МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра 311 «Прикладные программные средства и математические методы»

**Лабораторная работа №6**

по дисциплине:

«Информационная безопасность»

на тему:

**«**DES»

Работу выполнил:

студент гр. М3О-318Бк-19

Артиков Темур Улугбекович

Научный руководитель:

ассистент, Кос О.И.

Дата выдачи задания: 18.09.2021 .

Дата сдачи: 27.11.2021 .

.

Москва 2021 г.

Оглавление:

Глава 1: Введение…………………………………………………………………3

Глава 2: Алгоритм для реализации………………………………………………3

Глава 3: Код программы………………………………………………………….4

Глава 4: Демонстрация работы………………………………………………......7

Выводы………...…………………………………………………………………12

Список литературы………………………………………………………………13

**Глава 1: Введение**

**Цель работы:** программирование шифра DES (шифрование, дешифрование)

Данная лабораторная работа была выполнена с помощью языка программирования Python в среде программирования

**Глава 2: Алгоритм для реализации**

**Стандарт шифрования данных (DES)**— *блочный шифр* с *симметричными ключами*, разработан Национальным Институтом Стандартов и Технологии (*NIST* – National Institute of Standards *and* *Technology*).

**История**

В 1973 году *NIST* издал запрос для разработки предложения национальной *криптографической системы* с *симметричными ключами*.

Предложенная IBM модификация проекта, названная Lucifer, была принята как *DES*. *DES* были изданы в эскизном виде в *Федеральном Регистре* в марте 1975 года как **Федеральный Стандарт Обработки Информации (FIPS – Federal Information Processing Standard)**.

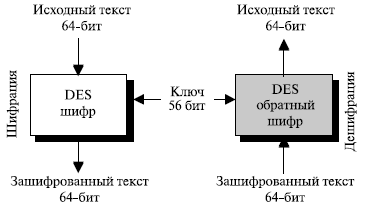
После публикации эскиз строго критиковался по двум причинам. Первая: критиковалась сомнительно маленькая *длина ключа* (только 56 битов), что могло сделать шифр уязвимым к атаке "грубой силой". Вторая причина: критики были обеспокоены некоторым скрытым построением внутренней структуры *DES*.

Они подозревали, что некоторая часть структуры (S -блоки) может иметь скрытую лазейку, которая позволит расшифровывать сообщения без ключа. Впоследствии проектировщики IBM сообщили, что внутренняя структура была доработана, чтобы предотвратить *криптоанализ*.

*DES* был наконец издан как *FIPS* 46 в Федеральном Регистре в январе 1977 года. Однако *FIPS* объявил *DES* как стандарт для использования в неофициальных приложениях. *DES* был наиболее широко используемым блочным шифром с *симметричными ключами*, начиная с его публикации. Позже *NIST* предложил новый стандарт (*FIPS* 46-3), который рекомендует использование *тройного DES* (трехкратно повторенный шифр *DES*) для будущих приложений. Как мы увидим далее, в лекциях 9-10, предполагается, что более новый стандарт *AES* заменит *DES*.

**Общие положения**

Как показано на рис. 1, *DES* — *блочный шифр*.



**Рис. 1.**Шифрование и дешифрование в DES

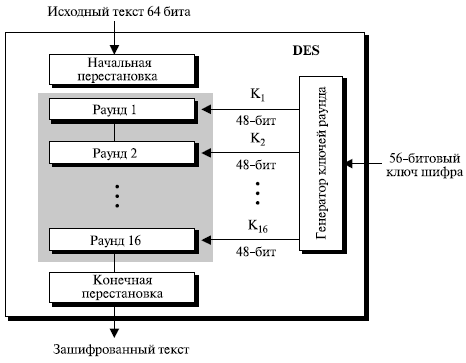
На стороне шифрования *DES* принимает 64 -битовый исходный текст и порождает 64 -битовый зашифрованный текст; на стороне дешифрования *DES* принимает 64 -битовый зашифрованный текст и порождает 64 -битовый исходный текст. На обеих сторонах для шифрования и дешифрования применяется один и тот же 56 -битовый ключ.

