

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

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники



Лабораторная работа 2.3

Информационная безопасность (Криптография)

Вариант 5

Группа: Р34151

Студент: Дау Конг Туан Ань

Преподаватель: [Маркина Татьяна Анатольевна](#)

г. Санкт-Петербург, 2024

## Оглавление

1. Задачи .....	3
Цель работы: .....	3
Порядок выполнения работы: .....	3
2. Вариант .....	3
3. Рабочий код .....	4
4. Результат программы .....	5
5. Заключение .....	6

## 1. Задачи

Атака на алгоритм шифрования RSA методом бесключевого чтения

### Цель работы:

Изучить атаку на алгоритм шифрования RSA посредством метода бесключевого чтения.

### Порядок выполнения работы:

- ознакомьтесь с теорией в [3], в подразделе («Бесключевое чтение»);
- получите вариант задания у преподавателя; – по полученным данным определите значения  $r$  и  $s$  при условии, чтобы  $e_1 \cdot r - e_2 \cdot s = 1$ . Для этого необходимо использовать расширенный алгоритм Евклида;
- используя полученные выше значения  $r$  и  $s$ , запишите исходный текст;
- результаты и промежуточные вычисления значений для любых трех блоков шифрованного текста оформите в виде отчета.

Примечание. Для выполнения практического задания рекомендуется использовать программу BCalc.exe.

## 2. Вариант

Вариант = 5

Модуль,  $N = 572953270159$

Экспонента,  $e_1 = 337903$

Экспонента,  $e_2 = 301933$

Блок зашифрованного текста,  $C_1 =$

342095517391  
19455909955  
221503536026  
316042040322  
311339725976  
339044089754  
359623172126  
138544673544  
148226083413  
3486028632  
23290754913  
425720995382

Блок зашифрованного текста, C2 =

32476529608  
452342848743  
506694128118  
262070340689  
206245109461  
116518622136  
147952236274  
457665805346  
27001690429  
396682057113  
239803556225  
519526641494

### 3. Рабочий код

[Link to github](#)

```
public class ModelreEnc implements Model{

    private InputOnePair input;

    private String result;

    public ModelreEnc(InputOnePair inp) {

        this.result = "";

        this.input = inp;

    }

    @Override

    public void solve() {

        try{

            for(String stri : input.getC()) {

                BigInteger y = BigInteger.valueOf(Long.parseLong(stri));

                BigInteger yi = y.modPow(input.getE(), input.getN());

                BigInteger res = BigInteger.ZERO;

                while(y.compareTo(yi) != 0) {
```

```

        res = yi;

        yi = yi.modPow(input.getE(), input.getN());
    }

    String temp = new String(res.toByteArray(), "windows-1251");

    if(temp.charAt(0) == 0) temp = temp.substring(1);

    this.result += temp.substring(0, temp.length());

}

} catch(UnsupportedEncodingException e) {}

}

public String getResult() {

    return this.result;

}

}

```

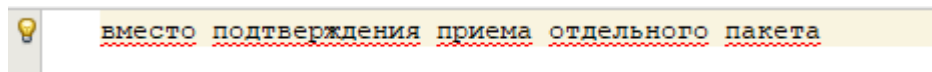
## 4. Результат программы

```

j --- 572953270159 (MESSAGE) 0 JA ---
337903
301933
[342095517391, 19455909955, 221503536026, 316042040322, 311339725976, 339044089754, 359623172126, 138544673544, 148226083413, 3486028632, 23290754913, 425720995382]
[32476529608, 452342848743, 506694128118, 262070340689, 206245109461, 116518622136, 147952236274, 457665805346, 27001690429, 396682057113, 239803556225, 519526641494]

r: 21346 s: -23889
Decode C[0]: 
Decode C[1]: 
Decode C[2]: 
Decode C[3]: 
Decode C[4]: 
Decode C[5]: 
Decode C[6]: 
Decode C[7]: 
Decode C[8]: 
Decode C[9]: 
Decode C[10]: 
Decode C[11]: 
- Text written to file successfully.

```



**message** = вместо подтверждения приема отдельного пакета

## 5. Заключение

В ходе выполнения данной лабораторной работы я ознакомился с методом бесключевого чтения для атаки на алгоритм шифрования RSA.