Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе № 2

“Блочное симметричное шифрование”

по дисциплине Информационная безопасность

Вариант 10

|  |  |
| --- | --- |
| Студент группы № P34151 | Шипулин Павел Андреевич |
| Преподаватель | Маркина Татьяна Анатольевна |

Санкт-Петербург

2024

# Цель работы

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

# Вариант задания

Реализовать систему симметричного блочного шифрования, позволяющую шифровать и дешифровать файл на диске с использованием заданного блочного шифра в заданном режиме шифрования. Перечень блочных шифров и режимов шифрования приведен в таблице. Номер шифра и режима для реализации получить у преподавателя.

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритм | | Режим шифрования | |
| Номер | Название | Номер | Название |
| 4 | ГОСТ 28147-89 | д | OFB |

# Ход работы

1. Ознакомиться с теоретическими основами шифрования данных.
2. Получить вариант задания у преподавателя.
3. Написать программу согласно варианту задания.
4. Отладить разработанную программу и показать результаты работы программы преподавателю.
5. Составить отчет по лабораторной работе.

# Листинг программ

Ссылка на репозиторий:

<https://github.com/PashcalE2/IS/tree/main/cryptography/first_part>

## Файл lab2.py

import random  
  
  
\_switch\_table = [  
 [9, 6, 3, 2, 8, 11, 1, 7, 10, 4, 14, 15, 12, 0, 13, 5],  
 [3, 7, 14, 9, 8, 10, 15, 0, 5, 2, 6, 12, 11, 4, 13, 1],  
 [14, 4, 6, 2, 11, 3, 13, 8, 12, 15, 5, 10, 0, 7, 1, 9],  
 [14, 7, 10, 12, 13, 1, 3, 9, 0, 2, 11, 4, 15, 8, 5, 6],  
 [11, 5, 1, 9, 8, 13, 15, 0, 14, 4, 2, 3, 12, 7, 10, 6],  
 [3, 10, 13, 12, 1, 2, 0, 11, 7, 5, 9, 4, 8, 15, 14, 6],  
 [1, 13, 2, 9, 7, 10, 6, 0, 8, 12, 4, 5, 15, 3, 11, 14],  
 [11, 10, 15, 5, 0, 12, 14, 8, 6, 2, 3, 9, 1, 7, 13, 4]  
]  
  
  
def gost\_main\_block(N: int, K: int) -> int:  
 A = N & 0xFFFFFFFF  
 B = (N >> 32) & 0xFFFFFFFF  
  
 new\_B = A  
 new\_A = A ^ K  
  
 S = [(new\_A >> ((7 - i) \* 4)) & 15 for i in range(8)]  
 for i in range(8):  
 S[i] = \_switch\_table[i][S[i]]  
  
 new\_A = 0  
 for i in range(8):  
 new\_A = new\_A | (S[i] << ((7 - i) \* 4))  
  
 new\_A = ((new\_A << 11) & 0xFFFFFFFF) + (new\_A >> 21)  
 new\_A = B ^ new\_A  
  
 return (new\_B << 32) + new\_A  
  
  
def gost\_ofb\_encrypt(file\_data: bytes, K: list[int], IV: int) -> bytes:  
 iters\_count = len(file\_data) // 8  
  
 T = IV  
 result = b''  
 for i in range(iters\_count):  
 T = gost\_main\_block(T, K[i % len(K)])  
  
 text\_block = int.from\_bytes(file\_data[i \* 8:(i + 1) \* 8])  
 T = T ^ text\_block  
  
 result += int.to\_bytes(T, 8)  
  
 return result  
  
  
def gost\_ofb\_decrypt(file\_data: bytes, K: list[int], IV: int) -> bytes:  
 iters\_count = len(file\_data) // 8  
  
 T = IV  
 result = b''  
 for i in range(iters\_count):  
 T = gost\_main\_block(T, K[i % len(K)])  
  
 text\_block = int.from\_bytes(file\_data[i \* 8:(i + 1) \* 8])  
 result += int.to\_bytes(T ^ text\_block, 8)  
  
 T = text\_block  
  
 return result  
  
  
def lab4(input\_file: str, encrypted\_file: str, decrypted\_file: str, K: list[int], IV: int):  
 with open(input\_file, "r", encoding="UTF-8") as file:  
 input\_data = file.read()  
  
 input\_bytes = bytes(input\_data, "UTF-8")  
 if len(input\_bytes) % 8 > 0:  
 input\_bytes = input\_bytes + (b' ' \* (8 - (len(input\_bytes) % 8)))  
  
 encrypted = gost\_ofb\_encrypt(input\_bytes, K, IV)  
 with open(encrypted\_file, "wb") as file:  
 file.write(encrypted)  
  
 decrypted = gost\_ofb\_decrypt(encrypted, K, IV)  
 with open(decrypted\_file, "wb") as file:  
 file.write(decrypted)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 """  
 Реализовать систему симметричного блочного шифрования,  
 позволяющую шифровать и дешифровать файл на диске с использованием  
 заданного блочного шифра в заданном режиме шифрования. Перечень  
 блочных шифров и режимов шифрования приведен в таблице. Номер  
 шифра и режима для реализации получить у преподавателя.  
   
 Вариант 10  
 Алгоритм шифрования: ГОСТ 28147-89  
 Режим шифрования: OFB  
 """  
  
 \_K = [random.randint(0, (1 << 32) - 1) for i in range(8)]  
 \_IV = random.randint(0, (1 << 64) - 1)  
  
 lab4("input.txt", "encrypted.txt", "decrypted.txt", \_K, \_IV)

# Выполнение

## Результат выполнения программы

Файл input.txt:



Файл encrypted.txt:



Файл decrypted.txt:



# Вывод

Изучил основные принципы современных алгоритмов блочного симметричного шифрования, ознакомился с алгоритмом ГОСТ 28147-89 и сделал его программную реализацию. Этот алгоритм до сих пор является актуальным, но криптостойкость его шифров зависит от таблицы замен, поэтому был специфицирован шифр «Магма» ГОСТ Р 34.12-2015 с тем же алгоритмом в режиме простой замены, но с фиксированной таблицей замен.