ЛАБОРАТОРНАЯ РАБОТА №2

“Блочное симметричное шифрование”

по дисциплине

‘Информационная безопасность’

Вариант 3Д

Выполнил:
Соболев Иван
Александрович
Группа: P34312

Преподаватель:
Маркина Татьяна
Анатольевна

Санкт-Петербург, 2024

Цель работы

Изучение структуры и основных принципов работы современных алгоритмов блочного симметричного шифрования, приобретение навыков программной реализации блочных симметричных шифров.

Программные и аппаратные средства

Для выполнения лабораторной работы был использован компьютер со следующими характеристиками:

- Процессор: Apple M2
- Видеокарта: Apple M2
- Объем оперативной памяти: 8GB
- Использована операционная система: macOS 14.4.1
- Версия Python: 3.12

Задание

Реализовать систему симметричного блочного шифрования, позволяющую шифровать и дешифровать файл на диске с использованием заданного блочного шифра в заданном режиме шифрования.

Алгоритм: RC6
Режим шифрования: OFB

Листинг разработанной программы

io_utils.py

```
import os
from typing import Optional, Any

ENCRYPT_ACTION = "enc"
DECRYPT_ACTION = "dec"

"""Функция для вывода ошибок"""
def print_red(message: str) -> None:
  print(f"\033[91m{message}\033[0m")


"""Функция чтения имени файла с открытым текстом из консоли"""
def read_filename() -> Optional[str]:
  try:
    filename = input("Введите названия файла с текстом для шифрования: ")
    if not filename.strip():
      print_red("Ошибка: имя файла не должно быть пустым.")
      return None

    if not os.path.isfile(filename):
      print_red(f"Ошибка: файл '{filename}' не найден.")
      return None
    return filename
```

```
    except Exception as e:
        print_red(f'Ошибка чтения из файла: {str(e)}')
        return None


"""Функция выбора действия"""
def decide_action() -> Any | None:
    print('Выберите действие: \n'
          '1. Зашифровать файл\n'
          '2. Расшифровать файл')
    try:
        option = int(input("Введите номер: "))
        if option not in [1, 2]:
            print_red(f'Ошибка: опция {option} недоступна!')
            return None
        if option == 1:
            return ENCRYPT_ACTION
        elif option == 2:
            return DECRYPT_ACTION
        else:
            return None
    except Exception:
        print_red(f'Ошибка: неверный ввод. Необходимо ввести целое число.')
        return None
```

RC6.py

```
import struct

w = 32  # длина слова в битах
r = 20  # число раундов
b = 16  # длина ключа
P32 = 0xB7E15163
Q32 = 0x9E3779B9
LEFT = "LEFT"
RIGHT = "RIGHT"

"""Функция циклического сдвига"""
def cyclic_shift(x, y, direction):
    y = y % w
    if direction == LEFT:
        return ((x << y) & (2 ** w - 1)) | (x >> (w - y))
    if direction == RIGHT:
        return (x >> y) | ((x << (w - y)) & (2 ** w - 1))


"""Функция подготовки ключа"""
def rc6_key_schedule(key):
    L = [0] * (b // 4)
    for i in range(b - 1, -1, -1):
        L[i // 4] = (L[i // 4] << 8) + key[i]
```

```
    S = [P32]
    for i in range(1, 2 * r + 4):
        S.append((S[i - 1] + Q32) % 2 ** w)

    A = 0
    B = 0
    i = 0
    j = 0
    for _ in range(3 * max(b // 4, 2 * r + 4)):
        A = cyclic_shift((S[i] + A + B) % 2 ** w, 3, LEFT)
        S[i] = cyclic_shift((S[i] + A + B) % 2 ** w, 3, LEFT)
        B = cyclic_shift((L[j] + A + B) % 2 ** w, (A + B) % w, LEFT)
        L[j] = cyclic_shift((L[j] + A + B) % 2 ** w, (A + B) % w, LEFT)
        i = (i + 1) % (2 * r + 4)
        j = (j + 1) % (b // 4)
    return S


"""Функция RC6 шифрования"""
def rc6_encrypt(plaintext, S):
    A, B, C, D = struct.unpack('<4I', plaintext)

    B = (B + S[0]) % 2 ** w
    D = (D + S[1]) % 2 ** w

    for i in range(1, r + 1):
        t = cyclic_shift(B * (2 * B + 1) % 2 ** w, 5, LEFT)  # 5 = log2(32)
        u = cyclic_shift(D * (2 * D + 1) % 2 ** w, 5, LEFT)
        A = (cyclic_shift(A ^ t, u, LEFT) + S[2 * i]) % 2 ** w
        C = (cyclic_shift(C ^ u, t, LEFT) + S[2 * i + 1]) % 2 ** w
        A, B, C, D = B, C, D, A
    A = (A + S[2 * r + 2]) % 2 ** w
    C = (C + S[2 * r + 3]) % 2 ** w

    return struct.pack('<4I', A, B, C, D)
```

