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 № 4

Тема: Симетричні криптосистеми: шифри складної заміни

Мета: навчитися розробляти програми для шифрування (дешифрування) методиками симетричних криптосистем.

Порядок виконання роботи:

Скласти програму шифрування повідомлення одним із шифрів складної заміни. Вихідне повідомлення, кодоване повідомлення, ключ і відновлене повідомлення зберігати у файлах. Алфавіт вихідного повідомлення:

Варіанти 11–20 – Український

Таблиця 4.1

N варіанта	Повідомлення
15-16	ШИФР – ОСНОВА ЗАХИСТУ

Варіант: 15

Скрипт програми:

```
ukr_alphabet = "АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ -"
message = "ШИФР - ОСНОВА ЗАХИСТУ"
key = "КЛЮЧ"

def vigenere_encrypt(text, key, alphabet):
    res = ""
    key = key.upper()
    key_index = 0
    for char in text:
        if char not in alphabet:
            res += char
            continue
        text_idx = alphabet.index(char)
        key_idx = alphabet.index(key[key_index % len(key)])
        res += alphabet[(text_idx + key_idx) % len(alphabet)]
        key_index += 1
    return res

def vigenere_decrypt(cipher, key, alphabet):
    res = ""
    key = key.upper()
    key_index = 0
    for char in cipher:
        if char not in alphabet:
            res += char
            continue
```

```
        cipher_idx = alphabet.index(char)
        key_idx = alphabet.index(key[key_index % len(key)])
        res += alphabet[(cipher_idx - key_idx) % len(alphabet)]
        key_index += 1
    return res

# Шифрування
encrypted = vigenere_encrypt(message, key, ukr_alphabet)
# Розшифрування
decrypted = vigenere_decrypt(encrypted, key, ukr_alphabet)

# Збереження у файли
with open("original.txt", "w", encoding="utf-8") as f:
    f.write(message)

with open("key.txt", "w", encoding="utf-8") as f:
    f.write(key)

with open("encrypted.txt", "w", encoding="utf-8") as f:
    f.write(encrypted)

with open("decrypted.txt", "w", encoding="utf-8") as f:
    f.write(decrypted)

print("Повідомлення:", message)
print("Ключ:", key)
print("Зашифроване:", encrypted)
print("Розшифроване:", decrypted)
```

Результат:

```
Повідомлення: ШИФР - ОСНОВА ЗАХИСТУ
Ключ: КЛЮЧ
Зашифроване: ЄХРЇЇ-ЙКЙЮ  ЧЇФЮНФБОЛ
Розшифроване: ШИФР - ОСНОВА ЗАХИСТУ
```

Рисунок 4.1 – Результат роботи програми

original	06.11.2025 23:04	Текстовый докум...	1 КБ
key	06.11.2025 23:04	Текстовый докум...	1 КБ
encrypted	06.11.2025 23:04	Текстовый докум...	1 КБ
decrypted	06.11.2025 23:04	Текстовый докум...	1 КБ

Рисунок 4.2 – Результат збереження файлів

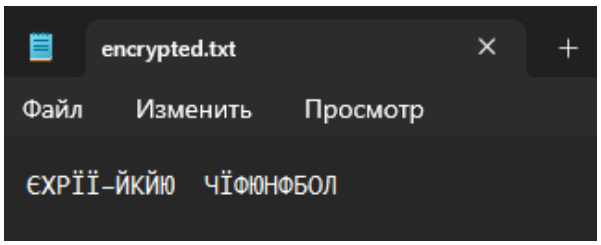

Рисунок 4.3 – Вміст файлу на прикладі encrypted

Висновок: на цій лабораторній роботі ми навчилися розробляти програми для шифрування (дешифрування) методиками симетричних криптосистем.

Контрольні питання:

1. Характерна ознака симетричних систем — використання одного й того ж ключа для шифрування та розшифрування повідомлення.

2. Ключ — це секретний параметр (набір символів або число), який визначає спосіб шифрування й розшифрування даних. Без ключа прочитати повідомлення неможливо.

3. Суть шифрування заміною (підстановкою) полягає в тому, що кожен символ відкритого тексту замінюється іншим символом за певним правилом або таблицею підстановки.

4. Спростити виконання алгебраїчних дій дозволяє перехід до числового представлення символів (наприклад, через ASCII-коди або алфавітні індекси).

5. Шифри складної заміни відрізняються тим, що заміна залежить не лише від самого символу, а й від його позиції чи попередніх символів, тоді як у простих підстановках кожна літера завжди шифрується однаково.

6. Одноразовою систему шифрування називають тому, що ключ використовується лише один раз, після чого його потрібно знищити — це робить злам практично неможливим.

7. Кодування здійснюється на підставі математичних або логічних перетворень, які змінюють вигляд даних за певними правилами (наприклад, арифметичні операції, таблиці заміни, перестановки тощо).

8. Переваги одноразової системи: абсолютна криптостійкість (неможливо зламати без ключа).

Недоліки: складність у зберіганні й передачі великої кількості унікальних ключів і неможливість повторного використання.