Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №2.5
по дисциплине «Защита информации»

Студент Лукьяненко В.А.

Группа ИУ7-71Б

Преподаватель Руденкова Ю.С.

2025 г.

1 Задание

1.1 Цель работы

Цель работы: разработка алгоритма симметричного шифрования (AES). Шифрование и расшифровка архивного файла на примере архива RAR.

1.2 Содержание работы

Для выполнения данной лабораторной работы необходимо решить следующие задачи:

1. реализовать программу шифрования симметричным алгоритмом AES;
2. обеспечить шифрование и расшифровку архивного файла (RAR) с использованием разработанной программы;
3. предусмотреть работу программы с пустым и однобайтовым файлом;
4. провести тестирование программы на различных архивных файлах.

2 Теоретическая часть

Вопросы для защиты работы

1. Приведите классификацию автоматизированных систем с точки зрения требований безопасности (3 класса защищённости).

В соответствии с требованиями к безопасности автоматизированные системы классифицируются по уровням защищённости:

— **Первый класс (высший уровень)** — обеспечивает полную конфиденциальность, целостность и доступность информации. Требует строгой аутентификации пользователей, разграничения прав доступа, защиты от утечек через технические каналы.

— **Второй класс (средний уровень)** — обеспечивает конфиденциальность и целостность информации при стандартных условиях эксплуатации. Допускает ограниченные возможности доступа пользователей в зависимости от их ролей.

— **Третий класс (базовый уровень)** — минимальная защита информации, акцент на предотвращение случайных ошибок и сбоев. Применяется в системах с низкими требованиями к безопасности.

2. Виды симметричного шифрования (поточные и блочные).

Существует два основных вида симметричного шифрования:

— **Поточные шифры** — выполняют шифрование по одному символу (биту или байту) за раз. Используют псевдослучайную гамму, которая комбинируется с открытым текстом. Пример: RC4.

— **Блочные шифры** — сообщение разбивается на блоки фиксированного размера (например, 128 бит в AES), каждый блок шифруется с использованием секретного ключа. Примеры: DES, AES.

3. Особенности алгоритма шифрования архивных файлов.

Архивные файлы представляют собой двоичные данные, а не только текстовую информацию. При их шифровании необходимо учитывать следующие особенности:

— данные должны считываться и обрабатываться в бинарном режиме, чтобы сохранить структуру архива;

— при шифровании используются блоки фиксированного размера (для AES — 128 бит), поэтому необходим механизм дополнения (*padding*);

— целостность архива важна: при ошибках в расшифровке даже небольшая потеря данных может сделать файл нечитаемым;

— шифрование не влияет на степень сжатия архива, так как производится уже после упаковки;

— для повышения криптостойкости предпочтительно использовать режимы работы AES с инициализационным вектором (например, CBC).

3 Практическая часть.

Листинг 3.1 – Файл `main.py`,

```python
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import os

def pad(data: bytes) -> bytes:
    padding_len = 16 - (len(data) % 16)
    return data + bytes([padding_len] * padding_len)

def unpad(data: bytes) -> bytes:
    padding_len = data[-1]
    return data[:-padding_len]

def encrypt_file(input_file: str, output_file: str, key: bytes):
    cipher = AES.new(key, AES.MODE_CBC)
    with open(input_file, "rb") as f:
        plaintext = f.read()

    padded_data = pad(plaintext)
    ciphertext = cipher.encrypt(padded_data)

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

def decrypt_file(input_file: str, output_file: str, key: bytes):
    with open(input_file, "rb") as f:
        iv = f.read(16)
        ciphertext = f.read()

    cipher = AES.new(key, AES.MODE_CBC, iv)
    decrypted_data = cipher.decrypt(ciphertext)
    unpadded_data = unpad(decrypted_data)

    with open(output_file, "wb") as f:
        f.write(unpadded_data)


if __name__ == "__main__":
    key = b"thisisasecretkey"
```

```
39
40      encrypt_file("input.rar", "encrypted.bin", key)
41      decrypt_file("encrypted.bin", "output.rar", key)
42
43      print("Шифрование␣и␣расшифровка␣завершены.")
```

