Лабораторная работа №4

Алгоритмы блочного шифрования.

Задание.

Напишите на языке Python или Java программу, в которой реализован блочного шифрования на основе ячейки Фейстеля. В программе должны быть функции:

- crypt, в которых передается строка сообщения длинной 8 символов, и ключ как строка 16 символов, а результат как строка зашифрованного текста длинной 8 символов.
- decrypt, в которых передается строка шифротекста длинной 8 символов, и ключ как строка 16 символов, а результат как строка расшифрованного текста длинной 8 символов.

Параметры шифрования таковы:

- Размер ключа: 128
- Размер блока данных: 64
- Раундов шифрования: 16
- S-блоков: 2x8 блока по 4 бита на каждый раунд отдельные
- P-блок: заменен циклическим сдвигом влево на 11

Ниже приведены схемы реализации S-блоков и общий алгоритм зашифровывания и расшифровывания.

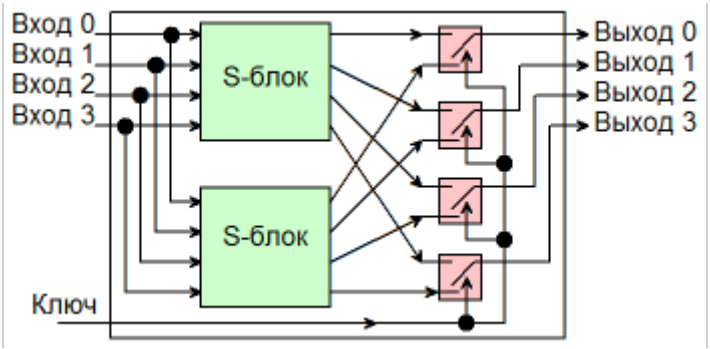


Рисунок 1: схема обработки 4х битов данных.

