Міністерство освіти і науки України

Національний технічний університет «ХПІ»

Навчально-науковий інститут комп’ютерних наук та інформаційних технологій

Кафедра комп’ютерної інженерії та програмування

ЗВІТ

з лабораторної роботи № 2

з дисципліни «Сучасні технології безпечного програмування»

«СИМЕТРИЧНЕ ШИФРУВАННЯ. АЛГОРИТМ AES»

Виконав:

студент гр. КН-Н922б

Кулик Д.І.

Перевірив:

Бульба С. С.

Харків – 2022

Мета роботи: Дослідити принципи роботи симетричного шифрування на прикладі алгоритму AES.

Індивідуальне завдання

Реалізувати алгоритм симетричного шифрування AES (будь-якої версії - 128 або 256).

Довести коректність роботи реалізованого алгоритму шляхом порівняння результатів з існуючими реалізаціями (напр. сайтом-утилітою https://cryptii.com).

Хід роботи

Розширений стандарт шифрування (AES) — це симетричний блоковий шифр, обраний урядом США для захисту секретної інформації. AES реалізовано в програмному та апаратному забезпеченні по всьому світу для шифрування конфіденційних даних. Це має важливе значення для комп'ютерної безпеки уряду, кібербезпеки та захисту електронних даних.

Важливі фрагменти програми

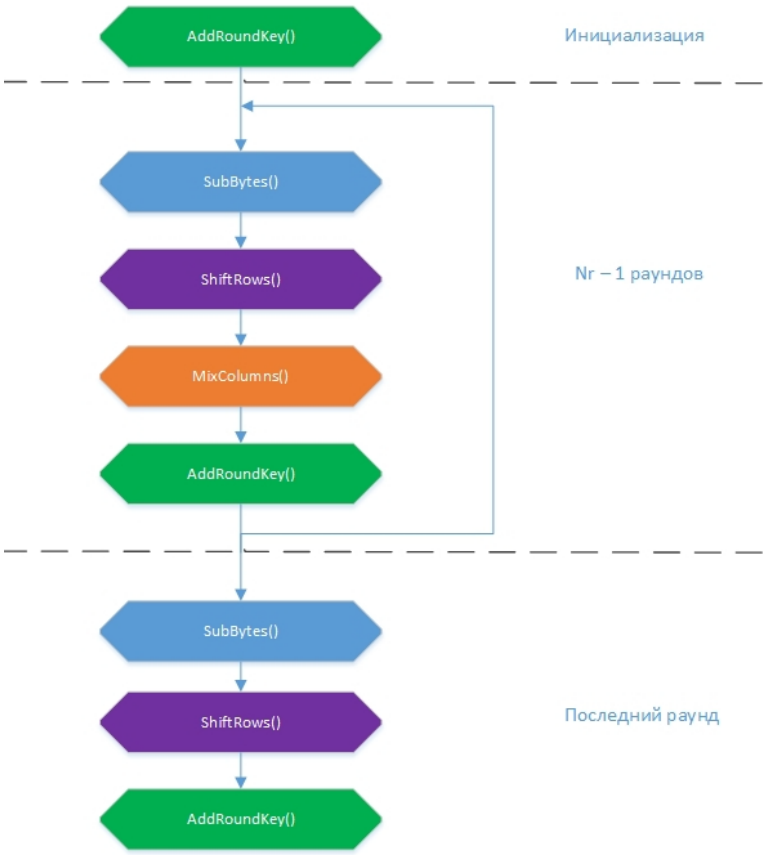

Рисунок 1 – Загальна схема шифрування

```
def add_round_key(s, k):
    for i in range(4):
        for j in range(4):
            s[i][j] ^= k[i][j]
def sub_bytes(s):
    for i in range(4):
        for j in range(4):
            s[i][j] = s_box[s[i][j]]
def shift_rows(s):
    s[0][1], s[1][1], s[2][1], s[3][1] = s[1][1], s[2][1], s[3][1], s[0][1]
    s[0][2], s[1][2], s[2][2], s[3][2] = s[2][2], s[3][2], s[0][2], s[1][2]
    s[0][3], s[1][3], s[2][3], s[3][3] = s[3][3], s[0][3], s[1][3], s[2][3]
def mix_columns(s):
    for i in range(4):
        mix_single_column(s[i])
```

