МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КУБГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчет**

**по практическому заданию №9**

**по курсу**

**«КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ»**

Работу выполнил

Студент 49 группы

Лобода Д.А.

Преподаватель:

Крамаренко А.А.

Краснодар 2024

**Постановка задачи.**

Реализовать программный продукт, позволяющий шифровать и

расшифровывать сообщения на русском и английском языках с помощью

двух блочных симметричных шифров(TripleDES, AES). Чтение открытого

текста и шифртекста должно быть возможно с клавиатуры и из файла, запись

результата шифрования/расшифрования возможна на экран и в файл. Ключ

формируется автоматически и сохраняется на весь сеанс шифрования. Ключ

сохраняется в отдельный файл. Возможно хранение нескольких ключей от

разных сеансов шифрования. У пользователя есть возможность выбора,

какой криптосистемой пользоваться. Шифрование реализовать для двух

различных кодировок текста. Для реализации криптоалгоритмов

пользоваться встроенными библиотеками используемых языков.

AES (Advanced Encryption Standard)

Основные характеристики и особенности:

1. Структура: AES - симметричный блочный шифр, который оперирует с блоками данных размером 128 бит.

2. Длина зашифрованного сообщения: Длина зашифрованного сообщения равна длине исходного сообщения.

3. Ключи: Использует ключи длиной 128, 192 или 256 бит.

4. Режимы работы: Может работать в различных режимах, таких как ECB (Electronic Codebook), CBC (Cipher Block Chaining), CTR (Counter) и других.

5. Применение: AES широко используется для защиты конфиденциальных данных, включая данные, передаваемые по сети, хранящиеся на диске, а также во многих других сценариях.

6. Современный стандарт: AES является стандартом шифрования, утвержденным Национальным институтом стандартов и технологий (NIST) США, и он заменил устаревший DES.

7. Преобразование: Алгоритм включает в себя серию подстановок и перестановок данных с использованием ключей для обеспечения криптографической стойкости.

Логика алгоритма AES:

1. Генерация подключей: Из основного ключа генерируются раундовые подключи для каждого раунда шифрования.

2. Режимы работы: Выбирается режим работы шифра (ECB, CBC, CTR и т. д.).

3. Деление сообщения на блоки: Исходное сообщение разбивается на блоки размером 128 бит.

4. Шифрование: Каждый блок шифруется с использованием ключей и выбранного режима работы.

5. Дешифрование: Если требуется восстановление исходного сообщения, процесс дешифрования применяется к зашифрованным блокам с использованием тех же ключей и режима работы.

TripleDES (Triple Data Encryption Standard)

Основные характеристики и особенности:

1. Структура: TripleDES - это блочный шифр, который работает с блоками данных размером 64 бита.

2. Длина зашифрованного сообщения: Длина зашифрованного сообщения равна длине исходного сообщения.

3. Ключи: Использует три ключа длиной 56 бит каждый, что обеспечивает более высокий уровень безопасности по сравнению с обычным DES.

4. Режимы работы: Может использоваться в различных режимах работы, таких как ECB (Electronic Codebook), CBC (Cipher Block Chaining) и других.

5. Применение: TripleDES широко используется для защиты конфиденциальных данных в различных приложениях, таких как банковские транзакции, обмен сообщениями и т. д.

6. Устаревший алгоритм: Несмотря на свою широкую распространенность, TripleDES считается устаревшим в силу увеличивающихся требований к безопасности и более современных альтернатив, таких как AES.

7. Преобразование: Алгоритм включает в себя серию перестановок, подстановок и других преобразований данных с использованием ключей для обеспечения криптографической стойкости.

Логика алгоритма TripleDES:

1. Генерация подключей: Из трех 56-битовых ключей генерируются 16 подключей, которые используются в процессе шифрования.

2. Режимы работы: Выбирается режим работы шифра (ECB, CBC и т. д.).

3. Деление сообщения на блоки: Исходное сообщение разбивается на блоки размером 64 бита.

4. Шифрование: Каждый блок шифруется с использованием ключей и выбранного режима работы.

5. Дешифрование: Если требуется восстановление исходного сообщения, процесс дешифрования применяется к зашифрованным блокам с использованием тех же ключей и режима работы.

