Міністерство освіти і науки України

Національний університет «Запорізька політехніка»

кафедра програмних засобів

ЗВІТ

з лабораторної роботи № 5

з дисципліни «Безпека та захист програм і даних» на тему:

«КРИПТОГРАФІЯ ПЕРЕТВОРЕННЯ НА ЕЛІПТИЧНИХ КРИВИХ»

Виконав:

ст. гр. КНТ-113сп Іван ЩЕДРОВСЬКИЙ

Прийняв:

доцент Тетяна ЗАЙКО

2025

1 Мета роботи

Ознайомитись з алгоритмами перетворення на еліптичних кривих.

2 Завдання до лабораторної роботи

2.1 Дано еліптичну криву $y^2 = x^3 + ax + b \, mod \, p$. Параметри a та b відповідають 30 варіанту з таблиці з методичних вказівок, а саме $a = 3, b = 9$

Необхідно:

- знайти всі точки, що належать заданій кривій та порядок кривої
- знайти базову точку
- представити кожну точку як кратну базовій (знайти показник кратності)

2.2 Реалізувати алгоритм цифрового підпису ГОСТ Р 34.10. Для тестування використати наступні параметри (у hex-формі):

Еліптична крива: $y^2 = x^3 + ax + b \, mod \, m$

Модуль перетворень:

m =8000000000000000000000000000000000000000000000000000000000000431

a=7

b =5FBFF498AA938CE739B8E022FBAFEF40563F6E6A3472FC2A514C0CE9DAE23B7E

Базова точка $P(x, y)$:

P_x =2

P_y =8E2A8A0E65147D4BD6316030E16D19C85C97F0A9CA267122B96ABBCEA7E8FC8

Порядок базової точки:

n=800000000000000000000000000000150FE8A1892976154C59CFC193ACCF5B3

Секретний ключ:

d=7A929ADE789BB9BE10ED359DD39A72C11B60961F49397EEE1D19CE9891EC3B28

Результат множення (відкритий ключ) – точка $Q(Q_x, Q_y)$:

Q_x =7F2B49E270DB6D90D8595BEC458B50C58585BA1D4E9B788F6689DBD8E56FD80B

Q_y =26F1B489D6701DD185C8413A977B3CBBAF64D1C593D26627DFFB101A87FF77DA

3 Відповіді на контрольні питання

Надати визначення еліптичної кривої

Еліптична крива – це математичний об'єкт, що може бути визначений над будь-яким полем, зокрема над полем дійсних чисел або над скінченим полем Галуа

Як визначити точку, обернену даній?

Щоб визначити точку, обернену даній потрібно помножити значення у на -1

Це працює тому, що функція є симетричною відносно осі x

Які параметри еліптичної кривої необхідно знати для її застосування?

Для застосування еліптичної кривої $E: y^2 = x^3 + ax + b \; mod \; p$ потрібно знати a, b та p, такі, щоб

Потрібно знати базову точку P, також потрібно знайти загальну кількість елементів

Також потрібно обрати приватний ключ d, та знайти публічний ключ T

З таким набором параметрів ми можемо використовувати еліптичні криві

Надати визначення порядку групи точок еліптичної кривої

Порядком групи точок, або ж group cardinality, еліптичної кривої E називається число елементів цієї групи та позначається як #E

Надати визначення порядку точки еліптичної кривої

Порядок точки еліптичної кривої це найменше додатне ціле число x, для якого вірно порівняння $x \cdot P = \Theta$, де P – базова точка

Також можна сказати, що порядок точки це кількість елементів n, які будуть знаходитись в групі якщо ця точка буде генератором. Для базової точки це будуть всі можливі елементи

Як визначити базову точку?

Точка P називається базовою точкою підгрупи точок еліптичної кривої E, або ж ще називається генератором, якщо будь-яка точка Q цієї підгрупи може бути подана у вигляді $Q = k \cdot P$, де k = 1,2,…,n, та n – порядок підгрупи #E, або ж простіше число елементів цієї групи

Для базової точки має місце рівняння $P = n \cdot P = \Theta$

Для визначення базової точки для кривої є складні методи, які не розглядаються в рамках цієї лабораторної роботи, а також в більшості випадків використовують стандартні криві, з визначеною базовою точкою та визначеною кількістю елементів

У яких криптографічних алгоритмах застосовуються еліптичні криві?

Еліптичні криві можна застосовувати в Diffie-Hellman, ECDH для створення секретного ключа через відкритий канал

Також ці криві застосовують для цифрового підпису, ECDSA для підтвердження, що дані не були змінені під час передачі та підписані заявленим відправником

І також еліптичні криві застосовують для шифрування та дешифрування інформації, ECIES

Наприклад, після того, як ми створити сесійний ключ через DH ми можемо взяти x координату та використати її як ключ для AES, зашифрувати інформації

