Лабораторная работа №6

Математические основы защиты информации и информационной безопасности

Николаев Дмитрий Иванович, НПМмд-02-24

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить работу алгоритмов разложения чисел на множители: ρ-Метод Полларда; Метод квадратов (Теорема Ферма о разложении); а также реализовать их программно.

2 Теоретическое введение

2.1 Разложение чисел на множители

Задача разложения на множители — одна из первых задач, использованных для построения криптосистем с открытым ключом.

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

На практике не обязательно находить каноническое разложение числа n. Достаточно найти его разложение на два нетривиальных сомножителя: $n = pq$, $1 \leq p \leq q < n$. Далее будем понимать задачу разложения именно в этом смысле.

2.2 ρ-Метод Полларда

Пусть n — нечетное составное число, $S = 0, 1, \dots, n-1$ и $f : S \to S$ — случайное отображение, обладающее сжимающими свойствами, например $f(x) = x^2 + 1 \mod n$. Основная идея метода состоит в следующем. Выбираем случайный элемент $x_0 \in S$ и строим последовательность $x_0, x_1, x_2, \dots$, 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 “зацикливается”). 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.

2.2.1 Алгоритм, реализующий ρ-Метод Полларда

Вход: Число n, начальное значение c, функция f, обладающая сжимающими свойствами.

Выход: Нетривиальный делитель числа n.

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

2.2.2 Пример

Найти ρ-методом Полларда нетривиальный делитель 1359331. Положим $c = 1$ и $f(x) = x^2 + 5 \mod n$. Работа иллюстрируется следующей последовательностью:

1. Рассмотрим первый цикл алгоритма
    1. $a = 1, b = 1$;
    2. $a \equiv 1^2+5 \mod n \equiv 6, b \equiv f(1^2+5 \mod n) \equiv 6^2+5 \mod n \equiv 41$;
    3. $d =$ НОД$(a - b, n) =$ НОД$(6 - 41, n) = 1$;
    4. $d = 1$, значит возвращаемся на второй шаг.
2. Рассмотрим второй цикл алгоритма

   1. $a \equiv 6^2 + 5 \mod n \equiv 41, b \equiv 123939;$
   2. $d = $ НОД$(a - b, n) = 1;$

3. Рассмотрим третий цикл алгоритма

   1. $a \equiv 41^2 + 5 \mod n \equiv 1686, b \equiv 391594;$
   2. $d = $ НОД$(a - b, n) = 1;$

4. Рассмотрим четвёртый цикл алгоритма

   1. $a \equiv 1686^2 + 5 \mod n \equiv 123939, b \equiv 438157;$
   2. $d = $ НОД$(a - b, n) = 1;$

5. Рассмотрим пятый цикл алгоритма

   1. $a \equiv 123939^2 + 5 \mod n \equiv 435426, b \equiv 582738;$
   2. $d = $ НОД$(a - b, n) = 1;$

6. Рассмотрим шестой цикл алгоритма

   1. $a \equiv 435426^2 + 5 \mod n \equiv 391594, b \equiv 1144026;$
   2. $d = $ НОД$(a - b, n) = 1;$

7. Рассмотрим седьмой цикл алгоритма

   1. $a \equiv 391594^2 + 5 \mod n \equiv 1090062, b \equiv 885749;$

Таким образом, $p = $ НОД$(a - b, n) = $ НОД$(1090062 - 885749, 1359331) = 1181$ является нетривиальным делителем числа 1359331.

2.3 Метод квадратов (Теорема Ферма о разложении)

Для любого положительного нечетного числа n существует взаимно однозначное соответствие между множеством делителей числа n, не меньших, чем $\sqrt{n}$, и множеством пар $\{s, t\}$ таких неотрицательных целых чисел, что $n = s^2 - t^2$.

2.3.1 Пример

У числа 15 два делителя, не меньших, чем $\sqrt{15}$ — это числа 5 и 15. Тогда получаем два представления:

1. $15 = pq = 3 \cdot 5$, откуда $s = 4, t = 1$ и $15 = 4^2 - 1^2$;
2. $15 = pq = 1 \cdot 15$, откуда $s = 8, t = 7$ и $15 = 8^2 - 7^2$.

3 Выполнение лабораторной работы

Действуя согласно [1], реализуем все описанные алгоритмы на языке Julia.

Реализуем функцию ρ-метода Полларда для нахождения нетривиального делителя составного числа (Рис.[3.1]) и метод квадратов (теорема Ферма о разложении) для разложения составного числа на 2 нетривиальных множителя (Рис.[3.2]). Найдём делители и разложим на множители числа 15 и 1359331 (Рис.[3.3]), в результате чего получим следующий вывод, представленный на Рис.[3.4].

