УДК 621.391

О. В. Гуменна
І.А. Клєопа

РЕАЛІЗАЦІЯ ШИФРУ ГІЛЛА ЗАСОБАМИ PYTHON ІЗ ВИКОРИСТАННЯМ ОПЕРАЦІЙ НАД МАТРИЦЯМИ

Вінницький національний технічний університет

Аннотація

Метою даної статті є вивчення принципів симетричного шифрування на основі лінійної алгебри, а також практична реалізація класичного криптографічного алгоритму — шифру Гілла — мовою програмування Python. У ході виконання роботи необхідно було реалізувати механізми шифрування та дешифрування тексту, забезпечити коректну роботу з українським алфавітом та перевірку оборотності матриці ключа.

Ключові слова: мова програмування Python, шифр Гілла, матриці.

Abstract

The purpose of this article is to study the principles of symmetric encryption based on linear algebra, as well as the practical implementation of a classical cryptographic algorithm — the Gill cipher — in the Python programming language. In the course of the work, it was necessary to implement mechanisms for encrypting and decrypting text, ensure correct work with the Ukrainian alphabet, and check the reversibility of the key matrix.

Keywords: Python programming language, Gill cipher, matrices.

Вступ

Шифр Гілла є симетричним блоковим шифром, запропонованим Лестером Гіллом у 1929 році. Його основою є математичні операції над матрицями, а саме множення векторів відкритого тексту на квадратну матрицю ключа з подальшим обчисленням результату за певним модулем. Важливою умовою коректної роботи шифру є оборотність матриці ключа за вибраним модулем, оскільки саме це дозволяє виконати зворотну операцію — дешифрування повідомлення.

У даній реалізації використовується український алфавіт, який містить 32 літери, а також додатково враховується пробіл. Таким чином, усі операції виконуються за модулем 33.

Результати дослідження

Програма реалізована у вигляді класу HillCipher, який відповідає за всю логіку шифрування та дешифрування. Під час створення об’єкта класу користувач передає матрицю ключа, після чого програма автоматично визначає її розмір та обчислює визначник. Для перевірки можливості використання матриці як ключа виконується перевірка умови НСД(det, 33) = 1. Якщо ця умова не виконується, матриця вважається необоротною і шифрування стає неможливим.

Перед шифруванням вхідний текст перетворюється у числовий вигляд. Кожній літері українського алфавіту ставиться у відповідність число від 0 до 31, а пробілу — число 32. У разі, якщо довжина тексту не кратна розміру матриці ключа, текст доповнюється пробілами до необхідної довжини. Це дозволяє коректно розбити текст на блоки та виконати матричне множення.

Процес шифрування полягає у множенні кожного числового блоку на матрицю ключа з подальшим обчисленням результату за модулем 33. Отримані числові значення перетворюються назад у символи, формуючи зашифрований текст. Для можливості коректного дешифрування програма зберігає початкову довжину повідомлення.

Дешифрування виконується з використанням оберненої матриці ключа. Програма автоматично обчислює обернену матрицю для випадків 2×2 та 3×3, після чого виконує зворотне множення блоків.

Після завершення операції зайві символи, що були додані під час шифрування, видаляються, і користувач отримує початковий текст.

Криптостійкість шифру Гілла

Незважаючи на математичну строгість та елегантність, шифр Гілла не вважається криптостійким за сучасними стандартами інформаційної безпеки. Основним його недоліком є лінійність алгоритму. Якщо зловмисник отримає достатню кількість пар «відкритий текст – зашифрований текст», він зможе скласти систему лінійних рівнянь і відновити матрицю ключа. Таким чином, шифр є вразливим до атак з відомим відкритим текстом.

Крім того, відсутність механізмів випадковості, таких як ініціалізаційні вектори або нелінійні перетворення, значно знижує рівень захисту. Саме тому шифр Гілла сьогодні використовується переважно в навчальних цілях для демонстрації застосування лінійної алгебри в криптографії, а не для реального захисту конфіденційних даних.

Повний код програми:

```
class HillCipher:
    alphabet = 'АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ'
    def __init__(self, key_matrix):
        self.key_matrix = key_matrix
        self.n = len(key_matrix)
    def _text_to_numbers(self, text):
        numbers = []
        for char in text.upper():
            if char in self.alphabet:
                numbers.append(self.alphabet.index(char))
            elif char == ' ':
                numbers.append(32)
        return numbers
    def _numbers_to_text(self, numbers):
        text = ''
        for num in numbers:
            if num < len(self.alphabet):
                text += self.alphabet[num]
            elif num == 32:
                text += ' '
        return text
    def _matrix_multiply(self, matrix, vector):
        result = []
        for i in range(len(matrix)):
            s = 0
            for j in range(len(vector)):
                s += matrix[i][j] * vector[j]
            result.append(s % 33)
        return result
    def encrypt(self, text):
        numbers = self._text_to_numbers(text)
        self.original_length = len(numbers)

        while len(numbers) % self.n != 0:
            numbers.append(32)
        encrypted = []
        for i in range(0, len(numbers), self.n):
            block = numbers[i:i+self.n]
            encrypted.extend(self._matrix_multiply(self.key_matrix, block))
```

```
        return self._numbers_to_text(encrypted)
def get_text_length_without_spaces(text):
    alphabet = 'АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ'
    count = 0
    for char in text.upper():
        if char in alphabet:
            count += 1
    return count
def determine_matrix_size(text):
    length = get_text_length_without_spaces(text)
    if length % 3 == 0:
        return 3
    elif length % 2 == 0:
        return 2
    else:
        pad2 = (2 - length % 2) % 2
        pad3 = (3 - length % 3) % 3
        return 2 if pad2 <= pad3 else 3
def input_matrix(size):
    print(f"\nВведіть елементи матриці {size}x{size}:")
    matrix = []
    for i in range(size):
        row = []
        for j in range(size):
            while True:
                try:
                    value = int(input(f"Елемент [{i + 1}][{j + 1}]: "))
                    row.append(value)
                    break
                except ValueError:
                    print("Помилка! Введіть ціле число.")
        matrix.append(row)
    return matrix
if __name__ == "__main__":
    print("=" * 50)
    print(" ШИФР ГІЛЛА (ТІЛЬКИ ШИФРУВАННЯ)")
    print("=" * 50)
    text = input("Введіть текст для шифрування: ").strip()
    if not text:
        print("Текст не може бути порожнім!")
        exit()
    matrix_size = determine_matrix_size(text)
    text_length = get_text_length_without_spaces(text)
    print(f"\nДовжина тексту (без пробілів): {text_length}")
    print(f"Обраний розмір матриці: {matrix_size}x{matrix_size}")
    key_matrix = input_matrix(matrix_size)
    cipher = HillCipher(key_matrix)
    encrypted = cipher.encrypt(text)
    print("\nОригінальний текст:", text)
    print("Зашифрований текст:", encrypted)
    print(f"Оригінальна довжина: {cipher.original_length}")
```

Приклади роботи програми:

```
================================================
 ШИФР ГІЛЛА (ТІЛЬКИ ШИФРУВАННЯ)
================================================
Введіть текст для шифрування: кіт

Довжина тексту (без пробілів): 3
Обраний розмір матриці: 3x3

Введіть елементи матриці 3x3:
Елемент [1][1]: 1
Елемент [1][2]: 6
Елемент [1][3]: 7
Елемент [2][1]: 0
Елемент [2][2]: 1
Елемент [2][3]: 8
Елемент [3][1]: 0
Елемент [3][2]: 0
Елемент [3][3]: 1

Оригінальний текст: кіт
Зашифрований текст: ГТТ
Оригінальна довжина: 3
```

При шифрувані слова «кіт» використовується ключ-матриця 3 на 3.

При шифруваня слова «молоко» теж використовується матриця 3 на 3, але слово складається з 6 символів.

```
================================================
 ШИФР ГІЛЛА (ТІЛЬКИ ШИФРУВАННЯ)
================================================
Введіть текст для шифрування: молоко

Довжина тексту (без пробілів): 6
Обраний розмір матриці: 3x3

Введіть елементи матриці 3x3:
Елемент [1][1]: 1
Елемент [1][2]: 5
Елемент [1][3]: 8
Елемент [2][1]: 0
Елемент [2][2]: 1
Елемент [2][3]: 4
Елемент [3][1]: 0
Елемент [3][2]: 0
Елемент [3][3]: 1

Оригінальний текст: молоко
Зашифрований текст: ШЇЛБРО
Оригінальна довжина: 6
```

А для шифрування слова «ні» буде використовуватись матриця 2 на 2.

```
================================================
 ШИФР ГІЛЛА (ТІЛЬКИ ШИФРУВАННЯ)
================================================
Введіть текст для шифрування: ні

Довжина тексту (без пробілів): 2
Обраний розмір матриці: 2x2

Введіть елементи матриці 2x2:
Елемент [1][1]: 4 8 16 5
Помилка! Введіть ціле число.
Елемент [1][1]: 4
Елемент [1][2]: 8
Елемент [2][1]: 16
Елемент [2][2]: 5

Оригінальний текст: ні
Зашифрований текст: ФЬ
Оригінальна довжина: 2
```

Для слів довжина яких не кратна 2 або 3 діє система автоматичного оповнення тексту до кратної довжини. Для розв'язання цієї проблеми в програмі використовується функція determine_matrix_size, яка спочатку визначає довжину тексту без урахування пробілів, а потім автоматично обирає такий розмір матриці, за якого кількість необхідних символів доповнення буде мінімальною. В ролі символу доповнення виступає пробіл який відповідає значенню «32». Оскільки пробіли додаються лише в кінець повідомлення це не заважає при дешифровці, а також вони автоматично видаляються при виведенні завдяки тому, що була збереження довжина оригінального повідомлення.

```
================================================
 ШИФР ГІЛЛА (ТІЛЬКИ ШИФРУВАННЯ)
================================================
Введіть текст для шифрування: матір

Довжина тексту (без пробілів): 5
Обраний розмір матриці: 2x2

Введіть елементи матриці 2x2:
Елемент [1][1]: 4
Елемент [1][2]: 8
Елемент [2][1]: 16
Елемент [2][2]: 48

Оригінальний текст: матір
Зашифрований текст: ЮХІТЕЖ
Оригінальна довжина: 5
```

Висновок

У ході виконання даної роботи було реалізовано алгоритм шифру Гілла з підтримкою українського алфавіту. Програма дозволяє виконувати як шифрування, так і дешифрування текстових повідомлень, автоматично визначає оптимальний розмір матриці ключа та перевіряє її коректність. Робота дала змогу закріпити знання з лінійної алгебри, модульної арифметики та основ криптографії, а також на практиці продемонструвала обмеження класичних методів шифрування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Математичні методи криптології: Навчальний посібник [Електронний ресурс] (Для студентів техн. спец. вищ. навч. закл.) / [А.Д. Кожухівський, І.Д. Горбенко, Г.І. Гайдур, О.А. Кожухівська, В.В. Марченко]; М-во освіти і науки України, Державний університет телекомунікацій. Київ: ДУТ, 2021 – 244 с.

2. Ігнатович А.О. Критерій ефективності для визначення стійкості блокових шифрівна основі внесених змін статистичних характеристик шифрованого тексту / Ігнатович А.О., Глухова О.В., Лозинський А.Я., Яремчук Р.І. // ACIT'5 "Сучасні комп'ютерні інформаційні технології". ТНЕУ. – Тернопіль. 22-23 травня 2015. – С. 167-168.

Гуменна Ольга Віталіївна – студентка групи 1ЕХКБ – 25б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: olgahymenna2510@gmail.com

Науковий керівник: ***Клєопа Ірина Анатоліївна*** – pHd, доцент, кафедра вищої математики, Вінницький національний технічний університет, м. Вінниця, Хмельницьке шосе, 95, e-mail: paceka08@vntu.edu.ua

Humenna Olha Vitaliivna – student of group 1EHKB – 25b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: olgahymenna2510@gmail.com

Scientific supervisor: ***Klieopa Iryna Anatoliivna*** – PhD, Associate Professor, Department of Higher Mathematics, Vinnytsia National Technical University, Vinnytsia, Khmelnytske Shosse, 95, e-mail: paceka08@vntu.edu.ua