Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Основы криптографии с открытым ключом. Алгоритм RSA (вариант 3)

Руководитель от университета \_\_\_\_\_\_\_\_\_\_ Сидарас А.А.

подпись, дата

Студент КИ19-06б, 031940417 \_\_\_\_\_\_\_\_\_\_ Шнайдер А.В.

подпись, дата

Красноярск 2020

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc69410691)

[Ход выполнения работы 4](#_Toc69410692)

[1. Подробное описание алгоритма шифрования 4](#_Toc69410693)

[2. Листинг составленной программы 4](#_Toc69410694)

[3. Контрольные примеры работы программы 9](#_Toc69410695)

[ЗАКЛЮЧЕНИЕ 12](#_Toc69410696)

ВВЕДЕНИЕ

Задание к работе: согласно варианту разработать алгоритм шифрования и расшифрования RSA и составить программу, реализующую данный алгоритм. На ряде контрольных примеров открытого текста, состоящего из различного количества символов, проверить правильность работы алгоритмов шифрования и дешифрования.

Ход выполнения работы

1. Подробное описание алгоритма шифрования

Пусть абонент А хочет передать зашифрованное сообщение абоненту Б. В

этом случае абонент Б должен подготовить пару ключей (открытый и закрытый и

отправить свой открытый ключ пользователю А.

Первым этапом является генерация открытого и закрытого ключей. Для этого

последовательно проделываются следующие действия:

1. выбираются два больших и простых числа Р и Q;

2. вычисляется произведение N = P \*Q;

3. вычисляется значение функции Эйлера φ(N) = (P-1) \* (Q-1) = d;

4. случайным образом выбирается число s < d и взаимно простое с d;

5. вычисляется число e, такое, что e\*s mod d = 1.

После всех этих операций, числа s и N будут открытым ключом пользователя, а число e – закрытым ключом.

Так как пользователь Б хочет получить зашифрованное сообщение от пользователя А, значит пользователь Б должен отправить свой открытый ключ (s, N) пользователю А. Числа Р и Q больше не нужны, но при этом они остаются уязвимым местом алгоритма и поэтому подлежат удалению.

2. Листинг составленной программы

import random

from math import gcd

from math import sqrt

from sympy import isprime

def generate\_key\_pair(p, q):

if not (isprime(p) and isprime(q)):

raise ValueError('Числа должны быть простыми!')

elif p == q:

raise ValueError('Числа не должны быть одинаковыми!')

n = p \* q

phi = (p-1) \* (q-1)

e = random.randrange(1, phi)

g = gcd(e, phi)

while g != 1:

e = random.randrange(1, phi)

g = gcd(e, phi)

d = multiplicative\_inverse(e, phi)

return ((e, n), (d, n))

def multiplicative\_inverse(e, phi):

d = 0

x1 = 0

x2 = 1

y1 = 1

temp\_phi = phi

while e > 0:

temp1 = temp\_phi // e

temp2 = temp\_phi - temp1 \* e

temp\_phi = e

e = temp2

x = x2 - temp1 \* x1

y = d - temp1 \* y1

x2 = x1

x1 = x

d = y1

y1 = y

if temp\_phi == 1:

return d + phi

def encrypt(pk, plaintext):

key, n = pk

cipher = [pow(ord(char), key, n) for char in plaintext]

return cipher

def decrypt(pk, ciphertext):

key, n = pk

dba = [str(pow(char, key, n)) for char in ciphertext]

plain = [chr(int(char2)) for char2 in dba]

return ''.join(plain)

if \_\_name\_\_ == "\_\_main\_\_":

print("RSA crypt: ")

p = int(random.randrange(1, 99999999999999999999999999999999999999999999))

q = int(random.randrange(round(10000000000000000000000000000000000000000000/p), round(99999999999999999999999999999999999999999999/p)))

while not isprime(p):

p = int(random.randrange(1, 99999999999999999999999999999999999999999999))

while not isprime(q):

q = int(random.randrange(round(10000000000000000000000000000000000000000000/p), round(99999999999999999999999999999999999999999999/p)))

public, private = generate\_key\_pair(p, q)

print("Public key: ", public)

print("Private key: ", private)

message = input("Введите сообщение: ")

encrypted\_msg = encrypt(public, message)

print("Зашифрованное сообщение: ", ''.join(map(lambda x: str(x) + " ", encrypted\_msg)))

print("Расшифрованное сообщение: ", decrypt(private, encrypted\_msg))

**3. Контрольные примеры работы программы**

Примеры работы представлены на рисунках 1 – 10.



Рисунок 1 – Зашифровка и расшифровка строки “My name is Sasha. Меня зовут Саша. 0123456789()”



Рисунок 2 – Зашифровка и расшифровка строки “Какие вкусные багеты вы печёте! I like it. ...-\-\_-/-”



Рисунок 3 – Зашифровка и расшифровка строки “АААААААААААААААААААААААААААААААААААААААААФФФФФФФФФФФФФ”

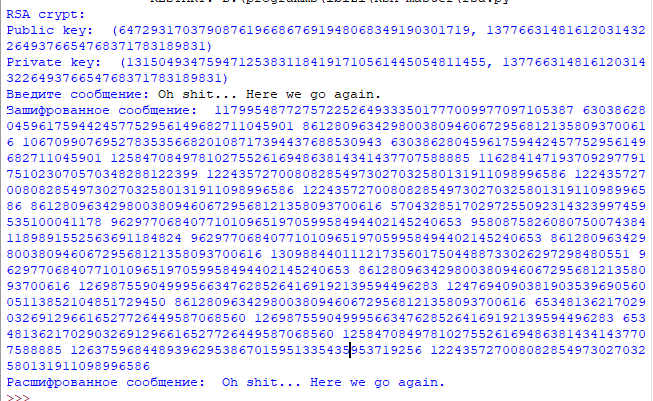


Рисунок 4 – Зашифровка и расшифровка строки “Oh shit... Here we go again.”



Рисунок 5 - Зашифровка и расшифровка строки “ aiuvbasnvansvonasvobxcvфмтщжфтмщфтмщтфмджтфмиймщ(Г№к9098е7983”

ЗАКЛЮЧЕНИЕ

В процессе выполнения данной лабораторной работы я познакомился с алгоритмом шифрования RSA, а также научился его использовать для составления программ шифраторов-дешифраторов.