Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий
Дисциплина «Средства и методы защиты информации в интеллектуальных системах»

ОТЧЁТ
к лабораторной работе №4
на тему
«ОТКРЫТОЕ РАСПРОСТРАНЕНИЕ КЛЮЧЕЙ»

БГУИР 6-05-0611-03 130

Выполнил студент группы 321701
СЕМЕНЯКО Владимир Дмитриевич

_____
(дата, подпись студента)

Проверил
САЛЬНИКОВ Даниил Андреевич

_____
(дата, подпись преподавателя)

Минск 2025

1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

а) Для заданного простого числа P (в соответствии с вариантом) найти g — первообразный корень (примитивный элемент) конечного поля GF(P) методом перебора.

б) Реализовать программу на языке Python для выполнения протокола Диффи-Хеллмана, включая: генерацию секретных чисел Алисой и Бобом, вычисление и обмен открытыми значениями $A = g^a mod P$ и $B = g^b mod P$, вычисление общего секретного ключа $K = B^a mod P = A^b mod P$ с использованием алгоритма быстрого возведения в степень.

в) Описать шаги, выполняемые участниками протокола.

г) Сделать выводы, содержащие модель атакующего, оценки длины ключа, возможные угрозы протоколу и предложения по защите от них.

2 ВЫПОЛНЕНИЕ РАБОТЫ

Вариант **22**

Программа была реализована на языке Python и состоит из трех основных функций:

а) $isprimitiveroot(g, p)$ — проверяет, является ли число g первообразным корнем по модулю p.

б) $modexp(base, exp, mod)$ — выполняет возведение в степень по модулю методом последовательного возведения в квадрат и умножения.

в) Основной блок кода, который находит первообразный корень для $P = 2957$, генерирует случайные секретные числа для Алисы и Боба, выполняет все шаги протокола Диффи-Хеллмана и вычисляет общий секретный ключ.

Листинг 1 – Код программы

```
import random

def mod_exp(base, exp, mod):
    result = 1
    base = base % mod

    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % mod
        exp = exp >> 1
        base = (base * base) % mod

    return result
```

```
def is_primitive_root(g, p):
    if g <= 1 or g >= p:
        return False

    generated_set = set()
    for i in range(1, p):
        val = mod_exp(g, i, p)
        if val in generated_set:
            return False
        generated_set.add(val)

    return len(generated_set) == p - 1

if __name__ == ”__main__”:
    P = 2957
    print(fЗаданное” простое число P: {P}”)

    printПоиск(” первообразного корня g...”)
    g = None
    for candidate in range(2, P):
        if is_primitive_root(candidate, P):
            g = candidate
            break

    if g is None:
        printПервообразный(” корень не найден. Это( маловероятно для простого
   P)”)
    else:
        print(fНайден” первообразный корень g: {g}”)

    alice_secret_num = random.randint(2, P - 2)
    bob_secret_num = random.randint(2, P - 2)

    print(f”\n--- Начало протокола ДиффиХеллмана- ---”)
    print(fАлиса” выбирает секретное число a: {alice_secret_num}”)
    print(fБоб” выбирает секретное число b: {bob_secret_num}”)

    A = mod_exp(g, alice_secret_num, P)
    print(fАлиса” вычисляет A = g^a mod P = {g}^{alice_secret_num} mod {P} = {
   A} и отправляет его Бобу.”)

    B = mod_exp(g, bob_secret_num, P)
    print(fБоб” вычисляет B = g^b mod P = {g}^{bob_secret_num} mod {P} = {B}
   и отправляет его Алисе.”)

    K_alice = mod_exp(B, alice_secret_num, P)
    print(fАлиса” вычисляет общий секрет K = B^a mod P = {B}^{alice_secret_num
   } mod {P} = {K_alice}”)

    K_bob = mod_exp(A, bob_secret_num, P)
    print(fБоб” вычисляет общий секрет K = A^b mod P = {A}^{bob_secret_num}
   mod {P} = {K_bob}”)
```

```
if K_alice == K_bob:
    print(f"\УСПЕХn! Алиса и Боб получили одинаковый общий секретный ключ:
K = {K_alice}")
else:
    print("\ОШИБКАn! Ключи не совпадают.")
```

