Лабораторная работа №7

Математические основы защиты информации и информационной безопасности

Минов Кирилл Вячеславович| НПМмд-02-23

Содержание

1 Цель работы

Реализовать на языке программирования p-метод Полларда для дискретного логарифмирования.

2 Теоретическое введение

Задача дискретного логарифмирования применяется во многих алгоритмах криптографии с открытым ключом. Предложенная в 1976 году У. Дифии и М. Хеллманом для установления сеансового ключа, эта задача послежила основой для создания протоколов шифрования и цифровой подписи, доказательств с нулевым разглашением и других криптографических протоколов.

3 Выполнение лабораторной работы

p-метод Полларда для дискретного логарифмирования реализуем по следующей схеме:

Код программы (рис. 1 - 3).

```java
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 64;
        int p = 107;

        int u = 2;
        int v = 2;
        int r = calculateR(a, p);

        int c = (int) (Math.pow(a, u) * Math.pow(b, v)) % p;
        int d = c;

        int u_c = u;
        int u_d = u;
        int v_c = v;
        int v_d = v;

        System.out.println("  c    |  log_a(c)  |   d   |  log_a(d)");
        System.out.println("-----------------------------------------");

        System.out.printf("c = %d |  %d + %dx |  d = %d |  %d + %dx%n", c, u_c, v_c, d, u_d, v_d);

        int[] resultC = calculateF(c, u_c, v_c, a, b, p);
        int[] resultD = calculateF(calculateF(d, u_d, v_d, a, b, p)[0], calculateF(d, u_d, v_d, a, b, p)[1], calculateF(d, u_d, v_d, a, b, p)[2], a, b, p);

        c = resultC[0];
        u_c = resultC[1];
        v_c = resultC[2];

        d = resultD[0];
        u_d = resultD[1];
        v_d = resultD[2];

        System.out.printf("c = %d |  %d + %dx |  d = %d |  %d + %dx%n", c, u_c, v_c, d, u_d, v_d);
```

Рис. 1: p-метод Полларда для дискретного логарифмирования

```java
        while (c % p != d % p) {
            resultC = calculateF(c, u_c, v_c, a, b, p);
            resultD = calculateF(calculateF(d, u_d, v_d, a, b, p)[0], calculateF(d, u_d, v_d, a, b, p)[1], calculateF(d, u_d, v_d, a, b, p)[2], a, b, p);

            c = resultC[0];
            u_c = resultC[1];
            v_c = resultC[2];

            d = resultD[0];
            u_d = resultD[1];
            v_d = resultD[2];

            System.out.printf("c = %d |  %d + %dx |  d = %d |  %d + %dx%n", c, u_c, v_c, d, u_d, v_d);
        }

        int x = 1;
        while ((u_c + v_c * x) % r != (u_d + v_d * x) % r) {
            x = x + 1;
        }

        System.out.println("\nПоказатель x = " + x);
    }

    private static int calculateR(int a, int p) {
        int r = 1;
        while ((Math.pow(a, r) - 1) % p != 0) {
            r = r + 1;
        }
        return r;
    }
```

Рис. 2: p-метод Полларда для дискретного логарифмирования

```java
    private static int calculateR(int a, int p) {
        int r = 1;
        while ((Math.pow(a, r) - 1) % p != 0) {
            r = r + 1;
        }
        return r;
    }

    private static int[] calculateF(int x, int u, int v, int a, int b, int p) {
        int[] result = new int[3];
        int r = calculateR(a, p);

        if (x < r) {
            result[0] = (int) ((a * x) % p);
            result[1] = u + 1;
            result[2] = v;
        } else {
            result[0] = (int) ((b * x) % p);
            result[1] = u;
            result[2] = v + 1;
        }

        return result;
    }
}
```

Рис. 3: p-метод Полларда для дискретного логарифмирования

4 Выводы

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

Список литературы

1. p-метод Полларда для дискретного логарифмирования [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm_for_logarithms.