Рисунок 2 – Методи для шифрування

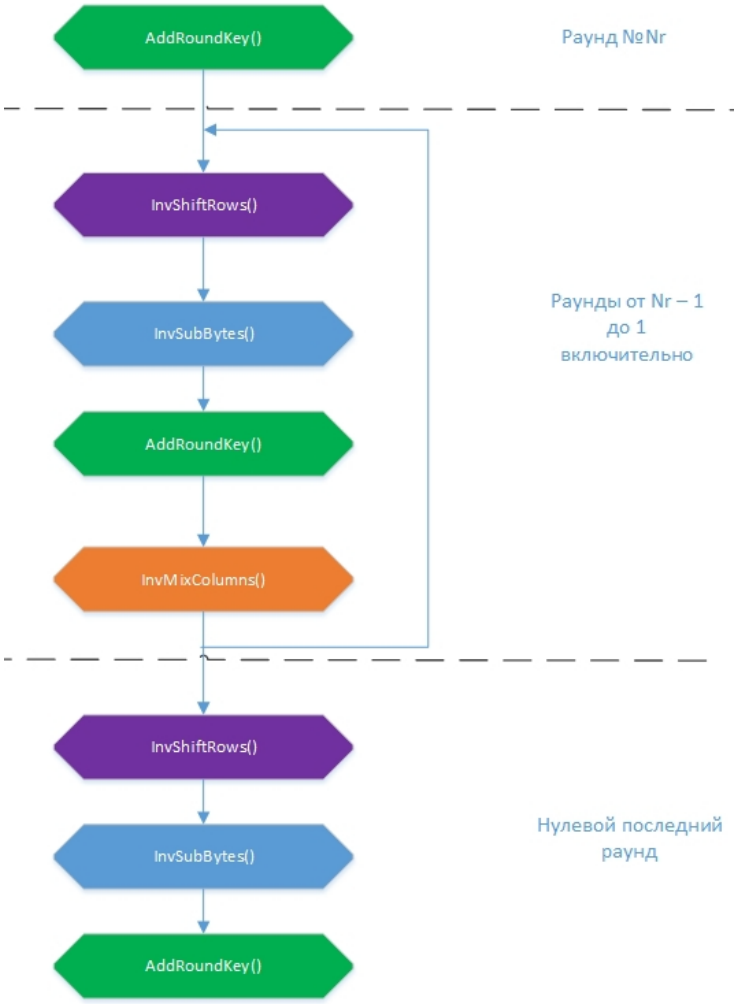

Рисунок 3 – Загальна схема дешифрування

```
def inv_sub_bytes(s):
    for i in range(4):
        for j in range(4):
            s[i][j] = inv_s_box[s[i][j]]

def inv_shift_rows(s):
    s[0][1], s[1][1], s[2][1], s[3][1] = s[3][1], s[0][1], s[1][1], s[2][1]
    s[0][2], s[1][2], s[2][2], s[3][2] = s[2][2], s[3][2], s[0][2], s[1][2]
    s[0][3], s[1][3], s[2][3], s[3][3] = s[1][3], s[2][3], s[3][3], s[0][3]

def inv_mix_columns(s):
    # see Sec 4.1.3 in The Design of Rijndael
    for i in range(4):
        u = xtime(xtime(s[i][0] ^ s[i][2]))
        v = xtime(xtime(s[i][1] ^ s[i][3]))
        s[i][0] ^= u
        s[i][1] ^= v
        s[i][2] ^= u
        s[i][3] ^= v

    mix_columns(s)
```

Рисунок 4 – Методи для дешифрування

```
    def encrypt_block(self, plaintext):

        assert len(plaintext) == 16

        plain_state = bytes2matrix(plaintext)

        add_round_key(plain_state, self._key_matrices[0])

        for i in range(1, self.n_rounds):
            sub_bytes(plain_state)
            shift_rows(plain_state)
            mix_columns(plain_state)
            add_round_key(plain_state, self._key_matrices[i])

        sub_bytes(plain_state)
        shift_rows(plain_state)
        add_round_key(plain_state, self._key_matrices[-1])

        return matrix2bytes(plain_state)
```

Рисунок 5 – Реалізація шифрування

```python
def decrypt_block(self, ciphertext):

    assert len(ciphertext) == 16

    cipher_state = bytes2matrix(ciphertext)

    add_round_key(cipher_state, self._key_matrices[-1])
    inv_shift_rows(cipher_state)
    inv_sub_bytes(cipher_state)

    for i in range(self.n_rounds - 1, 0, -1):
        add_round_key(cipher_state, self._key_matrices[i])
        inv_mix_columns(cipher_state)
        inv_shift_rows(cipher_state)
        inv_sub_bytes(cipher_state)

    add_round_key(cipher_state, self._key_matrices[0])

    return matrix2bytes(cipher_state)
```

