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 операцию подстановки (замены) для 4-х битного вектора. Таблица замены S указана в таблице 1. Для проверки была выполнена операция подстановки над 64-х битным вектором A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}. Результат преобразований представлен в таблице 2. Код программы представлен в репозитории на github в файле *LB3_1_Replace-4bit.cpp*. Результат работы программы представлен на рисунке 1.

Таблица 2 — Результат преобразования 4-х битного вектора (подстановка)

S	0x01	0x07	0x0E	0x0D	0x00	0x05	0x08	0x03	0x04	0x0F	0x0A	0x06	0x09	0x0C	0x0B	0x02
A	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
Replace-4bit	0x01	0x07	0x0E	0x0D	0x00	0x05	0x08	0x03	0x04	0x0F	0x0A	0x06	0x09	0x0C	0x0B	0x02

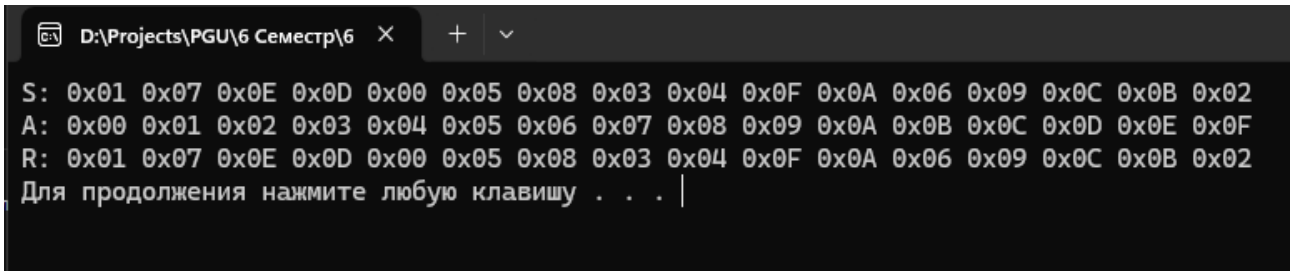

Рисунок 1 — Работа программы LB3_1_Replace-4bit

3.2 Была реализована подстановка двух смежных 4х битных векторов с использованием эквивалентной таблицы замены S* (таблицу S* представлена в таблице 3). Была выполнена подстановку над вектором A={1, 23, 45, 67, 89, AB, CD, EF}. Результат преобразований представлен в таблице 4. Код программы

представлен в репозитории на github в файле *LB3_2_Replace-8bit.cpp*. Результат работы программы представлен на рисунке 2.

Таблица 3 — Таблица замены S*

0x23	0xBF	0xB8	0x9E	0x84	0xDB	0x4D	0xD2	0xE7	0x48	0x66	0x68	0xED	0x87	0x37	0xC5
0x04	0x5A	0xC4	0x43	0x0A	0x33	0x0C	0x64	0xDE	0xD6	0x9F	0xA1	0xCD	0xE0	0x57	0x56
0xD8	0x4B	0x5E	0xE9	0x63	0xD1	0xDF	0xFD	0xB3	0xCA	0xD7	0x7D	0x13	0x6B	0x67	0x7C
0x55	0x82	0xC2	0x12	0x09	0x60	0xFB	0xE1	0xFA	0xDA	0x93	0x2B	0xC7	0x1E	0xE8	0x50
0xC1	0x42	0x2E	0xF0	0xB5	0xEE	0x40	0xE5	0xBE	0x08	0x20	0xF7	0xCC	0xBA	0xE3	0x5C
0x81	0xF2	0xAD	0xEB	0x76	0x15	0x8F	0x38	0xBD	0x24	0x83	0x99	0xEA	0x45	0x62	0x10
0xF4	0xFC	0x9C	0xAE	0x0D	0x58	0xF3	0x0B	0x69	0x54	0x03	0xC9	0xCE	0x6F	0x0F	0x73
0xA7	0x19	0x77	0x8C	0xB1	0x2D	0x61	0xD0	0x11	0x07	0x9B	0x65	0x4F	0x52	0x80	0x92
0xF6	0xA9	0x25	0x5D	0xB0	0xAB	0xC8	0x88	0x7A	0x00	0x2A	0x95	0xA5	0x36	0xC3	0x35
0x85	0x02	0xAC	0x28	0x6C	0x79	0x74	0xEF	0xB7	0x47	0x8B	0x2C	0xB9	0x8E	0xF9	0x5B
0x59	0x05	0x41	0x78	0x75	0x9A	0x0E	0x72	0xAF	0x2F	0x91	0x31	0x3A	0x7E	0x4C	0xA0
0x26	0x3B	0xA4	0x6D	0x30	0x71	0xDC	0x86	0x7F	0x3F	0x5F	0x1C	0x8D	0x53	0x4A	0x3D
0x27	0x14	0x3C	0xF1	0xC0	0x06	0x51	0x01	0x6A	0xD3	0x70	0x22	0x94	0xA6	0x6E	0xBB
0xAA	0xE6	0x44	0x32	0xF8	0xE4	0x21	0xD9	0xCF	0x29	0x96	0xFE	0xD4	0x89	0xBC	0x1A
0x8A	0x16	0xA2	0xC6	0x90	0x7B	0x97	0xF5	0xEC	0x17	0xFF	0x49	0x4E	0xB2	0x18	0x1F
0x34	0x1B	0xB6	0xA3	0xDD	0x39	0xD5	0xB4	0x9D	0x98	0x46	0xCB	0x1D	0xE2	0x3E	0xA8

