Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра прикладной математики и кибернетики

Расчетно-графическая работа  
по дисциплине «Защита информации»  
по теме «Доказательство с нулевым знанием»

Выполнил:  
студент группы ИП-014

Обухов А.И.

Работу проверил:  
ст. преп. каф. ПМиК  
Дьячкова И.С.

Новосибирск 2023 г.

Оглавление

[Задание 3](#_Toc151559539)

[Результат выполнения работы 5](#_Toc151559540)

[Листинг 6](#_Toc151559541)

# Задание

Это задание выполняется по вариантам, в зависимости от номера студента в журнале. Для определения номера варианта необходимо взять номер студента в журнале, вычислить его остаток от деления на 3 и прибавить 1.

Варианты задания:

1. Необходимо написать программу, реализующую протокол доказательства с нулевым знанием для задачи «Раскраска графа».
2. Необходимо написать программу, реализующую протокол доказательства с нулевым знанием для задачи «Гамильтонов цикл».
3. Необходимо написать программу, реализующую протокол доказательства с нулевым знанием Фиата-Шамира.

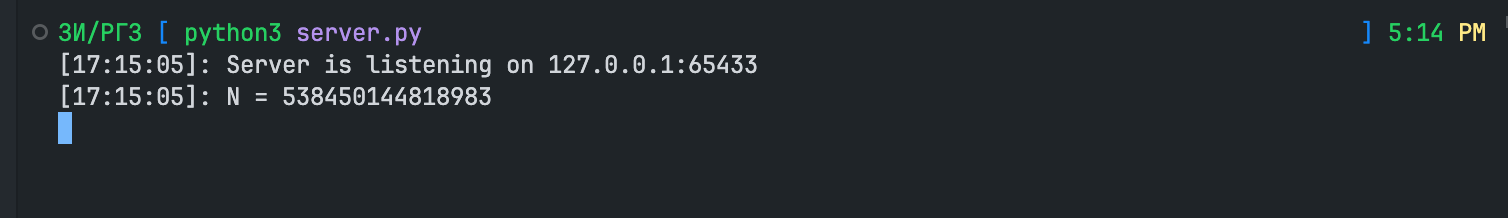
**Протокол Фиата-Шамира**

Для выполнения этого варианта задания необходимо разработать клиент-серверное приложение с авторизацией по протоколу Фиата-Шамира. Открытые ключи с соответствующими логинами должны храниться в файле (или базе данных) на сервере, клиентское приложение при этом не должно отправлять на сервер никаких закрытых данных, закрытый ключ нигде не хранится и используется исключительно для осуществления работы протокола с клиентской стороны. Все открытые параметры системы рассылаются сервером при установке соединения с клиентом.

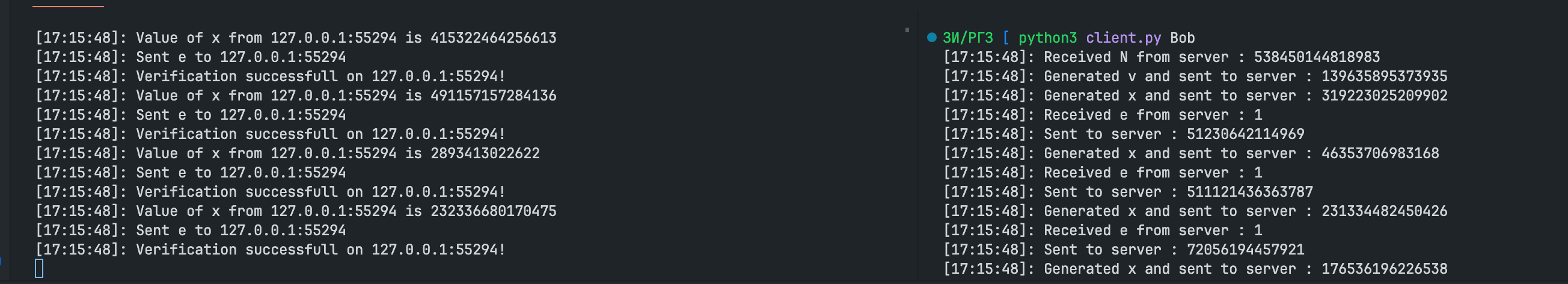
**Общие рекомендации**

Программа должна наглядно демонстрировать работу алгоритма, возможно (но не обязательно) в графическом режиме. Текст программы должен содержать исчерпывающие комментарии, тем не менее, следует воздержаться от описания очевидных действий. К РГР необходимо представить отчёт, оформленный в соответствии с требованиями, предъявляемыми к работам подобного типа. В отчёт должны быть включены в обязательном порядке: Титульный лист, содержание, постановка задачи, исходный код с комментариями, скриншоты с результатами работы программы.

