МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Информационная безопасность систем и технологий»

Отчет
по лабораторной работе №3
на тему «Программная реализация операций подстановки и перестановки»

Дисциплина: МиСКЗИ
Группа: 21ПИ1
Выполнил: Гусев Д. А.
Количество баллов:
Дата сдачи:
Принял: Липилин О. В.

2024

1 Цель работы: получение навыков по программной реализации основных криптографических преобразований.

2 Задание на лабораторную работу.

2.1 Реализовать операцию подстановки (замены) для 4-х битного вектора. Таблица замены указана в таблице 1. Для проверки выполнить операцию подстановки над 64-х битным вектором a={0000000100100011010001010110011111000100110101011110011011110111111}, (двоичный вектор a разбивается на 16 частей, над каждой частью выполняется замена).

Таблица 1 — Вариант 8

№ вар	Таблица замены S	Сдвиг p
8	1, 7, E, D, 0, 5, 8, 3, 4, F, A, 6, 9, C, B, 2	4

2.2 Реализовать подстановку двух смежных 4х битных векторов с использованием эквивалентной таблицы замены S* (таблицу S* сформировать заранее). Выполнить подстановку над вектором a (двоичный вектор a разбивается уже на 8 частей, над каждой частью выполняется замена). Сравнить результаты выполнения п.1 и п. 2. Указать размер таблицы S*.

2.3 Реализовать операцию перестановки над 8-ми битным вектором с использованием циклического сдвига вправо на p бит (p указано в варианте). Выполнить перестановку над вектором a (двоичный вектор a разбивается на 8 частей, над каждой частью выполняется перестановка).

2.4 Реализовать комбинацию операций подстановки и перестановки, (входные данные считываются блоком x по 8 бит, блок разбивается на 2 двоичных вектора x1 и x2 по 4 бита, над каждым вектором выполняется подстановка S. Результаты подстановки объединяются в блок a размером 8 бит, который циклически сдвигается вправо на p бит. Результатом преобразования блока x является блок b размером 8 бит).

2.5 Выполнить преобразование файла произвольного формата (размером не менее 1 Кб) с использованием преобразования из п. 4.

2.6 Реализовать комбинацию операций подстановки и перестановки, указанную на рисунке 1 с использованием эквивалентной подстановки S’(таблицу замены S’ сформировать заранее на основе подстановки S* и сдвига).

2.7 Выполнить преобразование файла из п.6 с использованием эквивалентной подстановки. Сравнить результаты. Указать размер таблицы S’

3 Выполнение лабораторную работы:

3.1 Была реализована операция подстановки (замены) для 4-х битного вектора. Результат представлен на рисунке 1. Код программы представлен ниже. Таблица с битами представлена в таблице 1. Код программы представлен в репозитории на github в файле *LB3_1_Replace.cpp*.

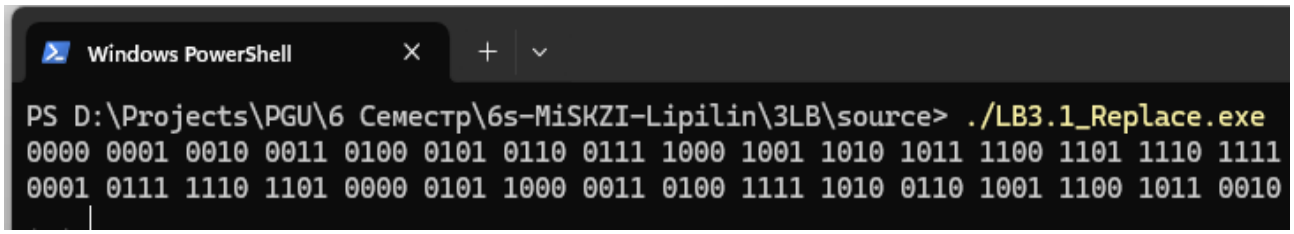

Рисунок 1 - Работа программы

Таблица 1 — Вектор после операции замены

Vector a	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Replaced	0x01	0x07	0x0E	0x0D	0x00	0x05	0x08	0x03	0x04	0x0F	0x0A	0x06	0x09	0x0C	0x0B	0x02
	0001	0111	1110	1101	0000	0101	1000	0011	0100	1111	1010	0110	1001	1100	1011	0010

Как видно из таблицы, получившийся вектор полностью соответсвует таблице замены, значит программа работает корректно.