4 Пример работы программы

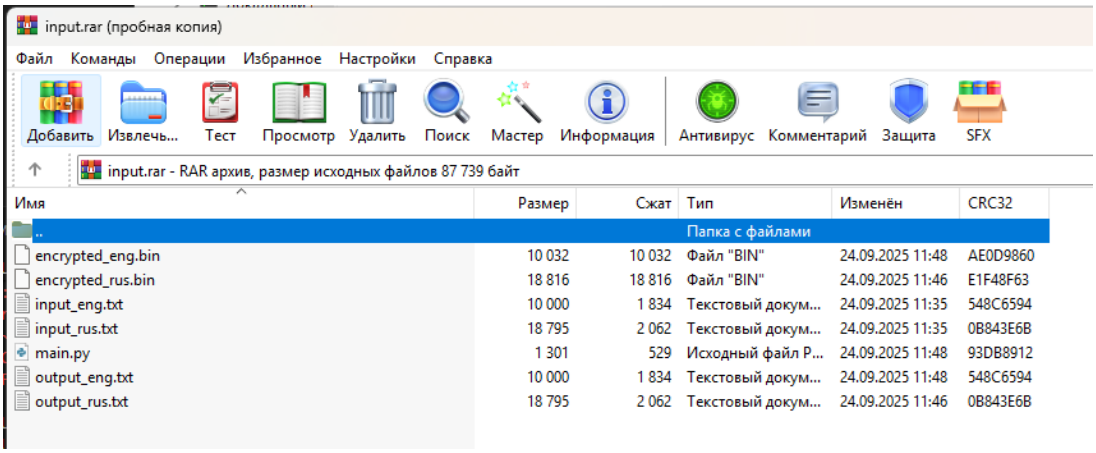

Рисунок 4.1 – Архив до шифрования

В результате шифрования получаются бинарный файл следующего содержания:

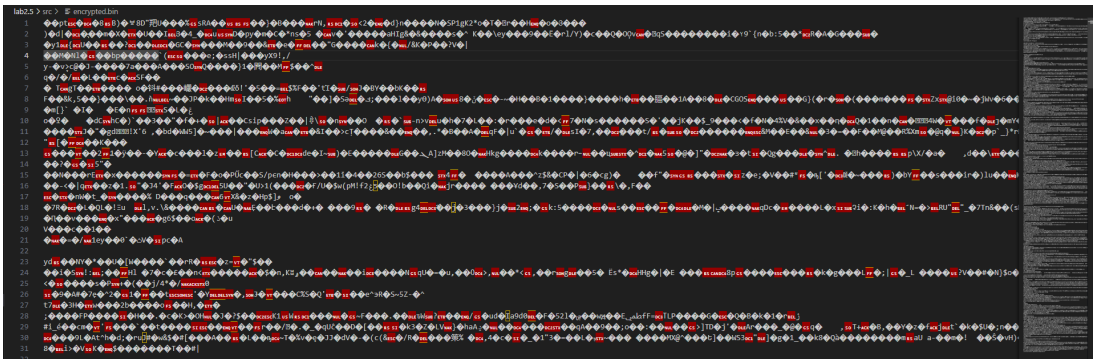

Рисунок 4.2 – Бинарный файл

После, расшифруем бинарный файл:

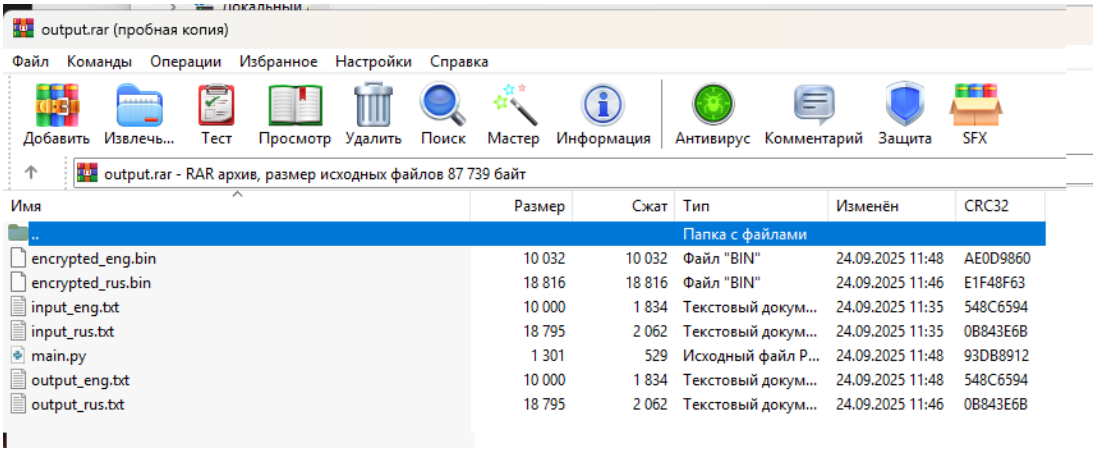

Рисунок 4.3 – Расшифрованный архив