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

1.1 Генерация основных параметров

Алиса подписывает безобидное сообщение m, а вместе с ним передается секретное сообщение m^*. p, q, r – большие простые числа одинаковой битовой длины ($m^* < r$). $n = p^2qr$. Выбирается параметр безопасности $k \geq 4$. n, k – открытый ключ, p, q, r – закрытый ключ Алисы. r известно Бобу.

1.2 Подпись сообщения

1. A: $\{h\}$, где $h = H(m)$ – хэш-функция со значением от 0 до $n - 1$.
2. A: $\{x\}$, где $x = m^* + ur$ и u – случайное число от 0 до $pq - 1$.
3. A: $\{a, b\}$, где $a = \left\lceil \frac{h - (x^k \bmod n)}{pqr} \right\rceil$, $b = a(kx^{k-1})^{-1} mod\ p$.
4. A: $\{s\}$, где $s = x + bpqr$.
5. $A \rightarrow W(B)$: $\{m, s\}$.

1.3 Проверка подписи

1. W: $\{h\}$, где $h = H(m)$.
2. W: $\{c\}$, где $c = \left\lceil \frac{3}{4} \log_2 n \right\rceil$.
3. W: проверяет, что $h \leq s^k\ mod\ n \leq h + 2^c$.
4. $W \rightarrow B$: $\{m, s\}$.

1.4 Получение секретного сообщения

1. B: Боб также проверяет подпись и извлекает секретное сообщение $m^* = s\ mod\ r$.

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 m^* и параметр безопасности k.

2.2 Тестирование программы

```
Введите длину p, q и r: 50
Введите безобидное сообщение m: 12345
Введите параметр безопасности k: 10
Введите секретное сообщение m*: 777
Генерация подписи:
Открытый ключ:
n = 868111887053388648697420317409940652029209229275896907243073
k = 10

Закрытый ключ:
p = 921008072297917
q = 1076188900015217
r = 950955777709321

h = 3832355829417688120
x = 539881343061035059949652467671131314023094437
a = -621105236920349
b = 758789312877463
s = 715209825071897819960022328772351965354869684276538215904584
A -> W(B): { 12345 715209825071897819960022328772351965354869684276538215904584 }


Проверка подписи:
h = 3832355829417688120
c = 150
Результат проверки: True
3832355829417688120 193380086565840251771535398805180971579750431 142724769270595988105828597328185096580043474 4

Получение секретного сообщения:
m* = 777
```

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)
```