3 Выполнение лабораторной работы

p-метод Полларда реализуем по следующей схеме:

На вход подается число n, начальное значение c, функция f, обладающая сжимающими свойствами.

1. Положить $a \leftarrow c, b \leftarrow c$.
2. Создать функцию $f(x, n) = (x^2 + 5)(mod\, n)$
3. Вычислить $a \leftarrow f(a, n), b \leftarrow f(f(b, n), n)$.
4. Найти $d \leftarrow$ НОД $(a - b, n)$
5. Если $1 < d < n$, то положить $p \leftarrow d$ и результат: p. При $d = n$ результат: “Делитель не найден”; при $d = 1$ вернуться на шаг 2.

Код программы (рис. 3.1).

```python
import numpy as np
import math

def f(x, n):
    return (x ** 2 + 5) % n

n = 1359331
a = b = 1
d = 1
i = 0
while d == 1:
    a = f(a,n)
    b = f(f(b,n),n)
    d = math.gcd(a - b, n)
    print('Итерация', i+1, ' ', 'a =',a, ' ', 'b =',b, ' ', 'd =',d)
    i = i + 1
if d == n:
    print('Делитель не найден')
else:
    print('Нетривиальный делитель числа', n, 'равен', d)
```

```
Итерация 1   a = 6   b = 41   d = 1
Итерация 2   a = 41   b = 123939   d = 1
Итерация 3   a = 1686   b = 391594   d = 1
Итерация 4   a = 123939   b = 438157   d = 1
Итерация 5   a = 435426   b = 582738   d = 1
Итерация 6   a = 391594   b = 1144026   d = 1
Итерация 7   a = 1090062   b = 885749   d = 1181
Нетривиальный делитель числа 1359331 равен 1181
```

Рис. 3.1: p-метод Полларда

4 Выводы

В ходе выполнения данной лабораторной работы был реализован p-метод Полларда.

Список литературы

1. ρ-метод Полларда [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm.