УДК 681

А.Ю. Калюжна
І.А. Клєопа

ВИКОРИСТАННЯ ДЕТЕРМІНАНТІВ У ЗВОРОТНИХ МАТРИЦЯХ ДЛЯ ШИФРУВАННЯ ПОВІДОМЛЕНЬ

Вінницький національний технічний університет

Анотація

У роботі розглянуто використання детермінантів при обчисленні зворотних матриць у задачах шифрування повідомлень. Основну увагу приділено математичним засадам матричних криптографічних алгоритмів, зокрема умовам існування зворотної матриці та ролі детермінанта в процесах шифрування й дешифрування. Показано, що значення детермінанта визначає можливість коректного відновлення зашифрованого повідомлення. Наведено приклади застосування матричних операцій у криптографії та окреслено можливості їх програмної реалізації.

Ключові слова: криптографія, шифрування повідомлень, матриця, зворотна матриця, детермінант, матричні операції

Abstract

The paper considers the use of determinants in computing inverse matrices for message encryption. Particular attention is paid to the mathematical foundations of matrix-based cryptographic algorithms, including the conditions for the existence of an inverse matrix and the role of the determinant in encryption and decryption processes. It is shown that the value of the determinant determines the possibility of correct recovery of encrypted messages. Examples of applying matrix operations in cryptography are presented, and the prospects for their software implementation are outlined.

Keywords: cryptography, message encryption, matrix, inverse matrix, determinant, matrix operations

Вступ

Сучасна криптографія широко використовує математичні методи для забезпечення конфіденційності та цілісності інформації. Одним із таких методів є застосування матричних перетворень, що базуються на апараті лінійної алгебри. Особливу роль у цих алгоритмах відіграють зворотні матриці, обчислення яких безпосередньо пов’язане з поняттям детермінанта.

Детермінант матриці є ключовим елементом, що визначає можливість існування зворотної матриці та, відповідно, коректність процесів шифрування і дешифрування повідомлень. У матричних криптографічних алгоритмах значення детермінанта впливає на надійність шифрування та відновлення початкового повідомлення. Тому дослідження використання детермінантів у зворотних матрицях є актуальним завданням у контексті математичних основ криптографії.

Результати дослідження

Розроблено програмне забезпечення мовою Python (з використанням бібліотеки NumPy) для реалізації криптографічного алгоритму Хілла.

Основними умовами є:

- Шифрування виконується методом множення цифрових векторів тексту на матрицю-ключ.
- Дешифрування реалізується через знаходження оберненої матриці (обчислення детермінанта та його мультиплікативної інверсії за модулем довжини алфавіту).