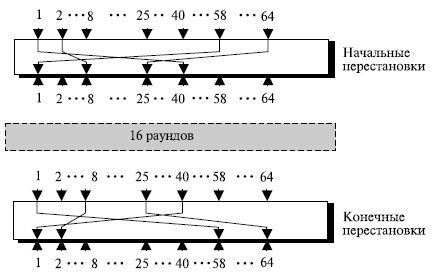
**Структура DES**

Рассмотрим сначала *шифрование*, а потом *дешифрование*. Процесс шифрования состоит из двух перестановок (P -блоки) — они называются начальные и *конечные* перестановки, — и шестнадцати раундов Файстеля. Каждый раунд использует различные сгенерированные 48 -битовые ключи. *Алгоритм* генерации будет рассмотрен позднее. Рисунок 2 показывает элементы шифра *DES* на стороне шифрования.

**Начальные и конечные перестановки**

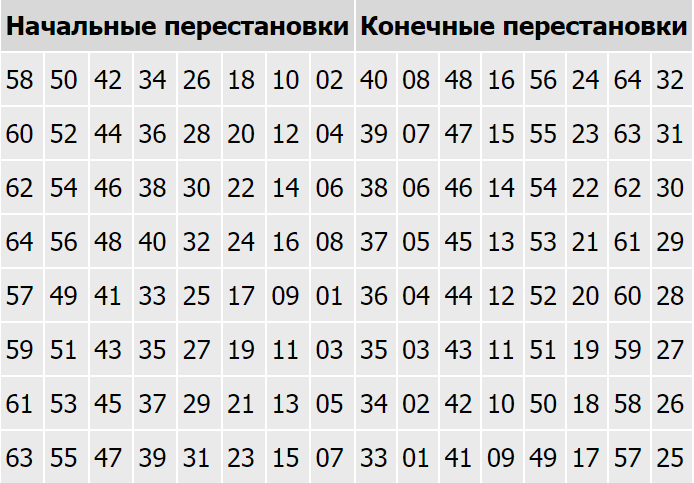
Рисунок 3 показывает начальные и конечные перестановки (P -блоки). Каждая из перестановок принимает 64 -битовый вход и переставляет его элементы по заданному правилу. Здесь показано только небольшое число входных портов и соответствующих выходных портов. Эти перестановки — прямые перестановки без ключей, которые инверсны друг другу. Например, в начальной перестановке 58 -й бит на входе переходит в первый бит на выходе. Аналогично, в конечной перестановке первый входной бит переходит в 58 -й бит на выходе. Другими словами, если между этими двумя перестановками не существует раунда, 58 -й бит, поступивший на вход устройства начальной перестановки, будет доставлен на 58 -й выход финальной перестановкой.

  
**Рис. 2.**Общая структура DES

  
**Рис. 3.**Начальные и конечные шаги перестановки DES

Правила перестановки для этого P -блока показаны в таблице 8.1. Таблицу можно представить как 64 -элементный массив. Заметим, что работу с таблицей обсуждали, значение каждого элемента определяет номер входного порта, а порядковый номер (индекс) элемента определяет номер выходного порта.

Таблица 8.1. Таблица начальных и конечных перестановок



Эти две перестановки не имеют никакого значения для криптографии в *DES*. Обе перестановки – без ключей и предопределенны. Причина, почему они включены в *DES*, не ясна и не была указана проектировщиками *DES*. Можно предположить, что *DES* был проектом, который предполагалось реализовать в аппаратных средствах (на чипах), и что эти две сложные перестановки должны были затруднить программное моделирование механизма шифрования.

**Функция Шеннона**

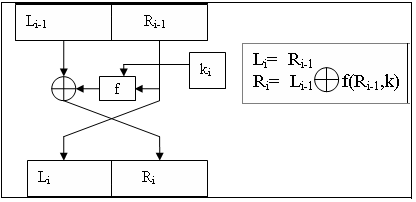
В области сжатия данных, **код Шеннона**, названный в честь его создателя, Клода Шеннона, — это алгоритм сжатия данных без потерь с помощью построения префиксных кодов на основе набора символов и их вероятностей (расчётное или измеренное). Он является субоптимальным в том смысле, что не позволяет достичь минимально возможных кодовых длин как в кодировании Хаффмана, и никогда не будет лучше, но иногда равным с кодом Шеннона-Фано.