3.2 Была реализована подстановка двух смежных 4х битных векторов с использованием эквивалентной таблицы замены S*. Вектор a был преобразован в вектор {0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF}. Была сформирована таблица замены S*. Таблица замены представлена в таблице 2. Работа

программы представлена на рисунке 2. Код программы представлен в репозитории на github в файле *LB3_2_2xReplace.cpp*.

Таблица 2 — Таблица замены S*

S*	0x0F	0x07	0x01	0x0D	0x0A	0x0E	0x08	0x03	0x04	0x09	0x0B	0x06	0x05	0x0C	0x02	0x00
	1111	0111	0001	1101	1010	1110	1000	0011	0100	1001	1011	0110	0101	1100	0010	0000

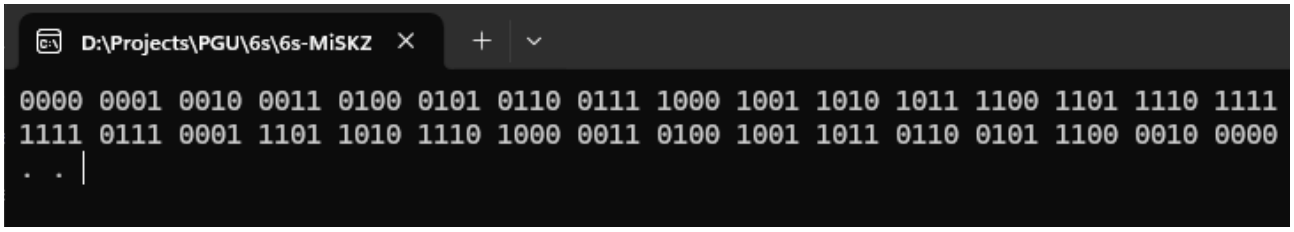

Рисунок 2 — Подстановка по таблице S* со смежными векторами

Как видно из таблицы, получившийся вектор полностью соответсвует таблице замены, значит программа работает корректно.

3.3 Была реализована операция перестановки над 8-ми битным вектором с использованием циклического сдвига вправо на p бит. Работа программы представлена на рисунке 3. Результат сдвига представлен в табилце 3. Код программы представлен в репозитории на github в файле *LB3_3_3_shift.cpp*.

Таблица 3 — Результат сдвигов

Vector a	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Shift	0x01	0x00	0x03	0x02	0x05	0x04	0x07	0x06	0x09	0x08	0x0B	0x0A	0x0D	0x0C	0x0F	0x0E
	0001	0000	0011	0010	0101	0100	0111	0110	1001	1000	1011	1010	1101	1100	1111	1110

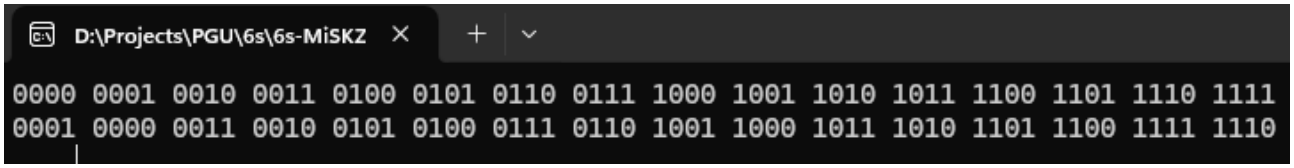

Рисунок 3 — Перестановка по таблице S*

3.4 Была реализована комбинацию операций подстановки и перестановки по таблице замены S*. Работа программы представлена на рисунке 4. Результат зашифрования в табилце 4. Код программы представлен в репозитории на github в файле *LB3_3_4_shift-replace.cpp*.