4 Текст програми

```
//// First task

a = 3
b = 9
p = 7

E = EllipticCurve(GF(p), [a, b])
N = E.order()
ZERO = E((0, 1, 0))

print(E)
print("E:", E.defining_polynomial())
print("ZERO: ", ZERO)

print("\nPoints of EC:")

for point in E.points():
   if point.is_zero():
      print("Identity", end='\t')
   else:
      print(point.xy(), end='\t')

print("\nPoints count:", len(E.points()))

print("\n#E: ", N)

P = E((0, 3, 1))
print("Base point:", P.xy())
print("Base check 3*P should not be ZERO.", not (3*P).is_zero())
print("Base check N*P should be ZERO. ", (N*P).is_zero())

print("\n| k | k*P | Q(x, y)|")
print("|---|-----|--------|")


for k in range(1, N):
   Q = k * P
   print(f"| {k} | {k}*P | {Q.xy()} |")
```

```
//// Second task

import random

# From LB 4
def hash_message_string(message_str: str) -> int:
    hash_value = 9 # H0
    MOD = 33

    for char in message_str:
        value = ord(char)

        current_sum = hash_value + value
        power = current_sum * current_sum

        hash_value = power % MOD
        hash_value = (hash_value + value) * power % MOD

    return hash_value

p = int("8000000000000000000000000000000000000000000000000000000000000431", 16)
a = 7
b = int("5FBFF498AA938CE739B8E022FBAFEF40563F6E6A3472FC2A514C0CE9DAE23B7E", 16)

Px = 2
Py = int("8E2A8A0E65147D4BD6316030E16D19C85C97F0A9CA267122B96ABBCEA7E8FC8", 16)

N = int("800000000000000000000000000000150FE8A1892976154C59CFC193ACCF5B3", 16)

d = int("7A929ADE789BB9BE10ED359DD39A72C11B60961F49397EEE1D19CE9891EC3B28", 16)

Qx = int("7F2B49E270DB6D90D8595BEC458B50C58585BA1D4E9B788F6689DBD8E56FD80B", 16)
Qy = int("26F1B489D6701DD185C8413A977B3CBBAF64D1C593D26627DFFB101A87FF77DA", 16)

MESSAGE = "HELLO, EC!"
H = hash_message_string(MESSAGE)

E = EllipticCurve(GF(p), [a, b])
ZERO = E((0, 1, 0))

P = E((Px, Py, 1))


print("E:", E)
print("#E or N:", N)
print("P:", P.xy())
print("N*P is zero?", (N*P).is_zero())

print("\nMessage:", MESSAGE)
print("Message hash:", H)

print("\nSignature generation")

k = random.randint(1, N - 1)
Pk = k * P
r = Integer(Pk.x()) % N
s = pow(k, -1, N) * (H + r * d) % N
```

```
print("K(session key):", k)
print("Pk:", Pk)
print("r:", r)
print("s:", s)

print("\nSignature verification")

u1 = H * pow(s, -Integer(1), N) % N
u2 = r * pow(s, -Integer(1), N) % N

U = u1 * P + u2 * E((Qx, Qy, 1))
print("u1:", u1)
print("u2:", u2)
print("U:", U)
print("Is r == Ux?", r == Integer(U.x()))
```

5 Результати роботи програми

Для виконання обох завдань використовувався сервіс Sage Cell від SageMath

Виконання першого завдання. В якості p було обрано число 7, оскільки число 5 не відповідає умові гладкості для моїх a та b

При виконанні завдання було знайдено всі точки, які належать заданій кривій, її порядок, базову точку та показник кратності для кожної точки

Результат виконання показаний на рисунку 1

```
Elliptic Curve defined by y^2 = x^3 + 3*x + 2 over Finite Field of size 7
E: -x^3 + y^2*z - 3*x*z^2 - 2*z^3
ZERO:  (0 : 1 : 0)

Points of EC:
Identity        (0, 3)  (0, 4)  (2, 3)  (2, 4)  (4, 1)  (4, 6)  (5, 3)  (5, 4)
Points count: 9

#E:  9
Base point: (0, 3)
Base check 3*P should not be ZERO, 3*P =  (5 : 4 : 1)
Base check N*P should be ZERO (0 : 1 : 0)

| k | k*P | Q(x, y)|
|---|-----|--------|
| 1 | 1*P | (0, 3) |
| 2 | 2*P | (2, 3) |
| 3 | 3*P | (5, 4) |
| 4 | 4*P | (4, 6) |
| 5 | 5*P | (4, 1) |
| 6 | 6*P | (5, 3) |
| 7 | 7*P | (2, 4) |
| 8 | 8*P | (0, 4) |
```

Рисунок 1 – Виконання першого завдання

В другому завданні в b було змінено літеру O на нуль, оскільки літера O не є валідним символом в hex-формі

Було написано програму, яка розраховує цифровий підпис (r, s) на основі хешу повідомлення. Для створення хешу було використано функцію з попередньої лабораторної роботи, але без таблиці

Також було виконано перевірку цифрового підпису

Результат другого завдання показаний на рисунку 2.

```
E: Elliptic Curve defined by y^2 = x^3 + 7*x + 433088765467672769057659045956509319959421117944510395832529688420338495804l4 over Finite Field of size 57896044618658097711785492504343953927082934583725450622380973592137631069619
#E or N: 57896044618658097711785492504343953927082934583725450622380973592137631069619
P: (2, 401897405653903750333544942293705977563573938990554508069097936521343156628O)
N*P is zero? True

Message: HELLO, EC!
Message hash: 16

Signature generation
K(session key): 30548690262232908052819476989623757788066709931152104193658743142007116050613
Pk: (55441998614036947052393561141644770115609239129342920137707266281448811235380 : 13794379148559630021638407541084491437713461639827582590646649260485426508831 : 1)
r: 55441998614036947052393561141644770115609239129342920137707266281448811235380
s: 92675489649606639443490592608141464986844624225901652906946498639850932733

Signature verification
u1: 42611707556719240095687239607911178556745224166214190110779786803604182549127
u2: 53767266522390771853803018357086933454481654847194701354943086024986050620636
U: (55441998614036947052393561141644770115609239129342920137707266281448811235380 : 13794379148559630021638407541084491437713461639827582590646649260485426508831 : 1)
Is r == Ux? True
```

Рисунок 2 – Виконання другого завдання

6 Висновки

Я ознайомився з алгоритмами перетворення на еліптичних кривих