# Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники



Лабораторная работа 2.1

Информационная безопасность (Криптография)

Вариант 5

Группа: Р34151

Студент: Дау Конг Туан Ань

Преподаватель: Маркина Татьяна Анатольевна

# Оглавление

1. Задачи	3
Цель работы:	3
Порядок выполнения работы:	3
2. Вариант	3
3. Выполнение	
4. Скриншот работы программы Bcalc.exe	4
5. Рабочий код	5
6. Результат программы	7
7. Заключение	7

## 1. Задачи

Атака на алгоритм шифрования RSA посредством метода Ферма

## Цель работы:

Изучить атаку на алгоритм шифрования RSA посредством метода Ферма.

### Порядок выполнения работы:

- ознакомьтесь с теорией, изложенной в [3]. («Взлом алгоритма RSA при неудачном выборе параметров криптосистемы»);
- получите вариант задания у преподавателя;
- используя разложение модуля на простые числа методом Ферма и полученные исходные данные, определите следующие показатели:
- множители модуля (р и q);
- значение функции Эйлера для данного модуля φ( ) N;
- обратное значение экспоненты по модулю φ() N;
- дешифруйте зашифрованный текст, исходный текст должен быть фразой на русском языке;
- результаты и промежуточные вычисления оформите в виде отчета.

## 2. Вариант

Вариант = 5

Модуль, N = 87046121832829

Экспонента, е = 2342047

Блок зашифрованного текста, С =

#### 3. Выполнение

$$n = [\operatorname{sqrt}(N)] + 1 = 9329852$$

$$t_1 = n + 1 = 9329853$$

$$w_1 = t_1^2 - N = 35168780$$

$$t_2 = n + 2 = 9329854$$

$$w_2 = t_2^2 - N = 53828487$$

$$t_3 = n + 3 = 9329855$$

$$w_3 = t_3^2 - N = 72488196$$

$$sqrt(w_3) = 8514$$

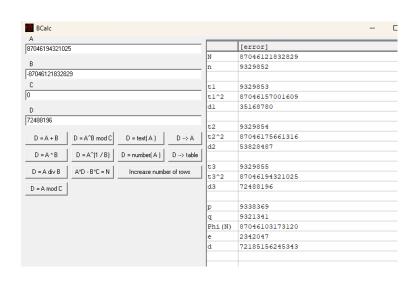
$$p = t_3 + \sqrt{w_3} = 9338369$$

$$q = t_3 - \sqrt{w_3} = 9321341$$

$$\varphi(N) = (p - 1)(q - 1) = 87046103173120$$

## 4. Скриншот работы программы Bcalc.exe

 $d = e^{-1} mod \varphi(N) = 72185156245343$ 



## 5. Рабочий код

#### Link to github

```
public class ModelFerma implements Model{
 private InputOnePair input;
 private BigInteger t;
 private BigInteger w;
 private BigInteger d;
 private String result;
 public ModelFerma(InputOnePair inp) {
   this.input = inp;
   this.result = "";
 }
 private void findTandW() {
    BigInteger N = input.getN();
    BigInteger n = N.sqrt().add(BigInteger.ONE);
   int counter = 1;
   while(true) {
     this.t = n.add(BigInteger.valueOf(counter++));
     this.w = this.t.multiply(this.t).subtract(N);
     if(this.w.sqrt().multiply(this.w.sqrt()).compareTo(this.w) == 0) break;
   }
 }
 private void findD() {
```

```
BigInteger p = this.t.add(this.w.sqrt());
  BigInteger q = this.t.subtract(this.w.sqrt());
  BigInteger phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
  BigInteger x = new BigInteger(String.valueOf(input.getE()));
  BigInteger y = new BigInteger("-1"); // Exponent
  BigInteger z = new BigInteger(String.valueOf(phi));
  this.d = x.modPow(y, z);
}
private void decode() throws UnsupportedEncodingException {
  ArrayList<String> encoded = this.input.getC();
  for(int i = 0; i < encoded.size(); ++i) {
    BigInteger temp = BigInteger
        .valueOf(
           Long.parseLong(
               encoded.get(i)
           )
        ).modPow(
           this.d,
           input.getN()
       );
    String t = new String(temp.toByteArray(), "windows-1251");
    if(t.charAt(0) == 0) t = t.substring(1);
   this.result += t;
  }
```

```
@Override
public void solve() {
   try{
     findTandW();
     findD();
     decode();
   } catch (UnsupportedEncodingException e) {
   }
}

public String getResult() {
   return this.result;
}
```

# 6. Результат программы

message = routes). Одно-маршрутный пакет появляется только

## 7. Заключение

В ходе лабораторной работы я изучил алгоритм атаки RSA с помощью метода Ферма, а также способы его реализации на языке Java.