Тема 4. SP-сеть

Практическая часть

Практическая часть

Составить компьютерную программу (на любом языке программирования), которая выполняет следующие действия:

1. По заданной текстовой строке, состоящей из **двух** символов, возвращает строку, зашифрованную с помощью SP-сети, состоящей из двух P-блоков и промежуточной батареи S-блоков.
2. По заданной **двухбуквенной** строке, зашифрованной с помощью с помощью SP-сети, состоящей из двух P-блоков и промежуточной батареи S-блоков, возвращает строку-оригинал.

SP-сеть работает следующим образом:

На вход подаётся сообщение, разбитое на части. Положим, что одна часть представляет собой **два символа**. Коды этих символов переводятся в 16-битные двоичные числа, так что возникает 32-битная последовательность. P-блок выполняет перестановку этих бит (данная перестановка известна как отправителю сообщения, так и его получателю). Далее последовательность из 32 бит разбивается на части по 4 бита и каждые 4 бита отправляются на S-блок. В S-блоке последовательность из 4 бит переводится в обычное целое число из диапазона от 0 до 15 и ему сопоставляется другое целое число из указанного диапазона согласно некоторой перестановке списка чисел от 0 до 15 (эта перестановка также известна как отправителю, так и получателю). После этого полученное целое число переводится в двоичную 4-битную последовательность. После того, как все 4-битные последовательности будут зашифрованы S-блоком, они склеиваются в 32-битную последовательность и снова проходят через P-блок (в прямом ходе). На выходе 32-битной последовательности сопоставляются два кода символов и, наконец, сами два символа шифрованного сообщения.

ВНИМАНИЕ! Перед процессом шифрования и расшифрования необходимо сгенерировать два небольших списка (где-то с 10-20 элементами) с перестановками списка чисел от 0 до 15 и списка чисел от 0 до 31. Именно из них будут извлекаться по номеру шифрующие перестановки.

Из плана работы SP-сети следует состав проекта. Какие подпрограммы можно разработать (**их перечень может быть другой**)?

1. Подпрограмму, которая по заданному символу возвращает 16-разрядную последовательность нулей и единиц (строку), являющуюся двоичным представлением кода заданного символа.

   К примеру: Ж → 0000010000010110

2. Подпрограмму, которая по строке, состоящей из двух (а потенциально из нескольких) символов, возвращает строку, состоящую из 16-битных представлений кодов заданных символов. Если символов два, то длина получающейся строки равна 32.

   К примеру: ЖП → 00000100000101100000010000011111

3. Подпрограмму, которая по списку десятичных целых чисел [0, 1, 2, ...] возвращает его случайную его перестановку (будет применяться для длинных списков). Чтобы получить случайную перестановку, необходимо случайным образом несколько раз (к примеру, 64 раза) переставить пары элементов исходного списка.
4. Подпрограмму, которая по списку десятичных целых чисел [0, 1, 2, ...] возвращает список **n** его случайных перестановок (будет применяться для длинных списков). При этом важно проверить, чтобы полученные перестановки отличались от исходного списка.
5. Подпрограмму, которая реализует шифрующий P-блок, который по заданной 32-битной последовательности нулей и единиц возвращает её перестановку, созданную по правилу, которое определяет некоторая перестановка списка чисел A = [0, 1, ..., 31], выбранная из списка случайных перестановок **по номеру**. Условимся, что P-блок будет шифровать только 32-битные последовательности.

К примеру, пусть процедура, указанная в п. 2, создала список перестановок элементов списка A, и стороны, участвующие в пересылке информации, договорились взять перестановку под номером 3, которая имеет вид:
[4, 7, 2, 21, 10, 5, 25, 15, 9, 24, 23, 3, 1, 18, 0, 12, 22, 19, 29, 16, 14, 20, 27, 30, 26, 17, 8, 13, 28, 11, 31, 6]
Тогда последовательность 00000100000101100000010000011111, соответствующая строке ЖП, будет переведена в последовательность 00010100000000000010101100011110. Это означает, что на позицию 0 будет переставлен бит, стоящий на 4 месте, на позицию 1 будет переставлен бит, стоящий на 7 месте и так далее.