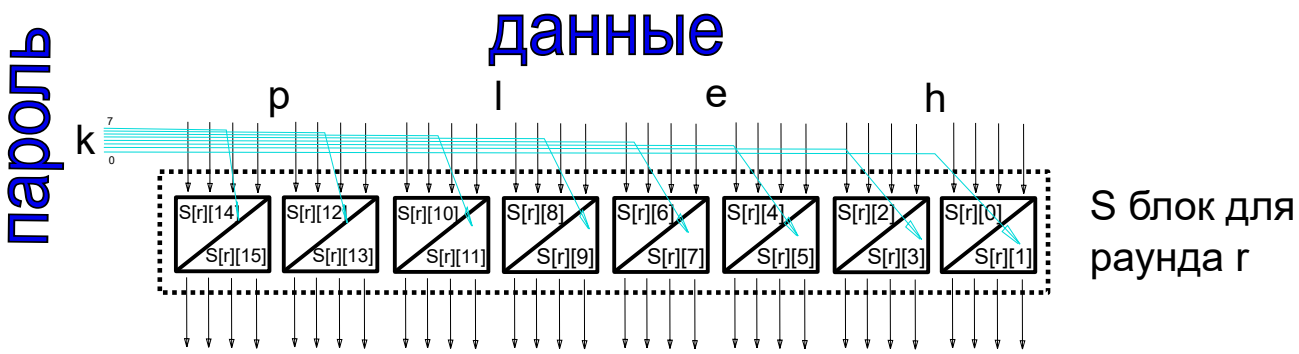


Рисунок 2: S-блок для одного раунда

Алгоритм зашифровывания и расшифровывания

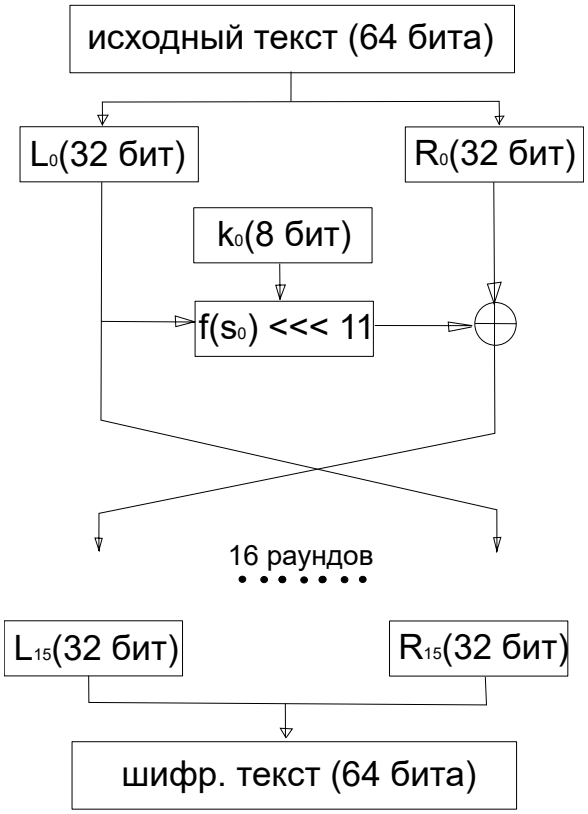


Рисунок 4: зашифровывание

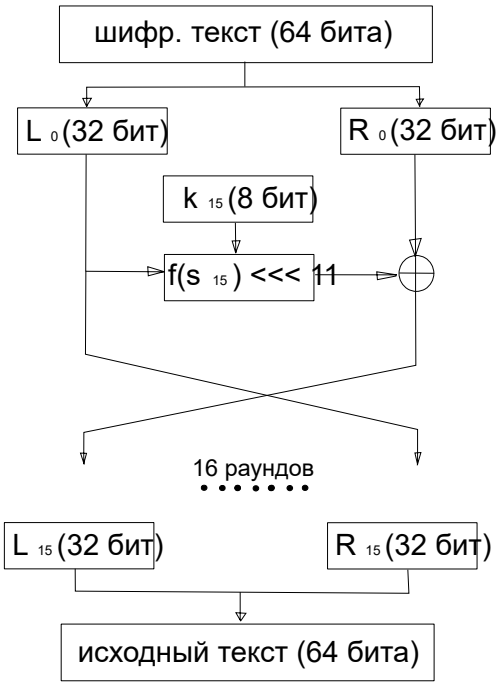


Рисунок 3: Расшифровывание

К материалам лабораторной работы прилагаются заготовки файлов java и python программы на основании которых необходимо решать задачу.

Пример вывода программы

```
ROUND: 0

  {11,11,3,11,10,0,12,4,9,11,9,3,3,4,7,1,},

  {10,12,8,4,7,6,15,8,14,1,5,4,7,0,9,9,},

  {10,3,12,13,2,14,1,3,6,14,4,9,9,1,11,14,},

  {7,3,2,11,0,10,6,9,7,6,4,15,7,10,15,11,},

  {4,15,14,1,15,9,9,10,4,12,0,13,14,8,10,5,},

  {0,4,9,3,5,4,1,4,11,2,14,10,0,2,11,11,},

  {1,5,1,3,8,8,12,11,12,0,14,14,6,12,5,4,},

  {3,11,9,12,5,2,4,6,10,12,1,1,10,0,9,8,},
```

{3,2,8,11,7,7,13,6,9,10,15,11,15,13,12,4,},

{6,10,14,4,9,1,1,3,7,4,10,13,10,7,7,10,},

{4,14,2,5,10,5,0,2,10,10,12,1,12,12,5,11,},

{13,0,6,12,6,0,14,7,2,8,12,7,0,10,4,5,},

{12,10,15,0,2,4,13,3,11,12,11,10,11,1,0,5,},

{3,9,7,12,6,8,8,14,2,8,12,12,5,14,8,13,},

{9,4,7,8,15,14,4,0,12,0,1,13,12,6,6,4,},

{4,4,9,14,10,1,3,15,2,4,5,0,3,0,4,14,},

……..

ROUND: 15

{1,0,3,6,9,6,13,6,9,2,0,8,4,10,14,2,},

{2,15,2,2,0,8,15,5,15,1,13,11,14,14,12,15,},

{3,14,3,7,1,4,13,13,1,8,5,14,2,4,8,0,},

{1,14,11,9,12,2,0,0,13,12,14,12,3,5,14,7,},

{1,8,1,0,15,5,14,14,15,7,15,5,4,11,3,3,},

{2,3,15,12,3,3,8,2,3,9,3,9,11,5,4,2,},

{5,12,13,3,14,15,2,9,13,9,13,15,7,14,14,1,},

{3,3,2,0,9,9,13,4,10,8,0,12,5,1,12,9,},

{14,12,10,6,0,13,8,0,9,4,12,3,8,3,12,3,},

{2,1,8,3,0,12,6,4,3,0,15,7,2,6,0,1,},

{10,10,8,7,11,11,2,9,7,12,3,6,2,10,8,4,},

{13,7,3,10,7,0,9,1,4,15,10,2,8,12,5,11,},

{4,13,6,14,15,10,1,3,7,13,11,5,3,3,4,7,},

{1,6,2,4,5,9,14,13,10,6,10,8,13,15,5,6,},

{14,5,4,2,6,9,5,3,0,5,2,7,0,11,2,4,},

```
  {13,15,15,3,7,3,11,9,15,14,3,5,15,1,5,2,},

==========

исходные данные(2x32бит): "helpword"

ключ шифрования(128 бит): "key for feistel "

зашифрованные данные: Í👽6î©ª¢

расшифрованные данные: helpword
```

Результат работы.

