

Отчет по лабораторной работе № 7.

Дискретное логарифмирование в конечном поле

**дисциплина: Математические основы защиты информации и
информационной безопасности**

Наливайко Сергей Максимович

Содержание

1	Цель работы	3
2	Задание	4
3	Выполнение лабораторной работы	5
3.1	Реализация алгоритма дискретного логарифмирования	5
4	Выводы	8

1 Цель работы

Познакомиться с дискретным логарифмированием в конечном поле и реализовать алгоритм, реализующий p -метод Полларда.

2 Задание

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

3 Выполнение лабораторной работы

3.1 Реализация алгоритма дискретного логарифмирования

Реализуем алгоритм, реализующий р-метод Полларда для задач дискретного логарифмирования на языке программирования C++.

Код функций данного алгоритма:

```
long long f(long long x, long long& a, long long& b,
    long long alpha, long long beta, long long N, long long r) {
    if (x < r) {
        x = alpha * x % N;
        a = (a + 1) % r;
    } else {
        x = beta * x % N;
        b = (b + 1) % r;
    }
    return x;
}
```

```
long long pollard_method(long long p, long long alpha, long long beta, long long c,
    long long a1, b1, a2, b2, c = 4, d = 4;
a1 = b1 = b2 = a2 = 2;
cout << "i" << setw(6) << "c" << setw(6) << "a1" << setw(6) << "b1"
    << setw(6) << "d" << setw(6) << "a2" << setw(6) << "b2" << "\n";
```

```

for(int i = 1; i < p - 1; ++i) {
    c = f(c, a1, b1, alpha, beta, p, r);
    d = f(f(d, a2, b2, alpha, beta, p, r), a2, b2, alpha, beta, p, r);
    cout << i << setw(6) << c << setw(6) << a1 << setw(6) << b1
        << setw(6) << d << setw(6) << a2 << setw(6) << b2 << "\n";
    if(c == d){
        for(long long j = 1; j < p; ++j) {
            long long tmp = (a1 + b1 * j - a2 - b2 * j) % r;
            if(tmp == 0) {
                cout << "Уравнение: " << a1 << " + " << b1 << "x" <<
                    << a2 << " + " << b2 << "x (mod " << r << ")\n";
                return j;
            }
        }
    }
}
return 0;
}

```

Полный листинг программного кода представлен в файле main.cpp (архив lab07, директория src).

Скомпилируем и запустим программу fig. 3.1.

```
sergey@sergey: ~/University/math_security/lab07/src
File Edit View Search Terminal Help
sergey@sergey:~/University/math_security/lab07/src$ ./main
Введите p: 107
Введите alpha: 10
Введите beta: 64
Введите r: 53
i      c      a1      b1      d      a2      b2
1      40      3       2      79      4       2
2      79      4       2      56      5       3
3      27      4       3      75      5       5
4      56      5       3       3       5       7
5      53      5       4      86      7       7
6      75      5       5      42      8       8
7      92      5       6      23      9       9
8       3       5       7      53     11       9
9      30      6       7      92     11      11
10     86      7       7      30     12      12
11     47      7       8      47     13      13
Уравнение:  $7 + 8x \equiv 13 + 13x \pmod{53}$ 
Показатель x для выражения  $(\alpha^x - \beta) \% p == 0$  равен 20
sergey@sergey:~/University/math_security/lab07/src$
```

Figure 3.1: Дискретное логарифмирование в конечном поле

4 Выводы

В ходе работы мы познакомились с дискретным логарифмированием в конечном поле и реализовали алгоритм, реализующий p -метод Полларда.