6. Подпрограмму, которая реализует расшифровывающий P-блок, который по заданной 32-битной последовательности нулей и единиц и по номеру перестановки списка чисел A = [0, 1, ..., 31], возвращает оригинал 32-битной последовательности.
7. Подпрограмму, которая переводит число из двоичной нумерации (представлено в строковой форме и, возможно, с ведущими нулями) в десятичную (представлено числом).

К примеру, строке 0111 должно быть сопоставлено число 7.

8. Подпрограмму, которая из десятичной нумерации переводит число в двоичную нумерацию (строку) заданной разрядности.

К примеру, числу 7 должна быть сопоставлена строка 0111 при заданной разрядности в 4 символа.

9. Подпрограмму, которая реализует шифрующий S-блок, на вход которого подаётся четырёхбитная последовательность нулей и единиц, которая преобразуется в число из диапазона от 0 до 15, ему сопоставляется с помощью выбранной по номеру перестановки новое число из диапазона от 0 до 15, а указанному десятичному числу – двоичная четырёхбитная последовательность.
10. Подпрограмма, которая реализует расшифровывающий S-блок, который по четырёхбитной последовательности нулей и единиц возвращает четырёхбитную последовательность-оригинал.
11. Подпрограмму, которая разрезает заданную 32-битную последовательность нулей и единиц на части из 4 бит, каждую **шифрует** S-блоком и далее собирает из зашифрованных частей новую 32-битную последовательность. Тем самым

будет реализована батарея из восьми S-блоков. Заметим, что каждый S-блок батареи шифрует одной и той же перестановкой.

12. Подпрограмму, которая разрезает заданную 32-битную последовательность нулей и единиц на части из 4 бит, каждую **расшифровывает** S-блоком и далее собирает из зашифрованных частей новую 32-битную последовательность. Тем самым будет реализована батарея расшифровки из восьми S-блоков. Заметим, что каждый S-блок батареи шифрует одной и той же перестановкой.
13. Подпрограмму, которая последовательности из нулей и единиц сопоставляет последовательность букв (каждой букве отводится 16 бит последовательности).

Основная программа должна состоять из двух частей:

1. Шифрование сообщения:
    a. двухбуквенную строку переводит в 32-битную последовательность;
    b. шифрует последовательность P-блоком;
    c. шифрует полученную последовательность батареей из восьми S-блоков;
    d. шифрует последовательность P-блоком, возвращает окончательный результат в виде двух букв.
2. Расшифрование сообщения:
    a. двухбуквенную строку переводит в 32-битную последовательность;
    b. расшифровывает последовательность расшифровывающим P-блоком;
    c. расшифровывает полученную последовательность батареей из восьми расшифровывающих S-блоков;
    d. расшифровывает последовательность расшифровывающим P-блоком, возвращает окончательный результат в виде двух букв.

Пример работы программы

```
========== Шифрование ==========
Исходное сообщение: ЖП
Битовая форма исходного сообщения: 00000100000101100000010000011111
Зашифрованная p-блоком битовая форма: 00000011000000011100010011100100
Зашифрованная батареей s-блоков битовая форма: 10100111101011001111101111011011
Зашифрованная p-блоком битовая форма: 10111110101011100110111110101101
Зашифрованное сообщение: 뿢瀉
========== Расшифрование ==========
Зашифрованное сообщение: 뿢瀉
Расшифрованная p-блоком битовая форма: 10100111101011001111101111011011
Расшифрованная батареей s-блоков битовая форма: 00000011000000011100010011100100
Расшифрованная p-блоком битовая форма: 00000100000101100000010000011111
Расшифрованное сообщение: ЖП
```