Математические основы защиты информации и информационной безопасности. Отчет по лабораторной работе № 7

Дискретное логарифмирование в конечном поле

Лубышева Ярослава Михайловна

Содержание

List of Figures

List of Tables

1 Цель работы

Выполнить задание к лабораторной работе № 7 [1].

2 Задание

1. Ознакомиться с алгоритмом, реализующим р-метод Полларда для задач дискретного логарифмирования.
2. Реализовать алгоритм программно.
3. Проверить правильность работы алгоритма путем вычисления логарифма для заданных значений.

3 Выполнение лабораторной работы

Для реализации алгоритмов вычисления наибольшего общего делителя была написана программа на языке программирования Python (fig. 3.1 - fig. 3.2).

```python
# расширенный алгоритм Евклида для нахождения НОД
# вход - целые числа 0<b<=a
# выход - d=НОД(a,b), целые числа x и y, что a)x+b)y=d
def extended_alg_Euclid(a, b):
  r = [a, b]
  x = [1, 0]
  y = [0, 1]
  i = 1
  while r[i] != 0:
    i += 1
    r.append(r[i-2]%r[i-1])
    if r[i]==0:
      d = r[i-1]
      x = x[i-1]
      y = y[i-1]
    else:
      x.append(x[i-2]-((r[i-2]//r[i-1])*x[i-1]))
      y.append(y[i-2]-((r[i-2]//r[i-1])*y[i-1]))
  return d, x, y
```

Figure 3.1: Программная реализация расширенного алгоритма Евклида для нахождения НОД

```python
# р-метод Полларда для задач дискретного логарифмирования
# вход: простое число p, число a порядка r по модулю p,
# целое число b (1<b<p), отображение f, обладающее сжимающими
# свойствами и сохраняющее вычислимость логарифма,
# u, v - произвольные целые числа
# выход: показатель x, если существует, для которого a^x=b(mod p)
def p_method_Pollard_log(p, a, r, b, u, v, f):
  c = a**u * b**v % p
  d = c
  uc, vc, ud, vd = u, v, u, v

  c, uc, vc = f(c, uc, vc)
  c %= p
  d, ud, vd, = f(*f(d, ud, vd))
  d %= p
  while c%p != d%p:
    c, uc, vc = f(c, uc, vc)
    c %= p
    d, ud, vd, = f(*f(d, ud, vd))
    d %= p

  v, u = vc-vd, ud-uc

  d, x, y = extended_alg_Euclid(v,r)
  while d != 1:
    v /= d
    u /= d
    r /= d
    d, x, y = extended_alg_Euclid(v,r)

  return x*u%r
```

Figure 3.2: Программная реализация р-метода Полларда для задач дискретного логарифмирования

Результаты работы алгоритмов представлены на рисунке ниже (fig. 3.3).

```
p = 107
a = 10
r = 53
b = 64
u = 2
v = 2

def f(c, u, v):
  if c<r:
    return 10*c%107, u+1, v
  else:
    return 64*c%107, u, v+1

print(f"Число x = {p_method_Pollard_log(p, a, r, b, u, v, f)}")
```

```
Число x = 20
```

Figure 3.3: Результаты работы p-метода Полларда для задач дискретного логарифмирования

4 Выводы

Выполнено задание к лабораторной работе № 7.

5 Список литературы

1. Методические материалы курса