МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра «Информационных технологий и компьютерные системы»

ОТЧЕТ

о выполнении лабораторной работы №7

по дисциплине: «Методы и алгоритмы защиты информации»

Выполнил:

ст.гр. ПИН/б-21-1-о

Зражевский А.С.

Проверил:

Лебедева М.А.

Севастополь, 2025г.

ЦЕЛЬ РАБОТЫ

Изучение особенностей реализации криптографических протоколов распределения ключей, асимметричной криптографии на эллиптических кривых, разработка системы распределения криптографических ключей.

ПОСТАНОВКА ЗАДАЧИ

1. Выбрать коэффициенты a,b и модуль p эллиптической кривой, координаты x, y точки G, а также секретные значения k1, k2 абонентов из таблицы 2.1 в соответствии с вариантом.

2. Разработать программную реализацию метода Диффи-Хеллмана. Предусмотреть проверку эллиптической кривой по формуле (2.2). Исходными данными являются параметры кривой, координаты точки и секретные значения каждого участника обмена. Результат работы программы – координаты произведения точки G на число, которые должны совпасть у каждого из участников.

3. Оформить отчет.

Таблица 1 – Исходные данные протокола Диффи-Хеллмана

№ вар	a	b	p	$G(x,y)$	k_1	k_2
10	-1	3	37	(2, 3)	13	5

ХОД РАБОТЫ

Протокол Диффи-Хеллмана на эллиптических кривых (ECDH) — это метод асимметричной криптографии, использующий свойства эллиптических кривых для обмена ключами. В отличие от классического Диффи-Хеллмана, основанного на модульной арифметике, ECDH использует операции над точками эллиптической кривой. Безопасность метода основана на сложности задачи дискретного логарифмирования в группе точек кривой (ECDLP).

Основные шаги метода:

1. Инициализация параметров кривой:
   - Задается уравнение эллиптической кривой: $y^2 = x^3 + a \cdot x + b \bmod p$.
   - Выбирается базовая точка G на кривой.
   - Проверяется условие кривой: $4a^3 + 27b^2 \neq 0 \bmod p$.
2. Генерация ключей:
   - Первый участник выбирает секретный ключ k_1 и вычисляет открытую точку $A = k_1 \cdot G$.
   - Второй участник выбирает секретный ключ k_2 и вычисляет открытую точку $B = k_2 \cdot G$.
3. Обмен и вычисление общего секрета:
   - Участники обмениваются открытыми точками A и B.
   - Первый участник вычисляет общий секрет $S_1 = k_1 \cdot B$.
   - Второй участник вычисляет общий секрет $S_2 = k_2 \cdot A$.
   - Если протокол выполнен верно, $S_1 = S_2$.

Особенности:

- Безопасность обеспечивается сложностью задачи ECDLP.
- Умножение точки на скаляр реализуется методом удвоения-сложения.
- Все операции выполняются по модулю p.

Описание исходных данных

1. Параметры эллиптической кривой:
   - Уравнение: $y^2 = x^3 – x + 3 \bmod 37$.
   - a = -1 — коэффициент при x.
   - b = 3 — свободный член.
   - p = 37 — простое число, определяющее конечное поле.
   - Проверка корректности: $4a^3 + 27b^2 \neq 0 \pmod p$

Вычисляем: $(4 \cdot (-1)^3 + 27 \cdot 3^2 = 4 \cdot (-1) + 27 \cdot 9 = -4 + 243 = 239)$.

$(239 \bmod 37 = 239 - 6 \cdot 37 = 239 - 222 = 17 \neq 0)$.

Условие выполнено, кривая корректна.

2. Базовая точка G:
   - $G = (2, 3)$.
   - Проверка принадлежности кривой:

Левая часть: $(y^2 = 3^2 = 9)$.

Правая часть: $(x^3 + a \cdot x + b = 2^3 + (-1) \cdot 2 + 3 = 8 - 2 + 3 = 9)$.

$(9 \bmod 37 = 9)$, $(9 = 9)$, точка (G) принадлежит кривой.

3. Секретные ключи:
   - $k_1=13$ — секретный ключ первого участника.
   - $k_2=5$ — секретный ключ второго участника.

Алгоритм работы программы

1. Инициализация параметров:
   Задаются $a = -1$, $b = 3$, $p=37$, $G=(2, 3)$, $k_1 = 13$, $k_2 = 5$.
2. Проверка корректности:
   - Проверка условия $4a^3 + 27b^2 \neq 0 \bmod p$.
   - Проверка принадлежности точки G кривой.
3. Генерация открытых ключей:
   - $A = k_1 \cdot G = 13 \cdot (2, 3)$.
   - $B = k_2 \cdot G = 5 \cdot (2, 3)$.
4. Вычисление общего секрета:
   - $S_1 = k_1 \cdot B$
   - $S_2 = k_2 \cdot A$
5. Проверка результата:
   - Сравнение координат S_1 и S_2

Операции на кривой:

- Сложение точек: используются формулы для λ, x_3, y_3.
- Умножение на скаляр: метод удвоения-сложения.
- Обратный элемент: расширенный алгоритм Евклида.