Таблица 4 — Результат зашифрования

Vector a	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Shift + Replace	0x07	0x01	0x0D	0x0E	0x05	0x00	0x03	0x08	0x0F	0x04	0x06	0x0A	0x0C	0x09	0x02	0x0B
	0111	0001	1101	1110	0101	0000	0011	1000	1111	0100	0110	1010	1100	1001	0010	1011

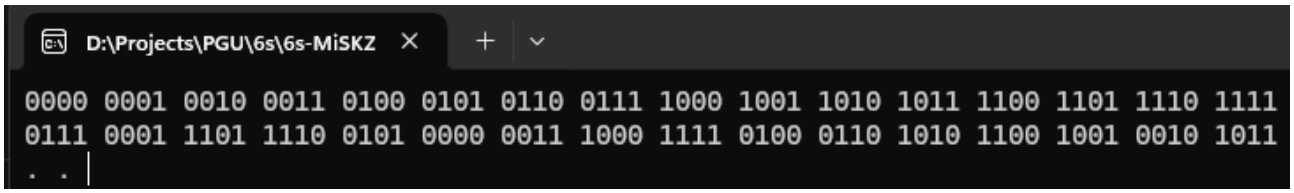

Рисунок 4 — Перестановка и подстановка по таблице S*

3.5 Было выполнено преобразование файла (размером 2,47 Кб) с использованием преобразования из п. 4. Результат представлен на рсиунке 5. Код программы представлен в репозитории на github в файле *LB3_3_5_transformation.cpp*.

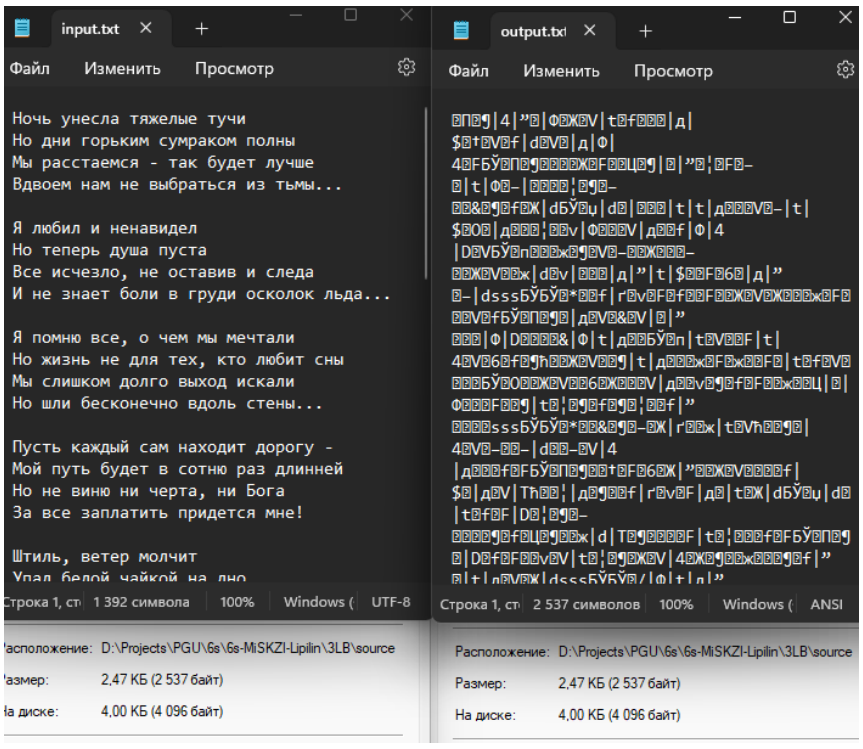

Рисунок 5 - Преобразование файла

Так как размер файла сохраняется после преобразований, можно сделать вывод, что программа работает корректно.

3.6 Была реализована комбинация операций подстановки и перестановки, с использованием эквивалентной подстановки S’(таблица замены S’ была сформирована заранее на основе подстановки S* и сдвига из *таблицы 4*). Код программы аналогичен *LB3_3_5_transformation.cpp*.

