МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил\_\_:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 2023

Ход работы

*Код программы*

import random

import hashlib

import math

class RabinSignature:

def load\_primes(self, filename):

primes = []

with open(filename, 'r') as file:

primes = list(map(int, file.read().split()))

return primes

def get\_random\_string(self, size):

res = ""

for i in range(size):

res += str((random.randint(0, 1)))

return res

def get\_hash\_value(self, s):

return int.from\_bytes(hashlib.sha256(s.encode()).digest(), 'big')

def is\_quadratic\_residue(self, val, pr):

return 1 == pow(val, (pr - 1) // 2, pr)

def get\_prime\_candidate(self, bits):

prime\_candidate = random.randint(2 \*\* (bits - 1), (2 \*\* bits) - 1)

if (prime\_candidate % 2 == 0):

prime\_candidate += 1

return prime\_candidate

def witness(self, a, k, m, p):

b = pow(a, m, p)

if b == 1:

return False

for i in range(k):

if ((-1 % p) == (b % p)):

return False

b = (b \* b) % p

return True

def initial\_check(self, prime\_candidate):

for prime in self.primes:

if prime\_candidate % prime == 0:

return False

return True

def miller\_rabin\_test(self, prime\_candidate):

if not self.initial\_check(prime\_candidate):

return False

temp = prime\_candidate - 1

k = 0

while temp % 2 == 0:

temp //= 2

k += 1

m = temp

for j in range(70):

a = random.randint(2, prime\_candidate - 2)

if self.witness(a, k, m, prime\_candidate):

return False

return True

def get\_prime(self, bits):

prime\_candidate = self.get\_prime\_candidate(bits)

while not self.miller\_rabin\_test(prime\_candidate):

prime\_candidate = self.get\_prime\_candidate(bits)

return prime\_candidate

def \_\_init\_\_(self, k):

self.primes = self.load\_primes('primes.txt')

self.k = k

def get\_public\_keys(self):

p = self.get\_prime(self.k)

q = self.get\_prime(self.k)

return (p, q)

def get\_signature(self, M, p, q):

n = p \* q

b = 10 \*\* 9 + 7

d = b \* pow(2, (p - 1) \* (q - 1) - 1, n)

U = self.get\_random\_string(60)

c = self.get\_hash\_value(M + U)

m = c + d \* d

while (not (self.is\_quadratic\_residue(m, p) and self.is\_quadratic\_residue(m, q))):

U = self.get\_random\_string(60)

c = self.get\_hash\_value(M + U)

m = c + d \* d

v1 = pow(m, (p + 1) // 4, p) \* q \* pow(q, p - 2, p)

v2 = pow(m, (q + 1) // 4, q) \* p \* pow(p, q - 2, q)

y = (v1 + v2) % n

x = (y - d) % n

return (U, x)

def verify(self, p, q, U, x, M):

n = p \* q

b = 10 \*\* 9 + 7

c = self.get\_hash\_value(M + U)

l\_side = x \* (x + b)

r\_side = c

if (l\_side % n == r\_side % n):

return 1

return 0

*Вывод программы*

*Сервер*

Сервер запущен. Ожидание подключения...  
Создание объекта цифровой подписи...  
Генерация открытого ключа p и q...  
Подключено к ('127.0.0.1', 53952)

Генерация подписи U, x...  
msg = 0100100001100101011011000110110001101111001000000101011101101111011100100110110001100100  
p = 173418224154056534435458786449777749159052369137042987412130087731237535368804969390706118525979920542318651066102445430990449875221098214580830225361953923119652638755559787657902124815138087706007333563071625788898451122566966976223520672103476238464951784844995202716216107313658865274666271028440378312667  
q = 175706186013739390042749844048478417727141916902618410252875965569719099326107933946604834292344518242149060768598657434554890968067509315157119817055369004623590509673563303609881628295284168831911845860276323986234815068571461804365030365643210459981011080895276265353345682076812491397097081774062796840811  
U = 100111101110001101010111001010000101110111011000001101010110  
x = 2315727991893490136232988849655429034863304055707911737232632971307905845379630768508666731531620343660400581774654485275729502636909069017434652880629381698807153658781623719331241654167695981543791815712449488482315096134261808216482928481102990091911167436391247638497946599275977795159207116136258797100123872772701813773012361184102092914754412116110403967196859070443324578146024134227609472898966898938930976271541065466992053868820695733494177496996101535674147010087424234479690410213614429198864399461554885811659026732287293564618078480515184961759396237469690955243118743123965420267177057339835539968025  
sinature.verify = 1  
Отправка сообщения и ключей p, q, U, x для проверки: Hello World, 173418224154056534435458786449777749159052369137042987412130087731237535368804969390706118525979920542318651066102445430990449875221098214580830225361953923119652638755559787657902124815138087706007333563071625788898451122566966976223520672103476238464951784844995202716216107313658865274666271028440378312667, 175706186013739390042749844048478417727141916902618410252875965569719099326107933946604834292344518242149060768598657434554890968067509315157119817055369004623590509673563303609881628295284168831911845860276323986234815068571461804365030365643210459981011080895276265353345682076812491397097081774062796840811, 100111101110001101010111001010000101110111011000001101010110, 2315727991893490136232988849655429034863304055707911737232632971307905845379630768508666731531620343660400581774654485275729502636909069017434652880629381698807153658781623719331241654167695981543791815712449488482315096134261808216482928481102990091911167436391247638497946599275977795159207116136258797100123872772701813773012361184102092914754412116110403967196859070443324578146024134227609472898966898938930976271541065466992053868820695733494177496996101535674147010087424234479690410213614429198864399461554885811659026732287293564618078480515184961759396237469690955243118743123965420267177057339835539968025

*Клиент*

Соединение успешно установлено  
Получено сообщение: 0100100001100101011011000110110001101111001000000101011101101111011100100110110001100100Получено p: 173418224154056534435458786449777749159052369137042987412130087731237535368804969390706118525979920542318651066102445430990449875221098214580830225361953923119652638755559787657902124815138087706007333563071625788898451122566966976223520672103476238464951784844995202716216107313658865274666271028440378312667Получено q: 175706186013739390042749844048478417727141916902618410252875965569719099326107933946604834292344518242149060768598657434554890968067509315157119817055369004623590509673563303609881628295284168831911845860276323986234815068571461804365030365643210459981011080895276265353345682076812491397097081774062796840811Получено U: 100111101110001101010111001010000101110111011000001101010110Получено x: 2315727991893490136232988849655429034863304055707911737232632971307905845379630768508666731531620343660400581774654485275729502636909069017434652880629381698807153658781623719331241654167695981543791815712449488482315096134261808216482928481102990091911167436391247638497946599275977795159207116136258797100123872772701813773012361184102092914754412116110403967196859070443324578146024134227609472898966898938930976271541065466992053868820695733494177496996101535674147010087424234479690410213614429198864399461554885811659026732287293564618078480515184961759396237469690955243118743123965420267177057339835539968025Проверка подписи sinature.verify(p, q, U, x) = 1