Таблица 4 - Результат преобразования 8-ми битного вектора (подстановка)

A	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF
Replace-8bit	0xBF	0xE9	0xEE	0x0B	0x00	0x31	0xA6	0x1F

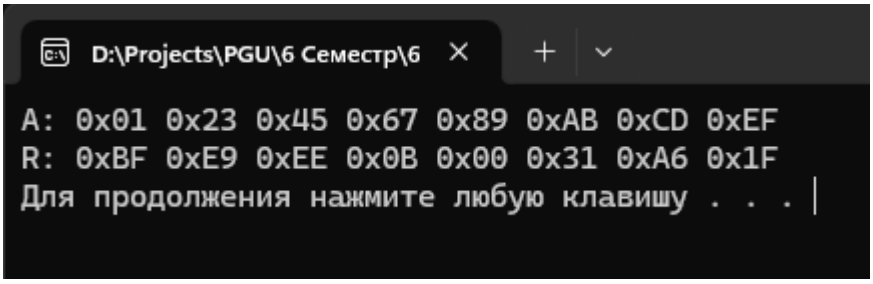

Рисунок 2 — Работа программы LB3_2_Replace-8bit

3.3 Была реализована операция перестановки над 8-ми битным вектором с использованием циклического сдвига вправо на 4 бит (p = 4, согласно варианту 8). Была выполнена перестановка над вектором A={1, 23, 45, 67, 89, AB, CD, EF}. Результат преобразований представлен в таблице 5. Код программы представлен в репозитории на github в файле *LB3_3_Shift-8bit.cpp*. Результат работы программы представлен на рисунке 3.

Таблица 5 - Результат преобразования 8-ми битного вектора (перестановка)

S	0x01	0x07	0x0E	0x0D	0x00	0x05	0x08	0x03
	0x04	0x0F	0x0A	0x06	0x09	0x0C	0x0B	0x02
A	0x01	0x23	0x45	0x67	0x89	0xab	0xcd	0xef
Shift-Replace-2x4bit	0x71	0xde	0x50	0x38	0xf4	0x6a	0xc9	0x2b

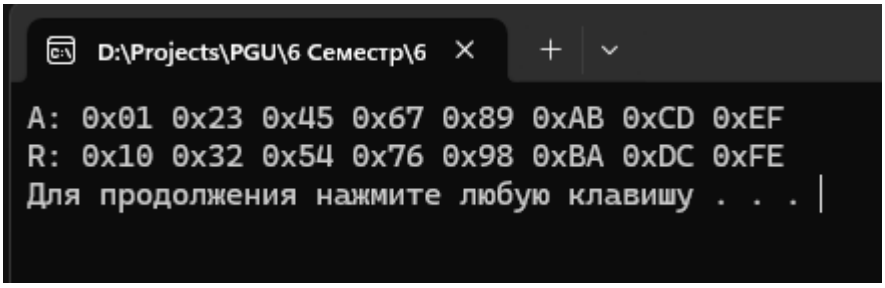

Рисунок 3 — Работа программы LB3_3_Shift-8bit

3.4 Была реализована комбинация операций подстановки и перестановки, (выполняется подстановка S). Было выполнено преобразование над вектором A={1, 23, 45, 67, 89, AB, CD, EF}. Результат преобразований представлен в таблице 6. Код программы представлен в репозитории на github в файле *LB3_4_Shift-Replace-2x4bit.cpp*. Результат работы программы представлен на рисунке 4.