**Текст программы:**

**Файл Lab\_9.py:**

from Crypto.Cipher import DES3, AES  
from Crypto.Random import get\_random\_bytes  
from Crypto.Util.Padding import pad, unpad  
import base64  
import os  
  
  
def generate\_key():  
 return get\_random\_bytes(16)  
  
  
def save\_key\_to\_file(key, filename):  
 with open(filename, 'wb') as file:  
 file.write(key)  
  
  
def load\_key\_from\_file(filename):  
 with open(filename, 'rb') as file:  
 return file.read()  
  
  
def encrypt\_message(message, key, algorithm, encoding):  
 if algorithm == 'TripleDES':  
 cipher = DES3.new(key, DES3.MODE\_ECB)  
 elif algorithm == 'AES':  
 cipher = AES.new(key, AES.MODE\_ECB)  
 else:  
 raise ValueError("Unsupported algorithm")  
  
 message\_bytes = message.encode(encoding)  
 padded\_message = pad(message\_bytes, max(len(key), 16))  
 encrypted\_message = cipher.encrypt(padded\_message)  
 return base64.b64encode(encrypted\_message).decode('utf-8')  
  
  
def decrypt\_message(encrypted\_message, key, algorithm, encoding):  
 if algorithm == 'TripleDES':  
 cipher = DES3.new(key, DES3.MODE\_ECB)  
 elif algorithm == 'AES':  
 cipher = AES.new(key, AES.MODE\_ECB)  
 else:  
 raise ValueError("Unsupported algorithm")  
  
 encrypted\_message = base64.b64decode(encrypted\_message)  
 decrypted\_message = cipher.decrypt(encrypted\_message)  
 decrypted\_message = unpad(decrypted\_message, max(len(key), 16))  
 return decrypted\_message.decode(encoding)  
  
  
def read\_from\_file(filename):  
 with open(filename, 'r') as file:  
 return file.read()  
  
  
def write\_to\_file(filename, content):  
 with open(filename, 'w') as file:  
 file.write(content)  
  
  
def main():  
 algorithm = input("Выберите алгоритм шифрования (TripleDES - T, AES - A): ").strip().upper()  
 if algorithm == 'T':  
 algorithm = 'TripleDES'  
 elif algorithm == 'A':  
 algorithm = 'AES'  
 else:  
 print("Неподдерживаемый алгоритм.")  
 return  
  
 key\_filename = input("Введите имя файла для сохранения/загрузки ключа: ").strip()  
 if os.path.exists(key\_filename):  
 key = load\_key\_from\_file(key\_filename)  
 else:  
 key = generate\_key()  
 save\_key\_to\_file(key, key\_filename)  
  
 encoding = input("Выберите кодировку текста (UTF-8 - U, Windows-1251 - W): ").strip().upper()  
 if encoding == 'U':  
 encoding = 'UTF-8'  
 elif encoding == 'W':  
 encoding = 'Windows-1251'  
 else:  
 print("Неподдерживаемая кодировка.")  
 return  
  
 mode = input("Выберите режим работы (шифрование - E, расшифрование - D): ").strip().upper()  
 if mode == 'E':  
 mode = 'Encryption'  
 elif mode == 'D':  
 mode = 'Decryption'  
 else:  
 print("Неподдерживаемый режим работы.")  
 return  
  
 input\_type = input("Выберите источник ввода (клавиатура - K, файл - F): ").strip().upper()  
 if input\_type == 'K':  
 input\_type = 'Keyboard'  
 elif input\_type == 'F':  
 input\_type = 'File'  
 else:  
 print("Неподдерживаемый источник ввода.")  
 return  
  
 if mode == 'Encryption':  
 message = ""  
 if input\_type == 'Keyboard':  
 print("Введите сообщение для шифрования (для завершения ввода введите 'end'):")  
 while True:  
 letter = input("Введите букву: ")  
 if letter.lower() == 'end':  
 break  
 message += letter  
 else:  
 message = read\_from\_file(input("Введите путь к файлу с открытым текстом: "))  
 encrypted\_message = encrypt\_message(message, key, algorithm, encoding)  
 output\_type = input("Выберите куда записать результат (экран - S, файл - F): ").strip().upper()  
 if output\_type == 'S':  
 print("Зашифрованное сообщение:", encrypted\_message)  
 elif output\_type == 'F':  
 write\_to\_file(input("Введите путь к файлу для записи шифртекста: "), encrypted\_message)  
 elif mode == 'Decryption':  
 encrypted\_message = input("Введите шифртекст: ") if input\_type == 'Keyboard' else read\_from\_file(  
 input("Введите путь к файлу с шифртекстом: "))  
 decrypted\_message = decrypt\_message(encrypted\_message, key, algorithm, encoding)  
 output\_type = input("Выберите куда записать результат (экран - S, файл - F): ").strip().upper()  
 if output\_type == 'S':  
 print("Расшифрованное сообщение:", decrypted\_message)  
 elif output\_type == 'F':  
 write\_to\_file(input("Введите путь к файлу для записи открытого текста: "), decrypted\_message)  
 else:  
 print("Неподдерживаемый режим работы")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()