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

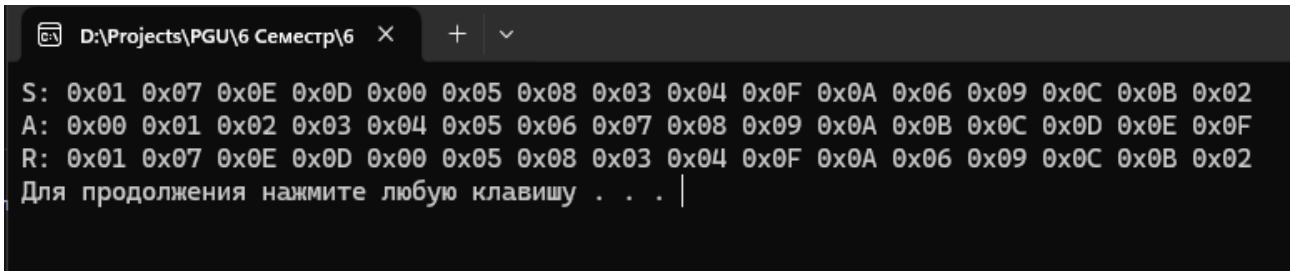

Рисунок 1 — Работа программы LB3_1_Replace-4bit

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*

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

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

A	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF
Replace-8bit	0x17	0xED	0x05	0x83	0x4F	0xA6	0x9C	0xB2

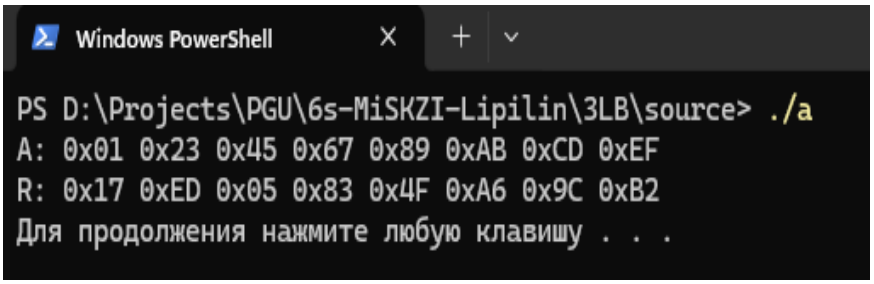

Рисунок 2 — Работа программы LB3_2_Replace-8bit

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)

A	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF
Shift-8bit	0x10	0x32	0x54	0x76	0x98	0xBA	0xDC	0xFE

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)

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

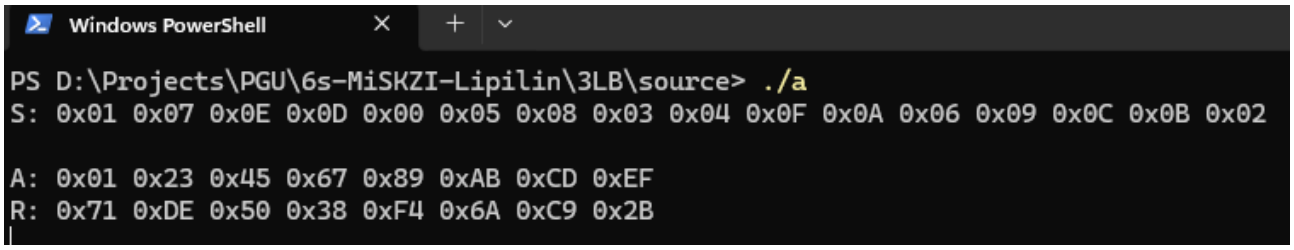

Рисунок 4 — Работа программы LB3_4_Shift-Replace-2x4bit

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’

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

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