В качестве результата предоставьте работающую программу на Java или Python, написанную на основе предоставленного образца. Программа должна зашифровывать заданный текст, заданным ключем. Также программа должна расшифровать тем же ключем шифротекст к исходному сообщению. Важно: программа будет проверятся специальной тестировочной программой на соответствие описанному алгоритму.
Дополнительные вопросы:

- Назовите близкие алгоритмы
- Обладает ли данный алгоритм лавинным эффектом.
- Продемонстрируйте лавинный эффектом, если он есть.
- Оцените криптостойкость алгоритма к атаке прямым перебором (количественно).
- Оцените общую криптостойкость — достоинства, недостатки.
- Как повысить надежность алгоритма.

Пример оформления кода на Python.

файл **lab4.py**

```
# This is a sample Python script.
import random

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
INT_BITS = 32
ROUNDS=16;

# -- feistel parameters
# разрядность блока данных для криптографии, менять нельзя т.к. определяет
# тип int функции фейстеля
DATA_BLOCK_WIDE = 32
#  разрядность S-блока (4)
S_BLOCK_WIDE=4
MAGIC_ROTATE=11
# разрядность ключа шифрования (128)
KEY_SIZE=int(ROUNDS*DATA_BLOCK_WIDE/S_BLOCK_WIDE)
# количество S-блоков в раунде (16)
S_BLOCKS=int(2*DATA_BLOCK_WIDE/S_BLOCK_WIDE)

 # -- блоки сети фейстеля
s =  [[[0 for x in range(int(2**S_BLOCK_WIDE))] for y in range(S_BLOCKS)]  for z in range(ROUNDS)]
#s = [ROUNDS][S_BLOCKS][int(2**S_BLOCK_WIDE)] # 16,16,16

def generate(studentNum):
    # import javarandom
    # rnd=javarandom.Random(1)
```

```
    random.seed(1)
    for  r in range(0,len(s)):
        print("ROUND: {}".format(r))
        for i in range(0, len(s[r])):
            print("  {",end='');
            for j in range(0, len(s[r][i])):
                # s[r][i][j] = rnd.nextInt(len(s[r][i]))
                s[r][i][j] = random.randint(0,len(s[r][i])-1)
                print("  {},".format(s[r][i][j]),end='');
            print("},")

def str2int(s):
    rez=0;
    for i in range (0,4):
        rez|=(ord(s[i])&255)<<(i*8)
    return rez

def int2str(l):
    rez=""
    for i in range (0,4):
        rez+=chr(l&255)
        l>>=8
    return rez

def leftRotate(n, d):
    return (n << d)|(n >> (INT_BITS - d))


def rightRotate(n, d):
    return (n >> d)|(n << (INT_BITS - d)) & 0xFFFFFFFF


# TODO

def crypt(message,pass_key):
    # TODO

def decrypt(message,pass_key):
    # TODO


def main():
    generate(100)
    str="helpword"
    pass_key="key for feistel "
    print("==========\nисходные данные(2x32бит): \"{}\"".format(str))
    print("ключ шифрования(128 бит): \"{}\"".format(pass_key))
    rez=crypt(str,pass_key)
    print("зашифрованные данные: "+rez)
    rez=decrypt(rez,pass_key)
    print("расшифрованные данные: "+rez)



# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    main()
```

Пример оформления кода на Java.

