МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ компьютерной безопасности и криптографии

**Скрытый канал связи на основе ESIGN**

ОТЧЁТ

ПО ДИСЦИПЛИНЕ

«КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ»

студента 5 курса 531 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Норикова Павла Сергеевича

|  |  |  |
| --- | --- | --- |
| Преподаватель  аспирант | \_\_\_\_\_\_\_\_\_\_ | Р. А. Фарахутдинов |
|  | подпись, дата |  |

Саратов 2023

**ВВЕДЕНИЕ**

Цель работы – изучение и реализация скрытого канала связи на основе ESIGN.

**1 Теория**

* 1. **Генерация основных параметров**

Алиса подписывает безобидное сообщение , а вместе с ним передается секретное сообщение . , , – большие простые числа одинаковой битовой длины (). . Выбирается параметр безопасности . , – открытый ключ, , , – закрытый ключ Алисы. известно Бобу.

* 1. **Подпись сообщения**

1. , где – хэш-функция со значением от 0 до .
2. , где и – случайное число от 0 до .
3. , где , .
4. , где .
5. .
   1. **Проверка подписи**
6. , где .
7. , где .
8. проверяет, что .
9. .
   1. **Получение секретного сообщения**
10. Боб также проверяет подпись и извлекает секретное сообщение .

**2 Практическая реализация**

**2.1 Описание программы**

Функция *gen\_q* генерирует большое простое число заданной длины.

Функция *gcd\_ex* реализует расширенный алгоритм Евклида.

На вход программе подается длина *L* чисел *p*, *q*, *r*, сообщение *m*, секретное сообщение и параметр безопасности *k*.

**2.2 Тестирование программы**

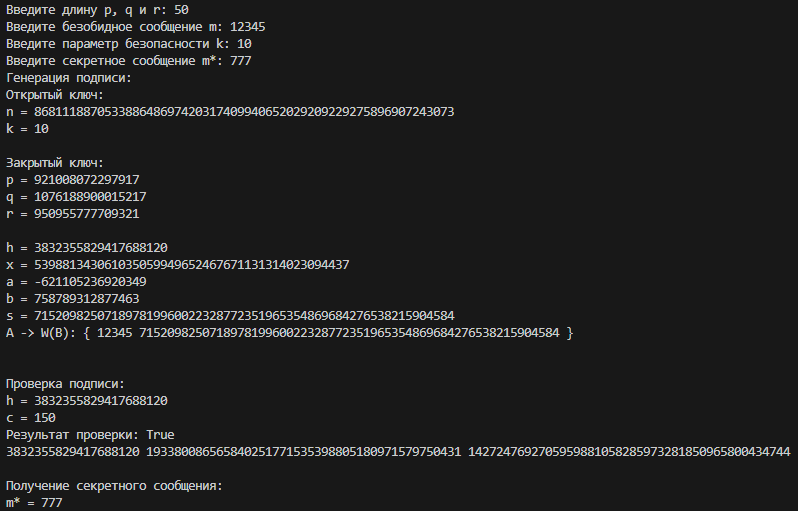


Рисунок 1 - Работа протокола

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

#import numpy

import random

import math

from sympy import isprime

def gcd\_ex(a, b):

if b == 0:

return a, 1, 0

else:

gcd, x1, y1 = gcd\_ex(b, a % b)

x = y1

y = x1 - (a // b) \* y1

return gcd, x, y

def gen\_q(L):

res = "0"

while not isprime(int(res, 2)):

res = ""

for i in range (1, L - 1):

random.seed()

res += str(random.randint(0,100)%2)

res = '1' + res + '1'

return int(res, 2)

def main(L, m, k, m\_):

print('Генерация подписи:')

p = gen\_q(L)

q = gen\_q(L)

r = gen\_q(L)

n = p\*p\*q\*r

print('Открытый ключ:', '\nn =', n, '\nk =', k, '\n')

print('Закрытый ключ:', '\np =', p, '\nq =', q, '\nr =', r, '\n')

h = abs(int(hash(str(m)))) % n

print('h =', h)

u = random.randint(1, p\*q - 2)

x = m\_ + u\*r

print('x =', x)

w = math.ceil((h - pow(x, k, n)) / p\*q\*r)

#a = math.ceil((h - pow(x, k, n)) / (p\*q\*r)) % p

a = math.ceil((h - pow(x, k, n)) / (p\*q\*r))

print('a =', a)

b = (a \* gcd\_ex(k\*pow(x, k-1, p) % p, p)[1]) % p

print('b =', b)

s = x + b\*p\*q\*r

#s = x + ((w \* gcd\_ex(k \* pow(x, k-1, p), p)[1]) % p) \* p \* q \* r

print('s =', s)

#print('s2 =', s2)

print('A -> W(B): {', m, s, '}\n')

print('\nПроверка подписи:')

h2 = abs(int(hash(str(m)))) % n

print('h =', h2)

c = math.ceil((3 \* math.log2(n)) / 4)

#c = math.ceil((2 \* math.log2(n)) / 3)

#c= math.ceil(math.log2(n))

print('c =', c)

print('Результат проверки:', h2 <= pow(s, k, n) and pow(s, k, n) <= h2 + pow(2, c))

print(h2, pow(s, k, n), h2 + pow(2, c))

print('\nПолучение секретного сообщения:')

B\_m\_ = s % r

print('m\* =', B\_m\_)

L = int(input('Введите длину p, q и r: '))

m = int(input('Введите безобидное сообщение m: '))

k = L\_q = int(input('Введите параметр безопасности k: '))

m\_= int(input('Введите секретное сообщение m\*: '))

main(L, m, k, m\_)

#main(30, 9999, 5, 321)