Национальный исследовательский университет «МЭИ» Институт автоматики и вычислительной техники

Кафедра вычислительных машин, систем и сетей

Лабораторная работа № 7

«Разработка программной реализации потокового шифра»

по курсу «Защита информации»

Выполнил: Кутузов И.Г.

Группа: А-08-16

Подпись: 

Преподаватель: Рытов А.А.

Москва, 2020 г.

```
In[ ]:= nvar = 10;
        (*1. Подготовить открытый текст для шифрования-
         строку,содержащую фамилию,имя и отчество.*)

In[ ]:= mytext = "Кутузов Илья Геннадьевич";


In[ ]:= (*2. Перевести открытый текст в последовательность кодов
         ToCharacterCode["string","encoding"] со спецификацией "ISOLatinCyrillic".*)
        код символа
        mytextencoded = ToCharacterCode[mytext, "ISOLatinCyrillic"]
                        код символа
Out[ ]= {186, 227, 226, 227, 215, 222, 210, 32, 184, 219, 236,
         239, 32, 179, 213, 221, 221, 208, 212, 236, 213, 210, 216, 231}

In[ ]:= (*3. Провести обратное преобразование
         (FromCharacterCode[,]) кодов в символы с той же спецификацией.*)
         символ по его коду
        mytextenchared = FromCharacterCode[mytextencoded, "ISOLatinCyrillic"]
                         символ по его коду
Out[ ]= Кутузов Илья Геннадьевич

In[ ]:= (*4. Провести поразрядное сложение BitXor[,]списка кодов
                                                сложение битов по модулю 2
         из п.2 со случайным числом из диапазона целых[100,200].*)
        SeedRandom[nvar]
        инициализация генератора псевдослучайных чисел
        ri = RandomInteger[{100, 200}]
             случайное целое число
        mytextencoded2 = BitXor[mytextencoded, ri]
                         сложение битов по модулю 2
Out[ ]= 185

Out[ ]= {3, 90, 91, 90, 110, 103, 107, 153, 1, 98, 85,
         86, 153, 10, 108, 100, 100, 105, 109, 85, 108, 107, 97, 94}

In[ ]:= (*5. Преобразовать коды в символы и зафиксировать результат.*)


In[ ]:= mytextenchared2 = FromCharacterCode[mytextencoded2, "ISOLatinCyrillic"]
                          символ по его коду
Out[ ]= Z [Zngk bUV
        lddimUlka^

In[ ]:= (*6. Провести повторное поразрядное сложение шифртекста п.4
         с тем же самым случайным числом и восстановить открытый текст.*)
        mytextenchared3 = FromCharacterCode[BitXor[mytextencoded2, ri], "ISOLatinCyrillic"]
                          символ по его коду    сложение битов по модулю 2
Out[ ]= Кутузов Илья Геннадьевич

In[ ]:= (*7. Подготовить два массива (Array) s и k
                                      массив
         длиной в 256 элементов и со смещением (origin) равным 0.*)
```

In[ ]:=
```
Origin = 0;
Array[S, 256, 0];
```
массив
```
Array[K, 256, 0];
```
массив
```
Length[s]
```
длина
```
Length[k]
```
длина

Out[ ]= 0

Out[ ]= 0

In[ ]:=
```
(*8. Инициализировать массив s линейно (Range) целыми числами от 0 до 255*)
```
диапазон

In[ ]:=
```
Do[S[i] = i, {i, 0, 255}]
```
оператор цикла

In[ ]:=
```
(*9. Установить генератор случайных чисел в начальное состояние с параметром N-
```
чис
```
 номером по списку в группе и инициализировать
   массив k случайными целыми числами из диапазона 0-255.*)
SeedRandom[nvar];
```
инициализация генератора псевдослучайных чисел
```
Do[K[i] = RandomInteger[255], {i, 0, 255}]
```
оператор··· случайное целое число

