Лабораторная работа 3

Программная реализация операций подстановки и перестановки

Цель: получение навыков по программной реализации основных криптографических преобразований.

Теоретические сведения

Подстановка является элементарной операцией, реализующей нелинейное биективное преобразование. Подстановка реализуется в виде таблицы замены, при этом заменяемое значение является адресом заменяющего значения. Во многих источниках таблицу замены называю *S-блок* (S-box, от англ. substitution). Таблица замены для блока из n бит содержит 2^n значений, представляющих собой один из вариантов перестановки возможных комбинаций из 2^n бит. Всего количество возможных таблиц $2^n!$ для блока n бит. Таблицы замены являются фиксированными и не составляют секрета.

При программной реализации подстановки реализуются с помощью программно-индексированного чтения из области памяти: заменяемый элемент x является адресом, по которому считывается заменяющий его элемент y из оперативной памяти:

$$y=S[x],$$

где S – таблица замены. Реализация подстановки требует выполнения одной команды чтения значения по адресу и команды загрузки адреса в регистр. Для хранения таблицы замены требуется объем памяти, определяемый соотношением:

$$|S|=2^{|x|},$$

где $|x|$– размер заменяемого блока. Поскольку объем памяти, требуемый для хранения таблицы подстановок, растет экспоненциально с увеличением размера блока, то подстановки выполняют над частями x_i преобразуемого блока, чаще всего размером от 4 до 8 бит. Для преобразования каждой части блока

используется своя таблица подстановок. Таким образом, для выполнения подстановки требуется объем памяти:

$$|S| = m2^{|x_i|},$$

где m – количество подблоков преобразуемого блока, $|x_i|$ - размер каждого подблока.

Таблица замены задается в виде вектора заменяемых значений. Например, для вектора x размером 4 бита таблица замены может быть следующей:

S={C, 4, 6, 2, A, 5, B, 9, E, 8, D, 7, 0, 3, F, 1}.

Тогда подстановка вектора x описывается следующим образом:

$$y = Vec_4 S\ (Int_4(x)).$$

При программной реализации преобразования из вектора в элемент кольца не выполняются, поскольку число представляется в компьютере в виде двоичного вектора. Для хранения в памяти таблицы замены используется массив, который может быть объявлен следующим образом:

```
char S[16]={0x0C, 0x04, 0x06, 0x02, 0x0A, 0x05, 0x0B, 0x09, 0x0E, 0x08, 0x0D, 0x07, 0x00, 0x03, 0x0F, 0x01};
```

Результатом подстановки вектора x={0111} будет вектор

$S[x]= S[7]$=0x09={1001}.

Объем памяти, требуемый для хранения этой таблицы замены 16 байт, хотя для хранения таблицы замены для блока 4 бита необходимо 8 байт. Неэффективное расходование памяти связано с тем, что архитектура процессоров не содержит эффективных команд для работы с тетрадами. В примере при задании таблицы замены в старшие разряды каждого байта заполнялись нулями. В таком случае в программных реализациях часто используются специальные таблицы, предназначенные для выполнения подстановки сразу над двумя смежными подблоками. Для рассмотренного примера результатом подстановки двух смежных векторов x_1={0111} и x_2={0111} будет вектор

$S[x_1] \,||\, S[x_2]= S^*[x_1||x_2]$={10011001}.

Размер таблицы замены *S** в этом случае будет составлять 256 байт. Очевидно, что экономии памяти это достичь не позволяет, но позволяет повысить эффективность реализации, поскольку все действия выполняются сразу над байтами данных. Таблица замены *S** в этом случае формируется следующим образом:

```
x1=0…n
    x2=0…n
        S*[x1||x2]=S[x1]||S[x2]
```

Перестановка отдельных бит или частей преобразуемого блока реализуется с целью распространения влияния отдельных частей преобразуемого блока на весь шифртекст. Блок, предназначенный для выполнения преобразования, обычно называют *P-блок* (P-box, от англ. permutation). Описывается *P*-блок таблицей перестановок, задающей соответствие позиций бит входного и выходного блока. Например *x*=(0,0,1,1), *P*=(4,1,3,2). Первое значение таблицы перестановки означает, что на первой позиции в блоке результата будет записано значение из четвертой позиции входного блока: *y*=(1,0,1,0). Выполнение перемешивания путем задания позиций в выходном блоке является неэффективном при программной реализации – для выполнения перестановки блока из *n* бит требуется по две команды (чтение индекса и запись) на каждый бит блока, т.е. 2*n* тактов процессора.