hill_cipher.py:

```
import numpy as np

# --- НАЛАШТУВАННЯ ---
ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ " # 27 символів
MODULO = len(ALPHABET)
CHAR_TO_INT = {c: i for i, c in enumerate(ALPHABET)}
```

```
INT_TO_CHAR = {i: c for i, c in enumerate(ALPHABET)}

# --- МАТЕМАТИКА ---

def egcd(a, b):
    if a == 0: return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1: return None
    else: return x % m

def get_matrix_inverse_mod(M, m):
    det = int(np.round(np.linalg.det(M)))
    det_inv = modinv(det, m)
    if det_inv is None: return None

    adj = np.linalg.inv(M) * det
    adj = np.round(adj).astype(int)
    matrix_inv = (det_inv * adj) % m
    return np.round(matrix_inv).astype(int) % m

# --- ВИГЛЯД ДІЙ ---

def encrypt(message, key_matrix):
    n = key_matrix.shape[0]
    # Доповнення пробілами
    while len(message) % n != 0:
        message += " "

    numbers = [CHAR_TO_INT[c] for c in message]
    encrypted_numbers = []

    print(f"\n{'='*10} ЕТАП ШИФРУВАННЯ {'='*10}")
    print(f"Матриця ключа:\n{key_matrix}\n")

    for i in range(0, len(numbers), n):
        # 1. Беремо блок
        vector = np.array(numbers[i:i+n])
        chars = "".join([INT_TO_CHAR[num] for num in vector])

        # 2. Множимо (лінійна алгебра)
        dot_product = np.dot(key_matrix, vector)

        # 3. Беремо модуль
        encrypted_vector = dot_product % MODULO
        encrypted_chars = "".join([INT_TO_CHAR[num] for num in encrypted_vector])

        encrypted_numbers.extend(encrypted_vector)

        # ВИВІД ДІЙ
        print(f"🔻 Блок '{chars}': {vector}")
```

```
        print(f"   ✕ Множення (Vector * Matrix): {dot_product}")
        print(f"   ÷ Mod {MODULO}: {encrypted_vector} -> '{encrypted_chars}'")
        print("-" * 30)

    return encrypted_numbers

def decrypt(encrypted_numbers, key_matrix):
    n = key_matrix.shape[0]
    inv_key_matrix = get_matrix_inverse_mod(key_matrix, MODULO)

    if inv_key_matrix is None: return "ПОМИЛКА МАТРИЦІ"

    print(f"\n{'='*10} ЕТАП ДЕШИФРУВАННЯ {'='*10}")
    print(f"Обернена матриця (Ключ^-1 mod 27):\n{inv_key_matrix}\n")

    decrypted_numbers = []
    for i in range(0, len(encrypted_numbers), n):
        # 1. Беремо зашифрований вектор
        vector = np.array(encrypted_numbers[i:i+n])

        # 2. Множимо на обернену матрицю
        dot_product = np.dot(inv_key_matrix, vector)

        # 3. Беремо модуль
        decrypted_vector = dot_product % MODULO

        # Корекція для numpy (іноді залишає float)
        decrypted_vector = np.round(decrypted_vector).astype(int)

        chars = "".join([INT_TO_CHAR[num] for num in decrypted_vector])
        decrypted_numbers.extend(decrypted_vector)

        # ВИВІД ДІЙ
        print(f"🧩 Шифр-вектор: {vector}")
        print(f"   ✕ Множення (Vector * InvMatrix): {dot_product}")
        print(f"   ÷ Mod {MODULO}: {decrypted_vector} -> '{chars}'")
        print("-" * 30)

    return "".join([INT_TO_CHAR[int(n)] for n in decrypted_numbers])

# --- ЗАПУСК ---

if __name__ == "__main__":
    # Матриця з детермінантом 22 (безпечна для mod 27)
    valid_key_matrix = np.array([
        [1, 2, 3],
        [0, 4, 5],
        [1, 0, 6]
    ])

    user_input = input("\nВведіть слово (наприклад, MATH): ")
    clean_message = "".join([c for c in user_input.upper() if c in ALPHABET])

    if clean_message:
```

```
    # Запуск процесу
    enc_nums = encrypt(clean_message, valid_key_matrix)
    dec_text = decrypt(enc_nums, valid_key_matrix)

    print(f"\n✅ ФІНАЛЬНИЙ РЕЗУЛЬТАТ: {dec_text}")
else:
    print("Введіть хоча б одну англійську літеру.")
```

Нижче на рис.1 покажемо результат нашого дослідження за допомогою використання детермінантів при обчисленні зворотних матриць у задачах шифрування повідомлень.

