



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 №2.4 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 произвольного файла.

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 произвольного файла с использованием разработанной программы;
3. предусмотреть работу программы с пустым и однобайтовым файлом;
4. провести тестирование программы на различных входных данных.

2 Теоретическая часть

Вопросы для защиты работы

1. Виды симметричного шифрования (поточные и блочные). Приведите схему для одного из видов.

Симметричное шифрование делится на два основных типа:

- **Поточные шифры** — данные шифруются последовательно, побитово или побайтово. Для генерации ключевой последовательности используется генератор псевдослучайной гаммы. Шифртекст получается как $C_i = M_i \oplus K_i$. Пример: RC4.
- **Блочные шифры** — открытый текст делится на блоки фиксированного размера (например, 64 бита в DES или 128 бит в AES). Каждый блок шифруется с использованием одного и того же ключа. Примеры: DES, AES.

Схема блочного шифра:

$$M_1, M_2, \dots, M_n \xrightarrow{DES} C_1, C_2, \dots, C_n$$

2. Опишите алгоритм шифрования AES.

Algorithm 1 Алгоритм шифрования AES

Require: Входной файл F_{in} , секретный ключ K длиной 128, 192 или 256 бит

Ensure: Зашифрованный файл F_{enc}

- 1: Разбить F_{in} на блоки по 128 бит.
 - 2: **for** каждый блок M_i **do**
 - 3: Выполнить начальное сложение блока с ключом (*AddRoundKey*).
 - 4: **for** $i = 1$ **to** $N_r - 1$ **do** $\triangleright N_r$ — число раундов (10, 12 или 14)
 - 5: Выполнить побайтовую подстановку (*SubBytes*).
 - 6: Переставить строки матрицы (*ShiftRows*).
 - 7: Смешать столбцы матрицы (*MixColumns*).
 - 8: Выполнить *AddRoundKey* с подключом K_i .
 - 9: **end for**
 - 10: Выполнить финальный раунд: *SubBytes*, *ShiftRows*, *AddRoundKey*.
 - 11: Записать зашифрованный блок в F_{enc} .
 - 12: **end for**
 - 13: Сохранить результат в F_{enc} .
-

3. Дайте определения алгоритмов перестановки и подстановки. Приведите примеры каждого из этих видов алгоритмов. Приведите пример алгоритма, использующего оба подхода.

- **Алгоритмы перестановки** — методы, при которых изменяется порядок символов или битов открытого текста без их замены. Пример: шифр маршрутной перестановки.
- **Алгоритмы подстановки** — методы, при которых каждый символ заменяется другим по определённом правилу. Пример: шифр Цезаря.
- **Алгоритмы, сочетающие оба подхода** — современные блочные шифры, использующие и перестановки, и подстановки для повышения стойкости. Пример: DES, AES.

3 Практическая часть.

Листинг 3.1 – Файл main.py,

```
1 from Crypto.Cipher import AES
2 from Crypto.Random import get_random_bytes
3 import os
4
5 def pad(data: bytes) -> bytes:
6     padding_len = 16 - (len(data) % 16)
7     return data + bytes([padding_len] * padding_len)
8
9 def unpad(data: bytes) -> bytes:
10    padding_len = data[-1]
11    return data[:-padding_len]
12
13 def encrypt_file(input_file: str, output_file: str, key: bytes):
14    cipher = AES.new(key, AES.MODE_CBC)
15    with open(input_file, "rb") as f:
16        plaintext = f.read()
17
18    padded_data = pad(plaintext)
19    ciphertext = cipher.encrypt(padded_data)
20
21    with open(output_file, "wb") as f:
22        f.write(cipher.iv + ciphertext)
23
24 def decrypt_file(input_file: str, output_file: str, key: bytes):
25    with open(input_file, "rb") as f:
26        iv = f.read(16)
27        ciphertext = f.read()
28
29    cipher = AES.new(key, AES.MODE_CBC, iv)
30    decrypted_data = cipher.decrypt(ciphertext)
31    unpadded_data = unpad(decrypted_data)
32
33    with open(output_file, "wb") as f:
34        f.write(unpadded_data)
35
36
37 if __name__ == "__main__":
38    key = b"thisisasecretkey"
```

```
39  
40     encrypt_file("input_rus.txt", "encrypted_rus.bin", key)  
41     decrypt_file("encrypted_rus.bin", "output_rus.txt", key)  
42  
43     print("Шифрование и расшифровка завершены.")
```


Рисунок 4.4 – Бинарный файл шифрования русского

```
lab24 > src > ≡ output_eng.txt
1 function function example encryption file number information world information result structure data computer number information model check check
```

```
lab24 > src >  output_rus.txt
1  привет обработка структура текст структура обработка программа модель информация система проверка привет информация шифрование текст ключ обработка
```

Рисунок 4.6 – Текстовый файл на русском после шифрования (10000 символов)