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 № 5

Тема: Шифрування алгоритмом DES

Мета: навчитися створювати програми шифрування (дешифрування) алгоритмом DES.

Порядок виконання роботи:

Реалізувати алгоритм DES (шифрування та дешифрування):

7−8 Робочий режим зворотний зв'язок за виходом OFB (Output Feed Back).

Варіант: 15

Скрипт програми:

```
from Crypto.Cipher import DES
from Crypto.Util.Padding import pad, unpad
import os

# Функція шифрування у режимі OFB
def des_encrypt_ofb(key, plaintext):
    iv = os.urandom(8)

    cipher = DES.new(key, DES.MODE_OFB, iv)

    padded_text = pad(plaintext.encode('utf-8'), 8)

    ciphertext = cipher.encrypt(padded_text)

    return iv, ciphertext


# Функція дешифрування у режимі OFB
def des_decrypt_ofb(key, iv, ciphertext):
    cipher = DES.new(key, DES.MODE_OFB, iv)

    decrypted_padded = cipher.decrypt(ciphertext)

    decrypted = unpad(decrypted_padded, 8)

    return decrypted.decode('utf-8')


key = b'12345678'

plaintext = "Це тестовий текст для шифрування DES OFB"

print("Вихідний текст:", plaintext)

# Шифрування
iv, ciphertext = des_encrypt_ofb(key, plaintext)
print("IV:", iv.hex())
print("Шифротекст:", ciphertext.hex())

# Дешифрування
decrypted_text = des_decrypt_ofb(key, iv, ciphertext)
print("Розшифрований текст:", decrypted_text)
```

Результат:

Рисунок 5.1 – Результат роботи програми

Висновок: на цій лабораторній роботі ми навчилися створювати програми шифрування (дешифрування) алгоритмом DES.

Контрольні питання:

1. Структура алгоритму DES

Алгоритм DES є блочним симетричним шифром.

– Довжина блока: 64 біти, ключа — 56 біт.

– Процес складається з:

1) Початкової перестановки (IP);

2) 16 раундів за схемою Фейстеля:

– поділ блока на півблоки L і R;

– у кожному раунді:

L_i = R_{i-1},

R_i = L_{i-1} XOR F(R_{i-1}, K_i);

3) Кінцевої перестановки (**IP⁻¹**).

Розшифрування виконується аналогічно, але підключі використовуються у зворотному порядку.

2. Функція шифрування F

Функція F(R, K) виконує:

1) Розширення (E): 32 біти → 48 біт;

2) XOR з підключем K;

3) S-перетворення: 8 S-блоків по 6→4 біти;

4) P-перестановка: перестановка 32 біт за таблицею.

Результат — 32 біти.

3. Алгоритм обчислення підключів

1) Ключ 64 біти → PC-1 → 56 біт (видалення бітів парності).

2) Поділ на C і D (по 28 біт).

3) Для кожного з 16 раундів:

– циклічний зсув вліво на 1 або 2 біти;

– PC-2 – формування підключу K_i (48 біт).

У розшифруванні ключі застосовуються у зворотному порядку.

4. Зв'язок IP і IP^{-1}

Матриця IP^{-1} є оберненою до IP.

Застосування IP, а потім IP^{-1} повертає початковий порядок бітів: $IP^{-1}(IP(дані)) = дані$.

5. Режим ECB (Electronic Code Book)

Кожен блок шифрується незалежно одним ключем:

C_i = E_K(P_i)

Недолік — однакові блоки породжують однакові шифроблоки.

6. Режим CBC (Cipher Block Chaining)

Кожен блок перед шифруванням XOR'иться з попереднім шифроблоком:

C_i = E_K(P_i XOR C_{i-1}), C_0 = IV.

Підвищує стійкість за рахунок залежності від попередніх блоків.

7. Режим CFB (Cipher Feedback)

Шифрується вхідний регістр (IV), результат XOR'иться з відкритим текстом: C_i = P_i XOR E_K(ShiftReg)

Регістр оновлюється частиною шифротексту.

Підходить для потокових даних.

8. Режим OFB (Output Feedback)

Вихід шифру подається назад на вхід: $O_i = E_K(O_{i-1})$, $C_i = P_i \text{ XOR } O_i$.

Помилки передачі не накопичуються, бо зворотного зв’язку по шифротексту немає.