In[ ]:=

In[ ]:=
```
(*10. Сформировать s-блок,
 выполнив следующие операции:Установим значение индекса j равным 0.
Затем:Для i от 0 до 255
j=(j+Si+Ki) mod 256
Поменяйте местами Si и Sj.*)
j = 0;
Do[
```
оператор цикла
```
 j = Mod[(j + S[i] + K[i]), 256];
```
остаток от деления
```
 {S[i], S[j]} = {S[j], S[i]};
 , {i, 0, 255}]
```

In[ ]:=
```
(*Сформировать случайный байт,
выполнив следующие операции:В алгоритме применяются два счетчика i и
   j с нулевыми начальными значениями.Чтобы сгенерировать случайный байт,
выполните следующие операции:i=(i+1) mod 256;
j=(j+Si) mod 256;
Поменяйте местами Si и Sj;
t=(Si+Sj) mod 256;
K=St
Байт K используется в операции BitXor с открытым текстом для получения шифртекста
                                        сложение битов по модулю 2
  или в операции BitXor с шифртекстом для получения открытого текста.*)
                  сложение битов по модулю 2
```

In[ ]:=
```
i = 0;
j = 0;
i = Mod[i + 1, 256];
    остаток от деления
j = Mod[j + S[j], 256];
    остаток от деления
{S[i], S[j]} = {S[j], S[i]};
t = Mod[S[i] + S[j], 256];
    остаток от деления
ByteK = S[t]
{S[i], S[j]} = {S[j], S[i]};
```

Out[ ]= 162

In[ ]:=
```
(*12. Зашифровать,с применением операции BitXor[,] первый символ открытого
                                          сложение битов по модулю 2
 текста.Аналогичным образом расшифровать первый символ шифртекста.*)
mytextencoded[[1]] = BitXor[mytextencoded[[1]], ByteK];
                     сложение битов по модулю 2
FromCharacterCode[mytextencoded, "ISOLatinCyrillic"]
символ по его коду
mytextencoded[[1]] = BitXor[mytextencoded[[1]], ByteK];
                     сложение битов по модулю 2
FromCharacterCode[mytextencoded, "ISOLatinCyrillic"]
символ по его коду
```

Out[ ]= утузов Илья Геннадьевич

Out[ ]= Кутузов Илья Геннадьевич

In[ ]:=
```
(*13. Определить длину открытого текста и провести поточное шифрование,получая для
 каждого символа открытого текста новый случайный байт шифрования (п.11).*)
```

In[ ]:=
```
i = 0;
j = 0;
bytelist = {};
Do[
оператор цикла
  i = Mod[i + 1, 256];
      остаток от деления
  j = Mod[j + S[j], 256];
      остаток от деления
  {S[i], S[j]} = {S[j], S[i]};
  t = Mod[S[i] + S[j], 256];
      остаток от деления
  ByteK = S[t];
  AppendTo[bytelist, ByteK];
  добавить в конец к
  mytextencoded[[l]] = BitXor[mytextencoded[[l]], ByteK];
                       сложение битов по модулю 2
  , {l, Length[mytextencoded]}];
       длина
FromCharacterCode[mytextencoded, "ISOLatinCyrillic"]
символ по его коду
mytextenchared2
```

Out[ ]=
```
ш93
ядWfsOпШЮЋдљ8Г6
```

Out[ ]=
```
Z [Zngk bUV
lddimUlka^
```

In[ ]:=
```
Do[
оператор цикла
 mytextencoded[[z]] = BitXor[mytextencoded[[z]], bytelist[[z]]]
                      сложение битов по модулю 2
 , {z, Length[mytextencoded]}]
      длина
```

In[ ]:=
```
FromCharacterCode[mytextencoded, "ISOLatinCyrillic"]
символ по его коду
bytelist
```

Out[ ]=
```
Кутузов Илья Геннадьевич
```

Out[ ]=
```
{162, 11, 219, 110, 228, 90, 216, 52, 87, 15,
 187, 137, 213, 252, 10, 21, 19, 123, 0, 21, 237, 97, 238, 228}
```