Рисунок 6 – Реалізація дешифрування

Результати роботи програми

```
C:\Users\Daniil\PycharmProjects\stbp\Scripts\python.exe C:/Users/Daniil/PycharmProjects/stbp/LABS/kulyk02/main.py
Заданий текст          :b'Daniil Kulyk'
Текст у байтах         :44616e69696c204b756c796b
Зашифрований текст     :b37913ec611d1d3acb6cf15d
Розшифрований текст    :b'Daniil Kulyk'
```

Рисунок 7 – Результат виконання програми

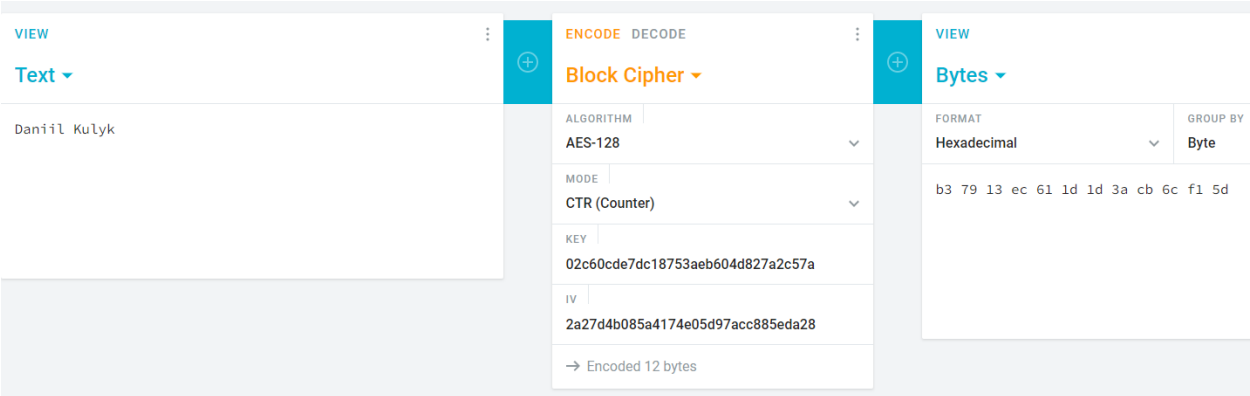

Рисунок 13 – Перевірка результату шифрування за допомогою сайту-утиліти https://cryptii.com

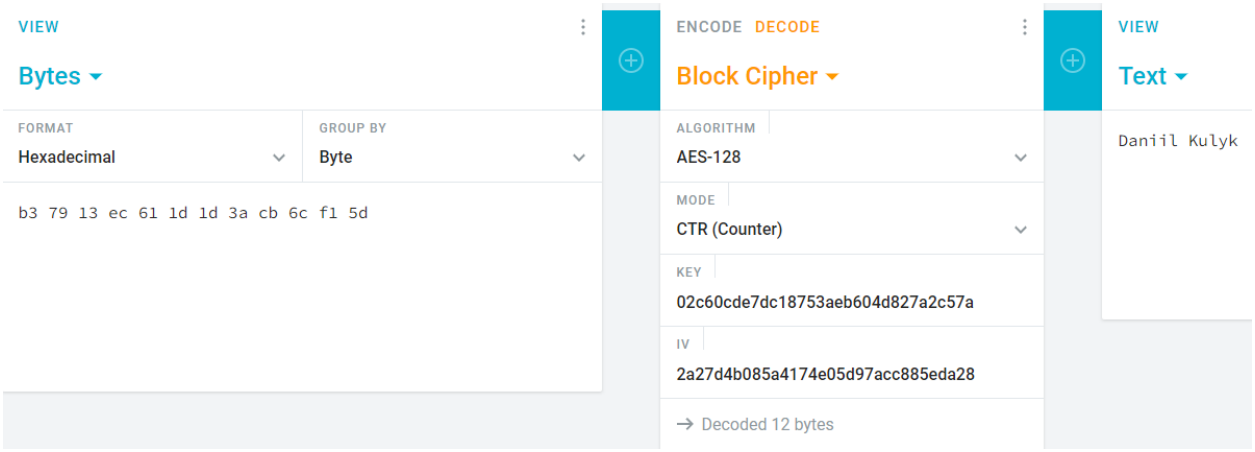

Рисунок 14 – Перевірка результату дешифрування за допомогою сайту-утиліти https://cryptii.com

При порівнянні можемо побачити що результат виконання реалізацій алгоритму однаковий.

Висновки: в результаті виконання лабораторної роботи було досліджено принципи роботи симетричного шифрування на прикладі алгоритму AES. В результаті порівняння власної реалізації алгоритму з вже реалізованими була виявлена ідентичність роботи, що доводить коректність першого.