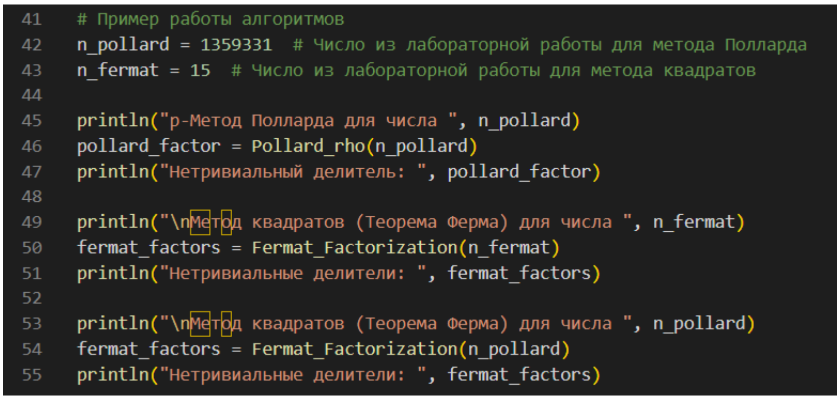
```julia
using Random

"""rho-Метод Полларда"""
function Pollard_rho(n::Int, f = x -> x^2 + 1)::Int
    # Начальное значение
    c = rand(1:n-1)
    a = c
    b = c
    # НОД, начальное значение 1
    d = 1
    while d == 1
        # Генерация последовательности
        a = f(a) % n
        b = f(f(b) % n) % n
        # НОД как разница между a и b
        d = gcd(abs(a - b), n)
    end
    if d == n
        return 1 # Делитель не найден
    else
        return d
    end
end
```

Рис. 3.1: Код алгоритма ρ-метода Полларда на Julia

```julia
"""Метод квадратов (Теорема Ферма о разложении)"""
function Fermat_Factorization(n::Int)::Tuple{Int, Int}
    # Начальное значение как округлённый корень исходного числа
    s = ceil(Int, sqrt(n))
    # Из соотношение n = s^2 - t^2
    t2 = s^2 - n
    # Пока соотношение не стало точным для целых чисел
    while sqrt(t2) != floor(sqrt(t2))
        s += 1 # Увеличиваем s
        t2 = x^2 - n
    end
    # Вычисляем t
    t = sqrt(t2)
    return (s - t, s + t)
end
```

Рис. 3.2: Код алгоритма метода квадратов (теорема Ферма о разложении) на Julia



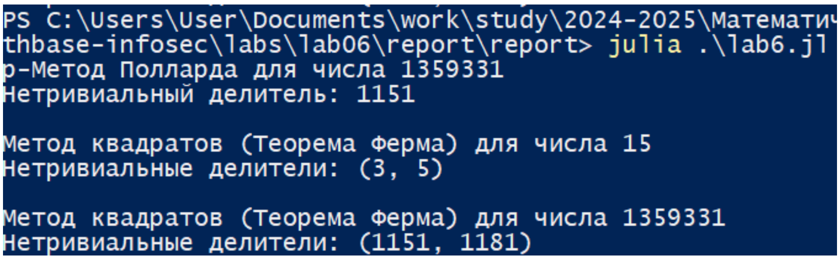
```
# Пример работы алгоритмов
n_pollard = 1359331  # Число из лабораторной работы для метода Полларда
n_fermat = 15  # Число из лабораторной работы для метода квадратов

println("p-Метод Полларда для числа ", n_pollard)
pollard_factor = Pollard_rho(n_pollard)
println("Нетривиальный делитель: ", pollard_factor)

println("\nМетод квадратов (Теорема Ферма) для числа ", n_fermat)
fermat_factors = Fermat_Factorization(n_fermat)
println("Нетривиальные делители: ", fermat_factors)

println("\nМетод квадратов (Теорема Ферма) для числа ", n_pollard)
fermat_factors = Fermat_Factorization(n_pollard)
println("Нетривиальные делители: ", fermat_factors)
```

Рис. 3.3: Начальные данные для сравнения алгоритмов разложения чисел на множители на Julia



```
PS C:\Users\User\Documents\work\study\2024-2025\Математич
thbase-infosec\labs\lab06\report\report> julia .\lab6.jl
p-Метод Полларда для числа 1359331
Нетривиальный делитель: 1151

Метод квадратов (Теорема Ферма) для числа 15
Нетривиальные делители: (3, 5)

Метод квадратов (Теорема Ферма) для числа 1359331
Нетривиальные делители: (1151, 1181)
```

Рис. 3.4: Результат выполнения кода и сравнения алгоритмов разложения чисел на множители на Julia

4 Выводы

В ходе выполнения лабораторной работы я изучил работу алгоритмов разложения чисел на множители: ρ-Метод Полларда; Метод квадратов (Теорема Ферма о разложении); а также реализовал их программно.

Список литературы

1. Лабораторная работа № 6. Разложение чисел на множители [Электронный ресурс]. Саратовский государственный университет имени Н.Г. Чернышевского, 2024.