Другим способом задания перестановок является использование операций циклического сдвига на определенное число позиций, реализуемых с использованием регистров сдвига. Например, результатом циклического сдвига вправо на 3 позиции над двоичным вектором *x*={0111} будет вектор *y*={1110}.

Если за подстановкой следует фиксированная операция перестановки, то их можно заменить одной эквивалентной подстановкой. Например, выполняется подстановка *S* над вектором *x*, затем над результатом подстановки – перестановка циклическим сдвигом на *p* бит вправо. Тогда эквивалентную подстановку (таблицу замены *S′*) можно определить следующим образом:

```
x=0…n
    S′[x]=(S[x])>>p
```

Задание на лабораторную работу

1) Реализовать операцию подстановки (замены) для 4-х битного вектора. Таблица замены указана в варианте. Для проверки выполнить операцию подстановки над 64-х битным вектором

a={0000000100100011010001010110011111000100110101011110011011110111 1},

$$a \in V_{64}$$

(двоичный вектор a разбивается на 16 частей, над каждой частью выполняется замена).

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*.

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).

4) Реализовать комбинацию операций подстановки и перестановки, указанную на рисунке 1 (входные данные считываются блоком x по 8 бит, блок разбивается на 2 двоичных вектора x_1 и x_2 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).

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

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).

7) Выполнить преобразование файла из п.6 с использованием эквивалентной подстановки. Сравнить результаты. Указать размер таблицы S′.

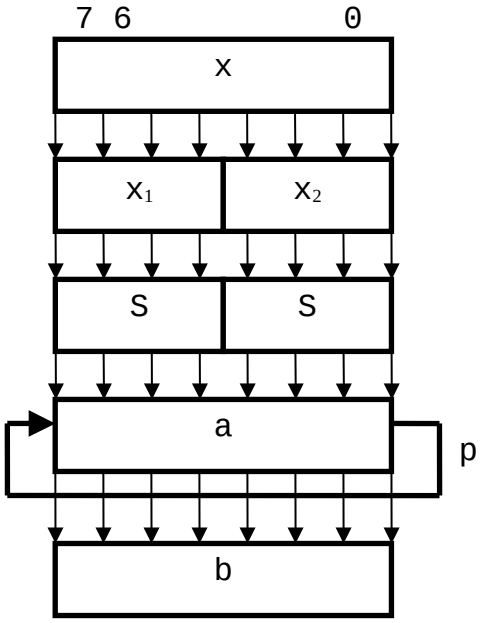

Рисунок 1 – Схема преобразования

Варианты задания.

Таблица замены задана шестнадцатеричным представлением 4х битного вектора (элементами кольца Z_{16}).

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

7	8, E, 2, 5, 6, 9, 1, C, F, 4, B, 0, D, A, 3, 7	3
8	1, 7, E, D, 0, 5, 8, 3, 4, F, A, 6, 9, C, B, 2	4
9	2, A, 5, B, 9, E, 8, D, C, 4, 6, 7, 0, 3, F, 1	5
10	6, 8, 2, 3, 9, 7, B, D, 0, F, A, 5, C, 1, E, 4,	3
11	8, 2, B, 3, 5, A, D, E, 1, F, 7, 4, C, 9, 6, 0	4
12	C, 8, 2, , F, 6, 7, 0, A, 5, 1, D, 43, 9, B, E	5
13	7, F, 5, D, E, B, A, 0, 9, 3, 8, 1, 6, 4, 2, C	3
17	5, D, F, C, A, B, 8, 7, 1, 4, 3, E, 0, 6, 9, 2	4
15	8, E, 1, C, F, 4, B, 2, 5, 6, D, A, 3, 7, 9, 0	5
16	1, 7, D, 0, A, 6, 9, C, 5, 8, 3, E, 4, F, B, 2	6