# Результат выполнения работы

  
Рис.1

На рисунке 1 изображен запуск сервера

  
Рис.2

На рисунке 2 изображен результат работы успешной аутентификации пользователя по протоколу Фиата-Шамира.

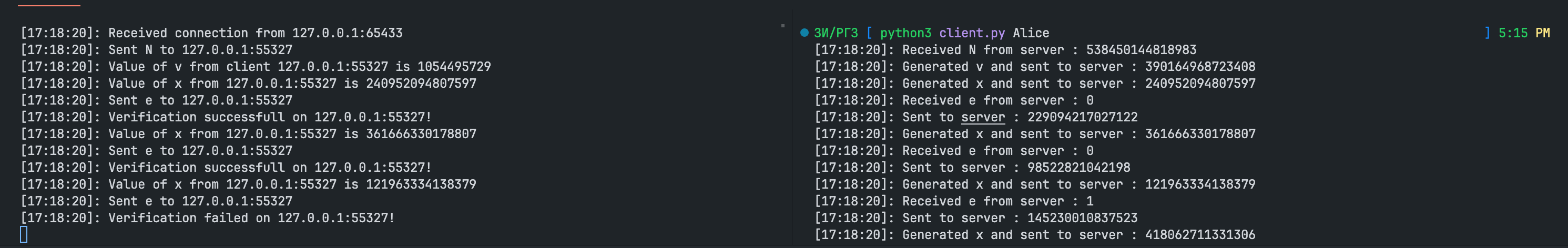


Рис.3

На рисунке 3 изображен результат при подмене секретного параметра клиентом (попытка обмануть сервер)

# Листинг

# server.py

import socket

import threading

import logging

import utils

import random

import signal

DATABASE\_NAME = 'db.json'

database = {}

server\_socket = None

def signal\_handler(sig, frame):

utils.save\_json(database, DATABASE\_NAME)

server\_socket.close()

exit()

def client\_thread(connection, address):

address = utils.format\_address(address)

logging.info(f'Sent N to {address}')

username = connection.recv(utils.BUFFER\_SIZE).decode()

connection.sendall(str(N).encode())

v = int(connection.recv(utils.BUFFER\_SIZE).decode())

if username not in database:

database[username] = v

else:

v = database[username]

logging.info(f'Value of v from client {address} is {v}')

for i in range(utils.ROUNDS):

x = int(connection.recv(utils.BUFFER\_SIZE).decode())

logging.info(f'Value of x from {address} is {x}')

e = random.getrandbits(1)

connection.sendall(str(e).encode())

logging.info(f'Sent e to {address}')

y = int(connection.recv(utils.BUFFER\_SIZE).decode())

if y == 0:

logging.error(f'Verification failed on {address}')

connection.close()

return

if y \*\* 2 % N == (x \* v\*\*e) % N:

logging.info(f'Verification successfull on {address}!')

else:

logging.error(f'Verification failed on {address}!')

connection.close()

return

database = utils.load\_json(DATABASE\_NAME)

if 'N' not in database:

p = utils.generate\_prime(10\*\*2, 10\*\*6)

q = utils.generate\_prime(10\*\*2, 10\*\*6)

N = p \* q

database['N'] = N

N = database['N']

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

signal.signal(signal.SIGINT, signal\_handler)

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.bind((utils.HOST, utils.PORT))

server\_socket.listen()

logging.info(f"Server is listening on {utils.format\_address(server\_socket.getsockname())}")

logging.info(f"N = {N}")

while True:

connection, address = server\_socket.accept()

logging.info(f"Received connection from {utils.format\_address(server\_socket.getsockname())}")

threading.Thread(target=client\_thread, args=(connection, address)).start()

# client.py

import socket

import utils

import logging

import random

import sys

username = sys.argv[1]

users\_s = utils.load\_json('users\_s.json')

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

connection = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

connection.connect((utils.HOST, utils.PORT))

connection.sendall(username.encode())

N = int(connection.recv(utils.BUFFER\_SIZE).decode())

logging.info(f'Received N from server : {N}')

s = utils.generate\_coprime(N)

if username not in users\_s:

users\_s[username] = s

utils.save\_json(users\_s, 'users\_s.json')

else:

s = users\_s[username]

v = (s \*\* 2) % N

connection.sendall(str(v).encode())

logging.info(f'Generated v and sent to server : {v}')

try:

for i in range(utils.ROUNDS):

r = random.randint(1, N)

x = (r \*\* 2) % N

connection.sendall(str(x).encode())

logging.info(f'Generated x and sent to server : {x}')

e = int(connection.recv(utils.BUFFER\_SIZE).decode())

logging.info(f'Received e from server : {e}')

if e == 0:

y = r

else:

y = (r \* s \*\* e) % N

connection.sendall(str(y).encode())

logging.info(f'Sent to server : {y}')

logging.info('Verification successfull')

except:

logging.error('Verification failed')

connection.close()

exit()