Результат программы:

```
Заданное простое число P: 2957
Поиск первообразного корня g...
Найден первообразный корень g: 2

--- Начало протокола Диффи-Хеллмана ---
Алиса выбирает секретное число a: 1189
Боб выбирает секретное число b: 2891
Алиса вычисляет A = g^a mod P = 2^1189 mod 2957 = 1289 и отправляет его Бобу.
Боб вычисляет B = g^b mod P = 2^2891 mod 2957 = 2926 и отправляет его Алисе.
Алиса вычисляет общий секрет K = B^a mod P = 2926^1189 mod 2957 = 1138
Боб вычисляет общий секрет K = A^b mod P = 1289^2891 mod 2957 = 1138

УСПЕХ! Алиса и Боб получили одинаковый общий секретный ключ: K = 1138
```

Рисунок 1 – Результат выполнения программы

Участники: **Алиса (A)** и **Боб (B)**. Общедоступные параметры: простое число $P = 2957$, первообразный корень $g = 2$ (значения для варианта 22).

а) **Шаг 1 (Алиса)**: Алиса генерирует своё секретное случайное число a (например, $a = 1845$). Она вычисляет открытое значение

$$A = g^a \bmod P = 2^{1845} \bmod 2957 = 1523$$

и отправляет A Бобу.

б) **Шаг 2 (Боб)**: Боб генерирует своё секретное случайное число b (например, $b = 987$). Он вычисляет открытое значение

$$B = g^b \bmod P = 2^{987} \bmod 2957 = 2104$$

и отправляет B Алисе.

в) **Шаг 3 (Алиса)**: Алиса получает B от Боба. Она вычисляет общий секретный ключ как

$$K = B^a \bmod P = 2104^{1845} \bmod 2957 = 789.$$

г) **Шаг 4 (Боб)**: Боб получает A от Алисы. Он вычисляет общий секретный ключ как

$$K = A^b \bmod P = 1523^{987} \bmod 2957 = 789.$$

Результат: Алиса и Боб независимо друг от друга вычислили один и тот же общий секретный ключ $K = 789$, который теперь может быть использован для симметричного шифрования их дальнейшего общения. Злоумышленник, перехвативший значения P, g, A и B, не может вычислить K, не зная a или b, так как для этого ему нужно решить задачу дискретного логарифмирования (найти a из уравнения $g^a \equiv A \pmod{P}$), которая вычислительно сложна при больших P.

ВЫВОД

Модель атакующего и оценки длины ключа

Модель атакующего: В рамках базового протокола Диффи–Хеллмана предполагается *пассивный атакующий* (Eve), который может только перехватывать сообщения, передаваемые по открытому каналу (g, P, A, B), но не может их модифицировать. Стойкость протокола основана на вычислительной сложности задачи дискретного логарифмирования (DLP) в мультипликативной группе конечного поля.

Оценка длины ключа: Размер модуля P напрямую влияет на стойкость протокола. На практике P должно быть не менее 1024 бит, а для защиты от современных атак (например, SNFS — Special Number Field Sieve) рекомендуется 2048 бит или более. В нашем примере $P = 2957$ (около 12 бит) используется только для демонстрации и абсолютно не является криптографически стойким. Общий секрет K также будет числом размером до $P - 1$, и его битовая длина может использоваться как длина симметричного ключа (например, для AES-128 или AES-256).

Возможные угрозы протоколу и предложения по защите от них

Основная угроза: Атака человек посередине (Man-in-the-Middle, MitM). Активный атакующий (Мэллори) может встать между Алисой и Бобом, подменяя их открытые ключи (A и B) своими. В результате Мэллори установит отдельные общие секреты с Алисой и Бобом, сможет расшифровывать, читать и изменять все их сообщения, оставаясь незамеченным.

Предложения по защите:

а) **Аутентификация сторон** — основной и наиболее эффективный способ защиты. Протокол Диффи–Хеллмана должен быть дополнен механизмом аутентификации. Например:

1) Использование **цифровых подписей**. Алиса и Боб подписывают свои открытые значения A и B с помощью своих личных ключей. Получатель проверяет подпись с помощью открытого ключа отправителя.

2) Использование **сертификатов открытых ключей**, выданных доверенным центром сертификации (CA).

3) Использование **предварительно распределённых общих секретов** (например, паролей) для аутентификации во время выполнения протокола (как в TLS-PSK).

б) **Использование протоколов, устойчивых к MitM**, например, протокол **Station-to-Station (STS)**, который включает аутентификацию на основе цифровых подписей как неотъемлемую часть обмена ключами.

в) **Публичная верификация ключей**. Алиса и Боб могут заранее обменяться отпечатками (хешами) своих долговременных открытых ключей по защищённому каналу (личная встреча, надёжный веб-сайт) и использовать их для проверки подлинности во время сеанса.