**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет программной инженерии и компьютерной техники

Дисциплина «Информационная безопасность»

**Лабораторная работа №2.2**

**«Атака на алгоритм шифрования RSA методом повторного шифрования»**

Вариант: 4

**Учебно-методическое пособие:** Криптографические системы с секретным и открытым

ключом: учебное пособие. / А.А. Ожиганов; УНИВЕРСИТЕТ ИТМО. — Санкт-Петербург, 2015

**Автор**: Калинин Даниил Дмитриевич

**Группа**: P34141

**Преподаватель**: Маркина Татьяна Анатольевна

г. Санкт-Петербург

2024

# Содержание

[**Содержание 2**](#_1j29sb1plksc)

[**Цель работы 2**](#_ic6vz9o4aev6)

[**Порядок выполнения работы 2**](#_7c966s7zbuo)

[**Вариант 3**](#_tk58wqserqjy)

[**Выполнение работы 4**](#_i73wvolbwe0b)

[Код 4](#_opx7en3yhfv9)

[Результаты работы программы 6](#_h1i0yqxqtni2)

[**Вывод 6**](#_uft2d2ek0ay3)

# Цель работы

Изучить атаку на алгоритм шифрования RSA посредством повторного шифрования.

# Порядок выполнения работы

* Ознакомьтесь с теорией в [3], рассмотренной в подразделе («Атака повторным шифрованием»);
* Получите вариант задания у преподавателя;
* По полученным исходным данным, используя метод перешифрования, определите порядок числа *e* в конечном поле
* Используя значение порядка экспоненты, получите исходный текст методом перешифрования;
* Результаты и промежуточные вычисления оформите в виде отчета.

# Вариант

|  |  |  |  |
| --- | --- | --- | --- |
| **Вариант** | **Модуль, *N*** | **Экспонента, *е*** | **Блок зашифрованного текста, *C*** |
| 4 | 489740760623 | 892627 | 237434928568  89382477865  257542914775  153947910848  219678068406  166466311168  49516725114  55375254449  370796045103  322927050068  196366079994  39243100230  299525662956 |

# 

# 

# 

# 

# 

# 

# Выполнение работы

## Код

def re\_encryption\_method(N, e, C):

*"""Определение порядка числа e"""*

print("-- Метод повторного шифрования --")

*# Выбираем первую часть, на которой будем искать степерь повтороного кодирования*

raw\_parts = C.split("\n")

y = 0

for i in range(len(raw\_parts)):

if raw\_parts[i].strip() != "":

y = int(raw\_parts[i].strip())

break

*# Выполняем повторное шифрование*

y\_i = y

i = 1

while True:

y\_i = pow(y\_i, e, N)

i+=1

if (y\_i == y):

break

print(f'y\_{i} = y = {y\_i}')

return i - 2 *#Порядок числа e*

def decode(N, e, C, m):

*"""Декодирует полученное сообщение в текст"""*

print("-- Дешифрование сообщения --")

*# Раздаляем закодированное сообщение на части и подготавливаем их*

raw\_parts = C.split("\n")

parts = []

for i in range(len(raw\_parts)):

if raw\_parts[i].strip() != "":

parts.append(int(raw\_parts[i].strip()))

*# Декодируем каждую часть*

original\_message = ""

for part in parts:

int\_decoded\_part = pow(part, pow(e, m), N)

decoded\_part = int\_decoded\_part.to\_bytes(4, byteorder='big').decode('cp1251')

original\_message += decoded\_part

print(f'Декодирована часть {part} -----> y\_{m+1} = {int\_decoded\_part} -----> {decoded\_part}')

return original\_message

if \_\_name\_\_ == '\_\_main\_\_':

*# Описание варианта*

N = 489740760623

e = 892627

C = """

237434928568

89382477865

257542914775

153947910848

219678068406

166466311168

49516725114

55375254449

370796045103

322927050068

196366079994

39243100230

299525662956

"""

print("-- Исходные данные --")

print(f'N = {N}')

print(f'e = {e}')

print(f'C = \"{C}\"')

print()

*# Определяем порядок числа e*

m = re\_encryption\_method(N, e, C)

print(f'x = y\_{m + 1}')

print(f'Порядок числа e = {m}')

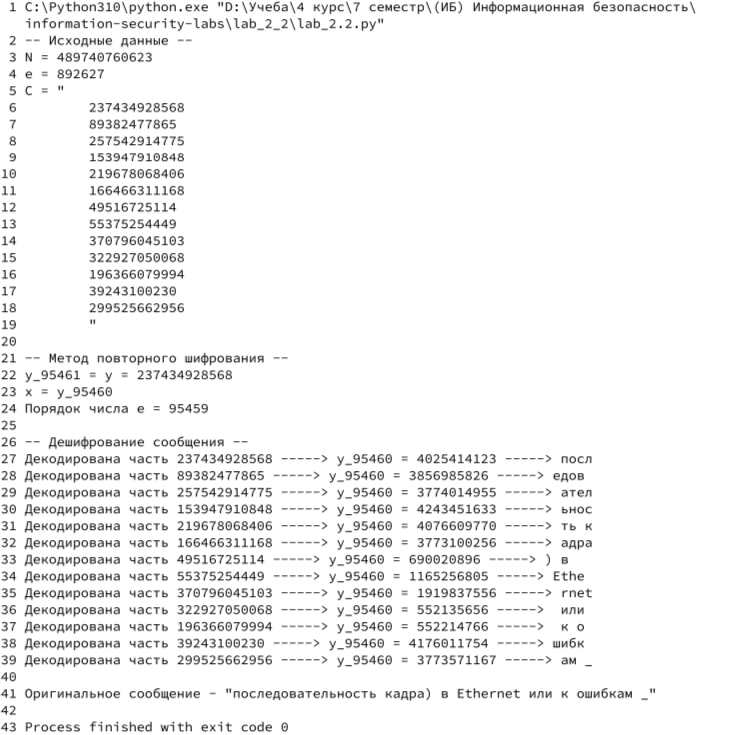
print()

*# Декодируем сообщение*

original\_message = decode(N, e, C, m)

print(f'\nОригинальное сообщение - \"{original\_message}\"')

## Результаты работы программы



# Вывод

В ходе лабораторной работы была совершена атака на алгоритм шифрования RSA посредством повторного шифрования, в следствии чего было декодировано исходное сообщение. В процессе выполнения был изучен алгоритм совершения атаки на алгоритм шифрования RSA посредством повторного шифрования.