файл **lab4.java**

```
import java.util.Random;


public class lab4 {

    // -- cipher parameters
```

```
// раундов шифрования (16)
public static final int ROUNDS = 16;

// -- feistel parameters
// разрядность блока данных для криптографии, менять нельзя т.к. определяет
// тип int функции фейстеля
public static final int DATA_BLOCK_WIDE = 32;
// разрядность S-блока (4)
public static final int S_BLOCK_WIDE = 4;
public static final int MAGIC_ROTATE = 11;
// разрядность ключа шифрования (128)
public static final int KEY_SIZE = ROUNDS * DATA_BLOCK_WIDE / S_BLOCK_WIDE;
// количество S-блоков в раунде (16)
public static final int S_BLOCKS = 2 * DATA_BLOCK_WIDE / S_BLOCK_WIDE;

// -- блоки сети фейстеля
static int s[][][] = new int[ROUNDS][S_BLOCKS][(int) Math.pow(2, S_BLOCK_WIDE)]; // 16,16,16

static void generate(int studentNum) {
    Random rand = new Random(studentNum);
    for (int r = 0; r < s.length; r++) {
        System.out.printf("ROUND: %d\n", r);
        for (int i = 0; i < s[r].length; i++) {
            System.out.print("  {");
            for (int j = 0; j < s[r][i].length; j++) {
                s[r][i][j] = rand.nextInt(s[r][i].length);
                System.out.print(s[r][i][j] + ",");
            }
            System.out.println("},");
        }
    }
}

static int str2int(String s) {
    int rez = 0;
    for (int i = 0; i < 4; i++) {
        rez |= (s.charAt(i) & 255) << (i * 8);
    }
    return rez;
}

static String int2str(int l) {
    String rez = "";
```

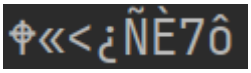
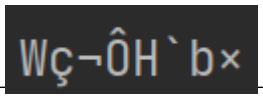
```
        for (int i = 0; i < 4; i++) {
            rez += (char) (l & 255);
            l >>= 8;
        }
        return rez;
    }
    //TODO
    public static String crypt(String message,String pass_key) {
       //TODO
    }
    public static String  decrypt(String message,String pass_key) {
       //TODO
    }
    public static void main(String[] args) {
        generate(100);
        String str="helpword";
        String pass_key="key for feistel ";
        System.out.printf("==========\nисходные данные(2x32бит): \"%s\"\n",str);
        System.out.printf("ключ шифрования(128 бит): \"%s\"\n",pass_key);
        String rez=crypt(str,pass_key);
        System.out.println("зашифрованные данные: "+rez);
        rez=decrypt(rez,pass_key);
        System.out.println("расшифрованные данные: "+rez);
    }
}
```

Варианты заданий.

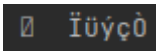
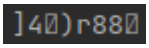
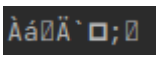
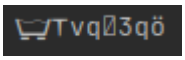
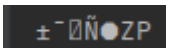
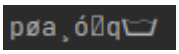
Внимание! Выходные данные могут выглядеть иначе, в зависимости от набора установленных системных шрифтов.

Для Java

Вариант	Входные данные	Выходные данные
1	StudentNum=1, str="baracuda", pass_key="feistel cipher 1"	áìäüÌs¼ú
2	StudentNum=2, str="piroman1", pass_key="password for my1"	×ß□ÓUí¼T
3	StudentNum=3, str="privet11", pass_key="hotel california"	ó¢■Û9Ø4□
4	StudentNum=4, str="hellosam", pass_key="abracadabra arba"	□Þ^Éé■Fé
5	StudentNum=5, str="robocop1", pass_key="reboot you servr"	=Ig@☙-»
6	StudentNum=6, str="stimpank", pass_key="rostov on	7Nv□»●hâ

	don 43"	
7	StudentNum=7, str="suburban", pass_key="wild dragon fire"	
8	StudentNum=8, str="hell cat", pass_key="superman dead 12"	
9	StudentNum=9, str="robhound", pass_key="toronto -- tokyo"	
10	StudentNum=10, str="barbosik", pass_key="perpetum mobiles"	

Для Java

1	StudentNum=1, str="baracuda", pass_key="feistel cipher 1"	
2	StudentNum=2, str="piroman1", pass_key="password for my1"	
3	StudentNum=3, str="privet11", pass_key="hotel california"	
4	StudentNum=4, str="hellosam", pass_key="abracadabra arba"	
5	StudentNum=5, str="robocop1", pass_key="reboot you servr"	
6	StudentNum=6, str="stimpank", pass_key="rostov on don 43"	
7	StudentNum=7, str="suburban", pass_key="wild dragon fire"	
8	StudentNum=8, str="hell cat", pass_key="superman dead 12"	
9	StudentNum=9, str="robhound", pass_key="toronto -- tokyo"	
10	StudentNum=10, str="barbosik", pass_key="perpetum mobiles"