OFB.py

```
from RC6 import rc6_key_schedule, rc6_encrypt

"""Функция для заполнения текста до нужной длины
  Заполняет числом равным длине финального паддинга"""
def pad(plaintext):
    padding_len = 16 - (len(plaintext) % 16)
    padding = bytes([padding_len] * padding_len)
    return plaintext + padding


"""Функция для удаления паддинга"""
def unpad(padded_plaintext):
    padding_len = padded_plaintext[-1]
    return padded_plaintext[:-padding_len]
```

```
"""Функция операции XOR"""
def xor_bytes(a, b):
    return bytes(x ^ y for x, y in zip(a, b))

"""Функция для шифрования в режиме OFB"""
def rc6_ofb_encrypt(key, iv, plaintext):
    S = rc6_key_schedule(key)
    ciphertext = bytearray()
    feedback = iv
    padded_plaintext = pad(plaintext)

    for i in range(0, len(padded_plaintext), 16):
        keystream = rc6_encrypt(feedback, S)
        block = padded_plaintext[i:i + 16]
        ciphertext_block = xor_bytes(block, keystream)
        ciphertext.extend(ciphertext_block)
        feedback = keystream

    return bytes(ciphertext)

"""Функция для дешифрования в режиме OFB"""
def rc6_ofb_decrypt(key, iv, ciphertext):
    S = rc6_key_schedule(key)
    plaintext = bytearray()
    feedback = iv

    for i in range(0, len(ciphertext), 16):
        keystream = rc6_encrypt(feedback, S)
        block = ciphertext[i:i + 16]
        plaintext_block = xor_bytes(block, keystream)
        plaintext.extend(plaintext_block)
        feedback = keystream

    return unpad(bytes(plaintext))
```

main.py

```
import os
from OFB import rc6_ofb_encrypt, rc6_ofb_decrypt
from io_utils import decide_action, ENCRYPT_ACTION, DECRYPT_ACTION, read_filename, print_red

key = b'SuperSecretKey123'
encrypted_file = 'encrypted.bin'
decrypted_file = 'decrypted.txt'


"""Функция для чтения и шифрования открытого текста"""
def encrypt_file(symmetric_key, input_file, output_file):
    global iv
    try:
        iv = generate_iv()
```

```
        with open(input_file, 'rb') as f:
            plaintext = f.read()

        if not plaintext:
            raise ValueError("Файл пустой.")

        ciphertext = rc6_ofb_encrypt(symmetric_key, iv, plaintext)

        with open(output_file, 'wb') as f:
            f.write(iv + ciphertext)

        print(f"Файл зашифрован и сохранен в: {output_file}")

    except (ValueError) as e:
        print_red(f"Ошибка: {e}")
    except Exception as e:
        print_red(f"Произошла непредвиденная ошибка: {e}")

"""Функция для расшифрования зашифрованного текста и записи результата"""
def decrypt_file(symmetric_key, input_file, output_file):
    global iv

    try:
        if not os.path.exists(input_file):
            raise FileNotFoundError(f"Файл {input_file} не найден!")

        if os.path.getsize(input_file) == 0:
            raise ValueError(f"Файл {input_file} пустой.")

        with open(input_file, 'rb') as f:
            iv = f.read(16)
            ciphertext = f.read()

        decrypted_plaintext = rc6_ofb_decrypt(symmetric_key, iv, ciphertext)

        with open(output_file, 'wb') as f:
            f.write(decrypted_plaintext)

        print(f"Файл расшифрован и сохранен в: {output_file}")

    except (FileNotFoundError, ValueError) as e:
        print_red(f"Ошибка: {e}")
    except Exception as e:
        print_red(f"Произошла непредвиденная ошибка: {e}")

"""Функция для генерации вектора инициализации"""
def generate_iv():
    return os.urandom(16)
```

```
def main():
    action = decide_action()
    if action == ENCRYPT_ACTION:
        input_file = read_filename()
        if input_file is not None:
            encrypt_file(key, input_file, encrypted_file)

    elif action == DECRYPT_ACTION:
        decrypt_file(key, encrypted_file, decrypted_file)


if __name__ == "__main__":
    main()

iv = None
```

