БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

ОТЧЁТ  
по лабораторной работе №4  
по дисциплине

«СРЕДСТВА И МЕТОДЫ ЗАЩИТЫ ИНФОРАМЦИИ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ»

Выполнил студент гр. 221703

Вечорко Д. Н.

Проверил

Крищенович В. А.

Минск 2024

Тема: открытое распространение ключей

Цель: приобрести навыки понимания основных принципов шифрования ключей. Изучение и реализация протокола Диффи-Хеллмана.

Задача: Вариант 4: 3917

Для заданного простого P (в соответствии с вариантом) найти g – примитивный элемент конечного поля GF(P) и выполнить генерацию общего секрета. Для нахождения g воспользуйтесь методом перебора по возрастанию, возведения в степень по модулю P и проверки того факта, что все степени принимают значения от 0 до P - 1.

Функция get\_private\_key() будет генерировать случайное число, которое станет частью ключа:

import random

def get\_private\_key(x):

return random.randint(0, x-1)

Функция gcd() высчитывает наибольший общий делитель, необходимый для проверки взаимной простоты чисел:

def gcd(a, b):

while a != b:

if a > b:

a = a - b

else:

b = b - a

return a

Функция get\_primitive\_root() находит первообразный корень по модулю P. Первообразным корнем по модулю P называют такое число g, что все его степени по модулю P принимают значения всех чисел, взаимно простых с P.

def get\_primitive\_root(mod):

required = set(num for num in range(1, mod) if gcd(num, mod) == 1)

for g in range(1, mod):

actual = set(pow(g, power) % mod for power in range(1, mod))

if required == actual:

return g

Функция get\_public\_key() вычисляет открытый ключ для стороны, используя ее закрытый ключ a, общий параметр g (первообразный корень) и модуль p. Открытый ключ вычисляется как g^a mod p.

def get\_public\_key(a, g, p):

public\_key = g

while a > 0:

public\_key = public\_key \* g % p

a = a - 1

return public\_key

Приведем пример работы:

p = 3917

primitive\_root = get\_primitive\_root(p)

alice\_private = get\_private\_key(p)

bob\_private = get\_private\_key(p)

alice\_public = get\_public\_key(alice\_private, primitive\_root, p)

bob\_public = get\_public\_key(bob\_private, primitive\_root, p)

alice\_key = get\_public\_key(alice\_private, bob\_public, p)

bob\_key = get\_public\_key(bob\_private, alice\_public, p)

from tabulate import tabulate

print(

tabulate([

["Алиса", alice\_private, p, primitive\_root, alice\_public, alice\_key],

["Боб", bob\_private, p, primitive\_root, bob\_public, bob\_key],

], headers=["Абонент", "Закрытый ключ", "Простое число", "Первообразный корень", "Открытый ключ", "Ключ"])

)

Вывод: мы приобрели навыки понимания основных принципов шифрования ключей, а также изучили и реализовали протокол Диффи-Хеллмана.