Лабораторная работа №7

Дискретное логарифмирование в конечном поле

дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Койфман Кирилл Дмитриевич Группа: НФИмд-01-25

Цель работы Получение практических навыков реализации алгоритмов, решающих задачи дискретного логарифмирования.

Задачи

1. Реализовать алгоритм для задач дискретного логарифмирования.
2. Проверить работоспособность реализованного алгоритма.

---

1 задание

Для решения задачи данной работы была написана программа на C++, реализующая описанный в работе алгоритм:

```

```

---

1 задание

```
namespace functions
{
    long long function_with_compression_behavior_1(long long x, long long
n)
    {
        return static_cast<long long>(std::pow(x, 2) + 5) % n;
    }

    //c is function arg
    long long function_with_compression_behavior_2(long long a, long long
p, long long c)
    {
        return (a * c) % p;
    }

    //c is function arg
    long long function_with_compression_behavior_3(long long b, long long
p, long long c)
    {
```

```
            return (b * c) % p;
        }
    }

    using compressing_function_t = long long(*)(long long, long long);
    using compressing_function_t2 = long long(*)(long long, long long, long
    long);
```

---

2 задание

```
long long algorithmPollardDiscreteLogarifmation(long long p, long long a,
long long gamma, long long b, long long u = 2, long long v = 2,
    compressing_function_t2 function1 =
functions::function_with_compression_behavior_2, compressing_function_t2
function2 = functions::function_with_compression_behavior_3)
{
    long long c = static_cast<long long>(std::pow(a, u) * std::pow(b, v));
    long long d = c;

    double log_c = 0;
    double log_d = 0;

    long long A1 = u, B1 = v;
    long long A2 = u, B2 = v;

    int counter = 10000;
    while (counter >= 0)
    {
        if (c < gamma)
        {
            c = function1(b, p, c) % p;
            d = function1(b, p, d) % p;
            d = function1(b, p, d) % p;
            ++A1;
            A2 += 2;
        }
        else if (c >= gamma)
        {
            c = function2(b, p, c) % p;
            d = function2(b, p, d) % p;
            d = function2(b, p, d) % p;
            ++B1;
            B2 += 2;
        }

        if (c % p == d)
            break;
```

```
            --counter;
        }
        return c;
    }
```

Спасибо за внимание!