3.7 Было выполнено преобразование файла (размером 2,47 Кб) с использованием преобразования из п. 6. Результат представлен на рисунке 6. Код программы представлен в репозитории на github в файле *LB3_3_5_transformation.cpp*.

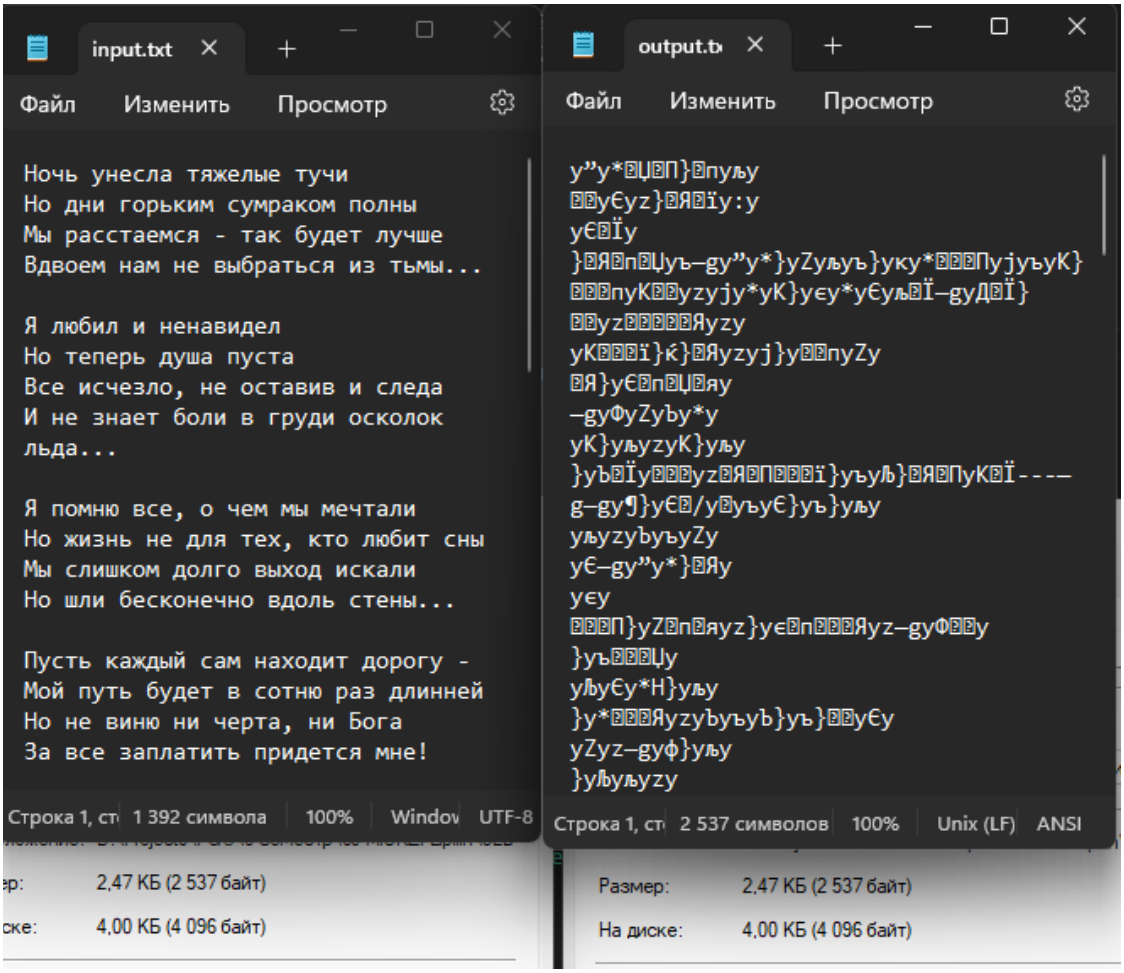

Рисунок 6 - Преобразование файла

4 Вывод: были получены навыки по программной реализации основных криптографических преобразований.