МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ЗВІТ

ЗВІТ З ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ»

Виконав студент групи КН-23-1
Полинько Ігор Миколайович
Перевірив: ассистент кафедри АІС Андреєв П. І.

Кременчук 2025

ЛАБОРАТОРНА РОБОТА № 7

Тема: Системи з відкритим ключем. Алгоритм RSA

Мета: вивчити алгоритм шифрування RSA.

Порядок виконання роботи:

Скласти програму шифрування повідомлення за допомогою алгоритму RSA і його розшифрування. Для подання даних результату піднесення до степені використовувати тип LongInt. Щоб не виникало помилки переповнення, вихідне повідомлення розглядати як послідовність символів з кодами 0, 1, ..., 9. Вихідне повідомлення, криптограму та відновлене повідомлення зберігати у файлах.

Таблиця 7.3

N варіанта	P	Q	e	d	Вихідний алфавіт
15-16	5	7	5	5	1234567890

Варіант: 15

Скрипт програми:

```
P = 5
Q = 7
e = 5
d = 5

n = P * Q # модуль
phi = (P - 1) * (Q - 1)

# Функція шифрування одного символу (цифри)
def encrypt_digit(m):
    return pow(m, e, n)

# Функція дешифрування одного числа
def decrypt_digit(c):
    return pow(c, d, n)

# Зчитуємо повідомлення з файлу
with open("input.txt", "r") as f:
    message = f.read().strip()

# Перевірка цифр
if not message.isdigit():
    raise ValueError("Повідомлення повинно містити тільки цифри 0-9.")

# Шифрування
cipher = [encrypt_digit(int(ch)) for ch in message]

# Запис криптограми у файл
with open("cipher.txt", "w") as f:
```

```
    f.write(" ".join(map(str, cipher)))

# Дешифрування
restored_digits = [decrypt_digit(int(x)) for x in cipher]
restored_message = "".join(str(x) for x in restored_digits)

# Запис відновленого повідомлення
with open("output.txt", "w") as f:
    f.write(restored_message)

print("Вихідне:", message)
print("Криптограма:", cipher)
print("Розшифровано:", restored_message)
```

Результат:

```
C:\Python313\python.exe C:\Users\RoRHaT\Documents\GitHub\tzi-polynko-kn-23-1\7\python\script.py
Вихідне: 1234567890
Криптограма: [1, 32, 33, 9, 10, 6, 7, 8, 4, 0]
Розшифровано: 1234567890

Process finished with exit code 0
```

Рисунок 7.1 – Результат роботи програми

Висновок: на цій лабораторній роботі ми вивчили алгоритм шифрування RSA. Зашифрували та розшифрували числове повідомлення і записали результати у текстові файли.

Контрольні питання:

1. Який сенс систем з відкритим ключем?

Вони розв’язують проблему безпечного обміну ключами. Дозволяють шифрувати дані без попередньої передачі секретного ключа, створювати електронні підписи та забезпечувати автентичність і цілісність повідомлень.

2. За допомогою яких ключів шифрується та розшифровується повідомлення в СВК?

– Шифрування виконується відкритим (публічним) ключем одержувача.

– Розшифрування виконується закритим (приватним) ключем одержувача.

3. Що таке “необоротні функції”? Які типи необоротних перетворень використовуються у СВК?

Необоротна функція – це така функція, яку легко обчислити, але практично неможливо обернути без додаткової інформації.

У СВК застосовують:

– факторизацію великих чисел (RSA),

– дискретний логарифм у скінченних полях (Diffie–Hellman, DSA),

– дискретний логарифм на еліптичних кривих (ECC).

4. Які головні вимоги висувають до СВК?

– Коректність (дешифрування дає початкове повідомлення).

– Криптостійкість.

– Неможливість знайти приватний ключ за відкритим.

– Ефективність та можливість безпечного управління ключами.

5. Для чого можна використовувати алгоритми криптосистем з відкритим ключем?

Для шифрування повідомлень, обміну ключами, створення та перевірки цифрових підписів, автентифікації, захисту мережевих протоколів (TLS, SSL), електронних документів тощо.

6. На яких математичних фактах ґрунтується алгоритм RSA?

– На складності факторизації числа $n = pq$.

– На теоремі Ейлера: $m^{\varphi(n)} \equiv 1 \ mod\ n$.

– На властивостях модульної арифметики зі степенями.

7. Як вибираються числа P і Q алгоритму RSA?

P і Q вибирають як два великі, різні, випадкові прості числа приблизно однакової довжини. Після вибору обов'язково перевіряють їх на простоту (тест Міллера–Рабіна).

8. Які значення засновник RSA повідомляє користувачам, а які зберігає в таємниці?

Публікує:

– $n = pq$,

– e – відкритий показник.

Зберігає в таємниці:

– $p, q,$

– d – приватний показник.

9. Чи можна розшифрувати повідомлення за допомогою відкритого ключа?

Ні. Відкритий ключ використовується лише для шифрування та перевірки підписів. Розшифрувати криптограму можна тільки приватним ключем.

10. Як обчислюють значення функції Ейлера? Для чого його використовують в RSA?

Якщо $n = pq$, де p і q – прості, то:

$$\varphi(n) = (p-1)(q-1)$$

У RSA $\varphi(n)$ використовують для знаходження приватного ключа d, який є мультиплікативним оберненим до eза модулем $\varphi(n)$.

11. За допомогою яких формул здійснюють шифрування та дешифрування?

Шифрування:

$$c \equiv m^e \mod n$$

Дешифрування:

$$m \equiv c^d \mod n$$

12. Чи зміниться криптограма, якщо числа P і Q поміняти місцями?

Ні. Добуток $n = pq$ не зміниться, тому параметри ключів також не змінюються.

13. $P = 3$, $Q = 13$. Які числа з $2, 3, 5, 9, 29$ можна використовувати як e?

Обчислюємо:

$$\varphi(n) = (3-1)(13-1) = 24$$

Потрібно: $\gcd(e, 24) = 1$ і $1 < e < 24$

Перевірка:

– 2 → Ні ($\gcd = 2$)

– 3 → Ні ($\gcd = 3$)

– 5 → Так

– 9 → Ні ($\gcd = 3$)

– 29 → Ні (перевищує $\varphi(n)$)

Відповідь: можна використовувати лише 5.

14. Приклад RSA: $P = 3$, $Q = 5$, $e = 3$, $M = \{3, 2\}$.

1) Обчислюємо параметри

$$n = 3 \cdot 5 = 15$$

$$\varphi(n) = (3-1)(5-1) = 8$$

Знайдемо d:

Потрібно $3d \equiv 1 \ mod\ 8$.

$$d = 3$$

бо $3 \cdot 3 = 9 \equiv 1 \ mod\ 8$.

2) Шифрування

Формула: $c = m^e \ mod\ n$.

– Для $m = 3$:

$$3^3 = 27, \qquad 27 \ mod\ 15 = 12$$

– Для $m = 2$:

$$2^3 = 8, \qquad 8 \ mod\ 15 = 8$$

Криптограма: {12,8}

3) Дешифрування

Формула: $m = c^d \ mod\ n$.

– Для $c = 12$:

$$12^3 = 1728, \qquad 1728 \ mod\ 15 = 3$$

– Для $c = 8$:

$$8^3 = 512, \qquad 512 \ mod\ 15 = 2$$

Отримане повідомлення: {3,2}.