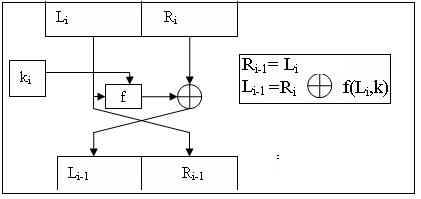
Этот метод был первым в своём роде, эта методика была использована для доказательства теоремы Шеннона о помехоустойчивом кодировании в 1948 в его статье «Математическая Теория связи»[[1]](https://www.wikiwand.com/ru/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A8%D0%B5%D0%BD%D0%BD%D0%BE%D0%BD%D0%B0#citenote1).

В кодировании Шеннона символы располагаются в порядке от наиболее вероятных к наименее вероятным. Им присваиваются коды, путём взятия первых     цифр из двоичного разложения кумулятивной



вероятности  . Здесь  обозначает функцию потолок, которая округляет  до ближайшего целого значения, большего либо равного .

**Преобразование Фейстеля**  
Это преобразование над векторами (блоками), представляющими собой левую и правую половины регистра сдвига. В алгоритме DES используются прямое преобразование сетью Фейстеля в шифровании (см. Рис.4) и обратное преобразование сетью Фейстеля в расшифровании (см. Рис.5).  
   
 Рис. 4. Прямое преобразование сетью Фейстеля.

  
 Рис. 5. Обратное преобразование сетью Фейстеля

**Глава 3: Код программы**

Исходный код программы:

from secrets import token\_bytes

from Cryptodome.Cipher import DES

key = token\_bytes(8)

def encrypt(msg):

cipher = DES.new(key, DES.MODE\_EAX)

nonce = cipher.nonce

ciphertext, tag = cipher.encrypt\_and\_digest(msg.encode('ascii'))

return nonce, ciphertext, tag

def decrypt(nonce, ciphertext, tag):

cipher = DES.new(key, DES.MODE\_EAX, nonce=nonce)

plaintext = cipher.decrypt(ciphertext)

try:

cipher.verify(tag)

return plaintext.decode('ascii')

except:

return False

nonce, ciphertext, tag = encrypt(input('Enter a message: '))

plaintext = decrypt(nonce, ciphertext, tag)

print(f'Cipher text: {ciphertext}')

if not plaintext:

print('Message is corrupted!')

else:

print(f'Plain text: {plaintext}')

**Глава 4: Демонстрация работы**

1. Введите ваш текст. В моё случае «Hello World!».



Рисунок 6. – Демонстрация работы программы.

**Результаты тестирования программы:**

1. Ввод: «Hello World!». Вывод: b'7\xfc&\xc0\xe0\xbc"o\xdf[\xb2.'
2. Ввод: « 7\xfc&\xc0\xe0\xbc"o\xdf[\xb2.». Вывод: «Hello World!».

**Выводы**

В данной работе я изучил метод шифрования DES. DES был национальным стандартом США в 1977—1980 гг., но в настоящее время DES используется (с ключом длины 56 бит) только для устаревших систем, чаще всего используют его более криптоустойчивый вид (3DES, DESX). 3DES является простой эффективной заменой DES, и сейчас он рассмотрен как стандарт. В ближайшее время DES и Triple DES будут заменены алгоритмом AES (Advanced Encryption Standard — Расширенный Стандарт Шифрования). Алгоритм DES широко применяется для защиты финансовой информации: так, модуль THALES (Racal) HSM RG7000 полностью поддерживает операции TripleDES для эмиссии и обработки кредитных карт VISA, EuroPay и проч. Канальные шифраторы THALES (Racal) DataDryptor 2000 используют TripleDES для прозрачного шифрования потоков информации. Также алгоритм DES используется во многих других устройствах и решениях THALES-eSECURITY.

Список литературы:

1. Библиотека pycryptodome. <https://pycryptodome.readthedocs.io/en/latest/src/introduction.html>
2. DES cipher Wikipedia. <https://ru.wikipedia.org/wiki/DES>