Результаты работы программы

Исходный текст:
У-у-у-у-у-гу-гуг-гуу! О, гляньте на меня, я погибаю. Вьюга в подворотне ревёт мне отходную, и я вою с ней. Пропал я, пропал. Негодяй в грязном колпаке – повар столовой нормального питания служащих центрального совета народного хозяйства – плеснул кипятком и обварил мне левый бок.
Какая гадина, а ещё пролетарий. Господи, боже мой – как больно! До костей проело кипяточком. Я теперь вою, вою, да разве воем поможешь.

Шифрование текста:

```
Выберите действие:
1. Зашифровать файл
2. Расшифровать файл
Введите номер: 1
Введите названия файла с текстом для шифрования: input.txt
Файл зашифрован и сохранен в: encrypted.bin
```

Результат шифрования (содержимое файла encrypted.bin):

y ????dǴFW3o??g?F6???????J?
??@lAX?w?D)?_?r o?j?h/?m |???o?:W(Pn????~P?;?3E??Q?(??|??n_?;xf`?V6???
???0????mdk?-?_?UP??????????a(?ʖ?FU? ?=??5??d???Uj??<????E?????ca????Uì?4??(?1??t??^?D?ス
?r??◌????e??^?Lq????@Y??????D??R?????v?ɯ?\?vD

ᵠ
?Z?0?ojjá??????????:?r????\????????t4?q?F5????W?tG??????
?H??????????????????,????T????n?????|?◌????P????gNG??U??l?:?B?~%#?M?Sv?p??{X?e"?mMJ4Ѧ?K??n!?
?F?:z??!r3{?F??????q
u◌?] I ??%?????%??6?t?1????N?zG1B?M'?
J????f??4)` =??????.n.??A????]???{?????m-5??戈b!???????s????????w?????????ROL?&◌CJO?\Ju?
?R??;xP?]?{?R??3g????2??U?-?_??8^?:?i?.-??
_????x?G_???k?lT????5????????ᑭE??????S?\h??j?????mv??*?J????!?n??????????y?g

?#??
3??◌?}U]?????L?!??b??Lhq??(|x?-??????????Y????
??{?|??????????yk$?(5vw??????G?????n??r+??P??*8ő?r??#?jsu?????ůP?,??o??KN?

Дешифрование текста:

```
Выберите действие:
1. Зашифровать файл
2. Расшифровать файл
Введите номер: 2
Файл расшифрован и сохранен в: decrypted.txt
```

Дешифрованный текст (содержимое файла decrypted.txt):
У-у-у-у-у-гу-гуг-гуу! О, гляньте на меня, я погибаю. Вьюга в подворотне ревёт мне отходную, и я вою с ней. Пропал я, пропал. Негодяй в грязном колпаке – повар столовой нормального питания служащих центрального совета народного хозяйства – плеснул кипятком и обварил мне левый бок.
Какая гадина, а ещё пролетарий. Господи, боже мой – как больно! До костей проело кипяточком. Я теперь вою, вою, да разве воем поможешь.