Текст программы

```
# Параметры эллиптической кривой: y^2 = x^3 + a*x + b mod p
a = -1
b = 3
p = 37
G = (2, 3)  # Базовая точка G(x, y)
k1 = 13     # Секретный ключ первого участника
k2 = 5      # Секретный ключ второго участника

def mod_inverse(a, m):
    """Вычисление обратного элемента по модулю с помощью расширенного алгоритма Евклида"""
    def extended_gcd(a, b):
        if a == 0:
            return b, 0, 1
        gcd, x1, y1 = extended_gcd(b % a, a)
        x = y1 - (b // a) * x1
        y = x1
        return gcd, x, y
    gcd, x, _ = extended_gcd(a, m)
    if gcd != 1:
        raise ValueError("Обратное число не существует")
    return (x % m + m) % m

def add_points(p1, p2):
    """Сложение двух точек на эллиптической кривой"""
    if p1 is None:
        return p2
    if p2 is None:
```

```
        return p1
    x1, y1 = p1
    x2, y2 = p2
    if x1 == x2 and (y1 != y2 or y1 == 0):
        return None  # Точки противоположны или удвоение точки с y=0
    if x1 == x2 and y1 == y2:  # Удвоение точки
        numerator = (3 * x1 * x1 + a) % p
        denominator = (2 * y1) % p
    else:  # Сложение разных точек
        numerator = (y2 - y1) % p
        denominator = (x2 - x1) % p
    lambda_val = (numerator * mod_inverse(denominator, p)) % p
    x3 = (lambda_val * lambda_val - x1 - x2) % p
    y3 = (lambda_val * (x1 - x3) - y1) % p
    return (x3, y3)

def multiply_point(point, k):
    """Умножение точки на скаляр методом удвоения-сложения"""
    if k == 0:
        return None
    result = None
    temp = point
    while k > 0:
        if k & 1:
            result = add_points(result, temp)
        temp = add_points(temp, temp)
        k >>= 1
    return result

def check_curve():
    """Проверка корректности параметров кривой"""
    term1 = (4 * pow(a, 3, p)) % p
    term2 = (27 * pow(b, 2, p)) % p
    result = (term1 + term2) % p
    return result != 0
```

```
def check_point(point):
    """Проверка принадлежности точки кривой"""
    x, y = point
    left = pow(y, 2, p)
    right = (pow(x, 3, p) + a * x + b) % p
    return left == right

# Основная логика программы
print("Параметры эллиптической кривой:")
print(f"a = {a}, b = {b}, p = {p}")
print(f"G = {G}")
print(f"k1 = {k1}, k2 = {k2}")
print()

if not check_curve():
    print("Ошибка: параметры кривой не удовлетворяют условию 4a³ + 27b² ≠ 0 mod p")
    exit()

if not check_point(G):
    print("Ошибка: точка G не принадлежит кривой")
    exit()

print(f"Уравнение кривой: y² = x³ + {a}x + {b} mod {p}")
print()

# Вычисление открытых точек
A = multiply_point(G, k1)
print(f"Первый участник: A = k1 * G = {A}")
B = multiply_point(G, k2)
print(f"Второй участник: B = k2 * G = {B}")
print()

# Вычисление общего секрета
S1 = multiply_point(B, k1)
print(f"Общий секрет S1 (первый участник): {S1}")
```

```
S2 = multiply_point(A, k2)
print(f"Общий секрет S2 (второй участник): {S2}")
print()

# Проверка совпадения секретов
if S1 == S2:
    print("Протокол выполнен успешно! Общие секреты совпадают.")
else:
    print("Ошибка: общие секреты не совпадают!")
```

Результат работы программы представлен на рисунке 1.

```
============== RESTART: C:/Users/kotte/OneDrive/Рабочий стол/7.py =====
Параметры эллиптической кривой:
a = -1, b = 3, p = 37
G = (2, 3)
k1 = 13, k2 = 5

Уравнение кривой: y² = x³ + -1x + 3 mod 37

Первый участник: A = k1 * G = (2, 34)
Второй участник: B = k2 * G = (23, 23)

Общий секрет S1 (первый участник): (23, 14)
Общий секрет S2 (второй участник): (23, 14)

Протокол выполнен успешно! Общие секреты совпадают.
```

Рисунок 1 – Результат работы программы

ВЫВОД

В ходе работы были изучены особенности протокола Диффи-Хеллмана на эллиптических кривых, реализована система распределения ключей на Python. Метод обеспечивает безопасность благодаря сложности ECDLP и эффективности за счет меньших размеров ключей по сравнению с классическим DH.