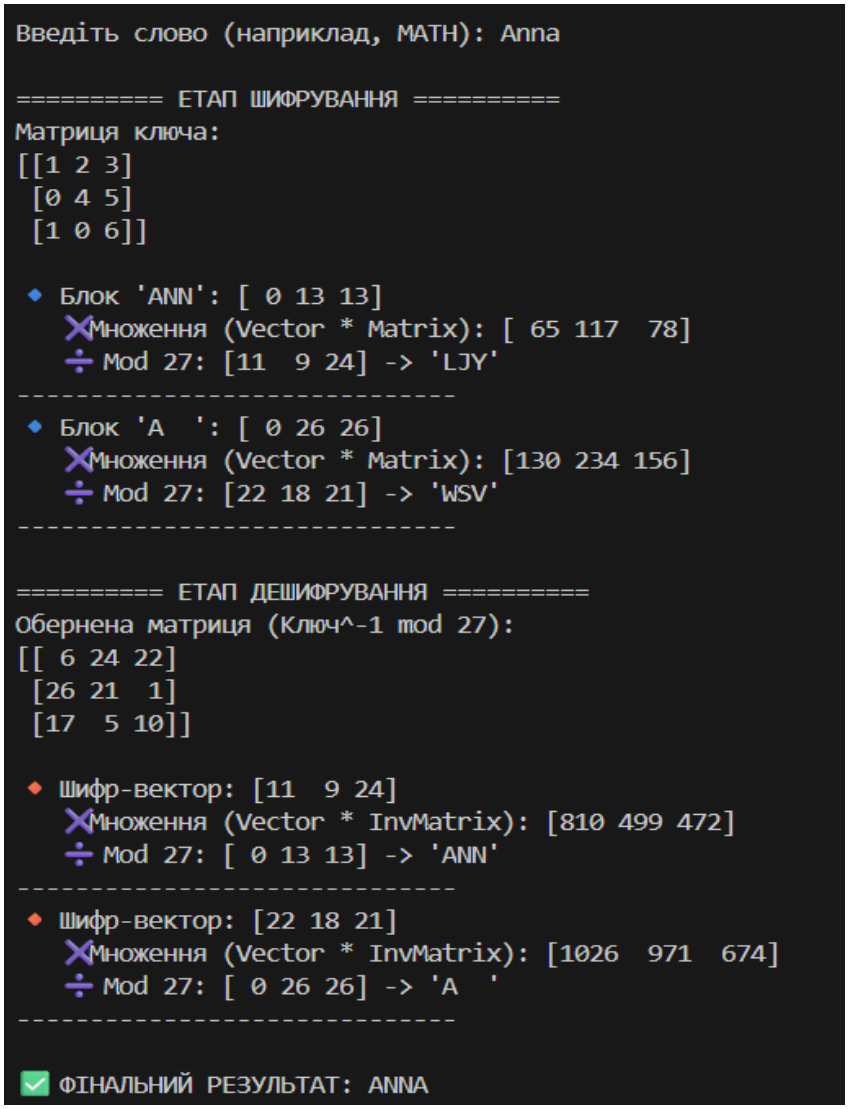

Рис. 1. – Задача шифрування повідомлень.

Висновок

У роботі розглянуто роль детермінанта у процесі обчислення зворотних матриць для шифрування повідомлень. Показано, що ненульовий детермінант є необхідною умовою існування зворотної матриці та забезпечує коректність процедур шифрування і дешифрування в матричних криптографічних алгоритмах. Застосування операцій над матрицями дозволяє ефективно реалізовувати алгоритми шифрування, що ґрунтуються на математичному апараті лінійної алгебри.

Отримані результати підтверджують доцільність використання детермінантів у криптографічних задачах та їх практичну значущість для програмної реалізації алгоритмів захисту інформації. Подальші

дослідження можуть бути спрямовані на аналіз стійкості матричних шифрів та оптимізацію обчислень у прикладних криптографічних системах.

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. Trappe W., Washington L. C. Introduction to Cryptography with Coding Theory. — Pearson, 2006.
3. Strang G. Linear Algebra and Its Applications. — Brooks/Cole, 2016.
4. Lay D. C. Linear Algebra and Its Applications. — Pearson Education, 2015.
5. Stroustrup B. The C++ Programming Language. — Addison-Wesley, 2013.
6. Deitel P., Deitel H. C++ How to Program. — Pearson Education, 2017.

Калюжна Анна Юріївна – студентка групи 1КІТС – 25б, факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, м. Вінниця, e-mail: kazhnaanna@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

Kayluzhna Anna Yurievna – student of group 1КІТС – 25b, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: kazhnaanna@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