Таблица 6 — Результат преобразования 8-ми битного вектора (подстановка + перестановка)

A	0x01	0x23	0x45	0x67	0x89	0xab	0xcd	0xef
Shift-Replace-2x4bit	0x71	0xde	0x50	0x38	0xf4	0x6a	0xc9	0x2b

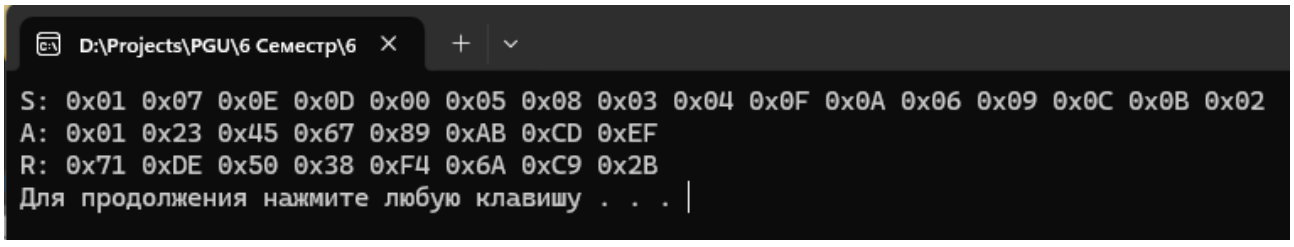

Рисунок 4 — Работа программы LB3_4_Shift-Replace-2x4bit

3.5 Было выполнено преобразование файла с использованием преобразования из п. 4. Файлы *input.txt* и *output.txt* представлены в репозитории

на gitgub. Код программы для преобразования файла находится в репозитории на github в файле *LB3_5_transformation.cpp*.

3.6 Была реализована комбинация операций подстановки и перестановки с использованием эквивалентной подстановки S’(таблица замены S’ была сформирована на основе подстановки S* и сдвига – Таблица 7). Код программы для преобразования файла находится в репозитории на github в файле *LB3_6-7_transformation.cpp*. Результат преобразования находится в репозитории на github в файле *output6-7.txt*.

Таблица 7 - Таблица замены S’

0x9E	0xD3	0xF7	0x9F	0x0B	0xEF	0xAB	0x44	0x5F	0xEB	0x3F	0x96	0x2B	0x88	0x1E	0x60
0x48	0x38	0x0C	0x0F	0x66	0x21	0xDE	0xD0	0xCB	0x12	0xB5	0x50	0x6A	0xA8	0x83	0xF8
0xFC	0x7F	0x26	0x71	0xEA	0x6E	0xA1	0x2E	0xD6	0x07	0x9D	0x25	0x34	0x9C	0xB0	0xF4
0x51	0x52	0xC3	0x4C	0x84	0x4F	0xBC	0x61	0x64	0x69	0x82	0xD7	0x10	0x75	0xCE	0x18
0x41	0xE2	0x76	0x43	0x17	0x81	0x1C	0xB7	0xA4	0x7E	0x8D	0x4B	0x49	0xF5	0x6C	0xAE
0x9A	0x6B	0xE7	0x94	0x16	0x33	0x53	0xAF	0x35	0x36	0xD5	0x74	0xFF	0xEE	0xC9	0x40
0xDD	0xD1	0x9B	0xC4	0x78	0xDB	0x3A	0x86	0x45	0x67	0xE9	0x3D	0xE6	0x37	0x5C	0xC8
0x27	0x6D	0x0D	0x5A	0xB3	0xB6	0xCF	0xAA	0xA5	0x2D	0xC2	0x85	0xC5	0xDA	0x6F	0xCA
0x5D	0xF2	0x1D	0x54	0x62	0x13	0xA6	0xA7	0xB9	0x32	0x7D	0x97	0xA9	0xBF	0x1F	0x06
0xBA	0x8B	0xA3	0x3B	0xEC	0x70	0x1B	0xF1	0x68	0x5E	0x59	0x31	0xF3	0x3C	0x89	0x99
0x42	0xBD	0x24	0x11	0xD2	0xB8	0x73	0x77	0x0A	0xC7	0x20	0x28	0x39	0x08	0xCC	0x95
0xFD	0xB2	0x57	0xF6	0x55	0x91	0x4D	0x8C	0x29	0x05	0x01	0xDC	0x63	0xBE	0x02	0xE1
0xDF	0xA0	0x7C	0xB1	0x72	0xD4	0x2F	0xFB	0x30	0x23	0x7A	0xE5	0xC6	0xE0	0xF0	0xC1
0x19	0x79	0x5B	0x2C	0xD9	0x09	0xB4	0x92	0xBB	0xAC	0x47	0xE3	0x8F	0x00	0xD8	0xF9
0xA2	0xC0	0x14	0x15	0x58	0x56	0xFE	0x93	0xE4	0x46	0x8A	0x80	0x3E	0x4A	0xED	0x65
0x90	0x1A	0xCD	0x87	0x98	0xAD	0x4E	0x03	0xE8	0x7B	0x04	0x22	0x0E	0x2A	0x8E	0xFA

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