РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Отчёт по лабораторной работе №7
Дискретное логарифмирование в конечном поле

Дисциплина: *Математические основы защиты информации и информационной безопасности*

Студент: Агеева Анастасия Сергеевна, 1032212304
Группа: НФИмд-02-21
Преподаватель: Кулябов Дмитрий Сергеевич,
д-р.ф.-м.н., проф.

Москва 2021

Содержание

1 Цель работы **4**

2 Задание **5**

3 Теоретическое введение **6**
3.1 ρ-алгоритм Полларда 6
3.1.1 Современная версия 6
3.2 Алгоритмы для дискретного логарифмирования 7

4 Выполнение лабораторной работы **9**

5 Выводы **11**

Список литературы **12**

List of Figures

4.1 Сжимающая функция f 9
4.2 Результаты p-метода Полларда 10

1 Цель работы

Цель данной лабораторной работы изучение задачи и алгоритмов дискретного логарифмирования в конечном поле.

2 Задание

1. Реализовать программно алгоритм, реализующий p-метод Полларда для задач дискретного логарифмирования.

3 Теоретическое введение

3.1 ρ-алгоритм Полларда

ρ-алгоритм (ρ-алгоритм) — предложенный Джоном Поллардом в 1975 году алгоритм, служащий для факторизации (разложения на множители) целых чисел. Данный алгоритм основывается на алгоритме Флойда поиска длины цикла в последовательности и некоторых следствиях из парадокса дней рождения. Алгоритм наиболее эффективен при факторизации составных чисел с достаточно малыми множителями в разложении [1].

Сложность алгоритма оценивается как $O(N^{1/4})$.

ρ-алгоритм Полларда строит числовую последовательность, элементы которой образуют цикл, начиная с некоторого номера n, что может быть проиллюстрировано, расположением чисел в виде греческой буквы ρ, что послужило названием семейству алгоритмов.

3.1.1 Современная версия

Пусть N составное целое положительное число, которое требуется разложить на множители. Алгоритм выглядит следующим образом: Случайным образом выбирается небольшое число x_0 и строится последовательность $\{x_n\}, n = 0, 1, 2, ...$, определяя каждое следующее как $x_{n+1} = F(x_n) \pmod N$.

Одновременно на каждом i-ом шаге вычисляется $d = \text{GCD}(N, |x_i - x_j|)$ для каких-либо i, j таких, что $j < i$, например, $i = 2j$. Если $d > 1$, то вычисление заканчивается, и найденное на предыдущем шаге число d является делителем

N. Если N/d не является простым числом, то процедуру поиска делителей продолжается, взяв в качестве N число $N' = N/d$.

На практике функция $F(x)$ выбирается не слишком сложной для вычисления (но в то же время не линейным многочленом), при условии того, что она не должна порождать взаимно однозначное отображение. Обычно в качестве $F(x)$ выбираются функции $F(x) = x^2 \pm 1 (\mathrm{mod}\, N)$ или $F(x) = x^2 \pm a (\mathrm{mod}\, N)$. Однако функции $x^2 - 2$ и x^2 не подходят.

Если известно, что для делителя p числа N справедливо $p \equiv 1 \,(\mathrm{mod}\, k)$ при некотором $k > 2$, то имеет смысл использовать $F(x) = x^k + b$.

Существенным недостатком алгоритма в такой реализации является необходимость хранить большое число предыдущих значений x_j.

3.2 Алгоритмы для дискретного логарифмирования

Существуют три различных категории алгоритмов для вычисления дискретных логарифмов [2]:

1. Алгоритмы, которые работают для произвольных групп, т.е. они не используют какие-либо специфические свойства групп. К этой категории относятся метод «шаги младенца – шаги гиганта» Шэнкса, ρ-метод Полларда (аналог метода ρ-факторизации Полларда) и λ-метод (также известный как «дикие и ручные кенгуру»).
2. Алгоритмы, которые хорошо работают в конечных группах, для которых порядок групп не имеет больших простых множителей. Хорошо известный алгоритм Сильвера – Поляга – Хеллмана, основанный на китайской теореме об остатках, относится к этой категории.
3. Алгоритмы, которые используют методы представления групповых элементов как продуктов элементов из относительно небольшого набора (также используя китайскую теорему об остатках); типичными алгоритмами в

этой категории являются алгоритм исчисления индекса Адлемана и алгоритм NFS Гордона.

4 Выполнение лабораторной работы

1. **Реализация p-метода Полларда**

    1. Задам функцию $f()$, обладающую сжимающими свойствами, в которую буду передавать числа c, u и v.

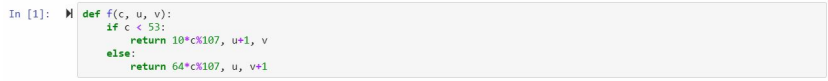

Figure 4.1: Сжимающая функция f

    2. Задам функцию $discrlog()$, в которую буду передавать параметры, необходимые для вычисления . По алгоритму, реализующему p-метода Полларда для задач дискретного логарифмирования, осуществляется нахождение показателя x, для которого верно $a^x \equiv b \pmod p$. В качестве результата возвращается показатель степени x. Вызову функцию для чисел $p = 107$, $a = 10$, $r = 53$, $b = 64$, $u = 2$ и $v = 2$. Алгоритм верно находит показатель степени $x = 20$.

In [3]:
```
def discrlog(p, a, r, b, u, v):
    c = a**u * b**v % p
    d = c
    uc, vc = u, v
    ud, vd = u, v
    c, uc, vc = f(c, uc, vc)
    c %= p
    d, ud, vd = f(*f(d, ud, vd))
    d %= p
    while (c-d)%p != 0:
        c, uc, vc = f(c, uc, vc)
        c %= p
        d, ud, vd = f(*f(d, ud, vd))
        d %= p
    v = vc - vd
    u = ud - uc
    d, x, y = extended_euclid(v, r)
    while d != 1:
        v /= d
        u /= d
        r /= d
        d, x, y = extended_euclid(v, r)
    return x*u%r
```

In [4]:
```
discrlog(107, 10, 53, 64, 2, 2)
```

Out[4]: 20.0

Figure 4.2: Результаты p-метода Полларда

5 Выводы

В ходе данной лабораторной работы я реализовала программно $/rho$-метод Полларда для задач дискретного логарифмирования.

Список литературы

1. Ро-алгоритм Полларда [Электронный ресурс]. Википедия, 2019. URL: https://ru.wikipedia.org/wiki/Ро-алгоритм_Полларда.
2. The discrete log problem [Электронный ресурс]. 2002. URL: http://www.cs.toronto.edu/~cvs/dlog/research_paper.pdf.