**Лабораторная работа № 3**

**Тема: Шифрование методом гаммирования**

**Цель работы:** Изучить шифрование и дешифрование данных с методом гаммирования

Реализовать процедуру шифрования и дешифрования файлов с использованием указанных методов. Для генерации гаммы использовать мультипликативный датчик со значениями a = 5, m = 4096, Y0 = 4003.

Program Main\_Program;

Const  а=5;  m=4096;  y0=3091;

Var

  Gamma: array [1..8] of byte;

procedure Rnd(var t:array of byte);

{Процедура формирования случайного числа}

Var  I, v : integer;

begin

  y:=y0;

  for i:=1 to 8 do begin   y:=(a\*y) mod m;    t[i]:=y;  end;

end;

procedure Code;

{Процедура кодирования}

Var

 Fin, FOut : text ;

 Buff, TextSh:array [1..8] of byte;

 i,j:integer;  ch : char;

 Mess,CodeMess: string;

 begin

   Assign(FIn, 'Source.txt'); Reset(FIn);

   Assign(FOut,'Coded.txt');  Rewrite(FOut);

   j:=0;

   repeat

Rnd(Gamma);

for i:=1 to 8 do begin   read(FIn,ch);  Buf[i]:=ord(ch);  end;

Mess:=''; CodeMess:='';

for i:=1 to 8 do begin

  TextSh[i]:=Buf[i] xor Gamma[i];

  Mess:=Mess+chr(Buf[i]);

  CodeMess:=CodeMess+chr(TextSh[i]);

  write(FOut,chr(TextSh[i]));

end;

inc(j);

   until Eof(FIn);

   Close(FOut); close(FIn);

 end;

 begin {Main\_Program}

   Code;

 end. {Main\_Program}

1. Что такое гаммирование? Что понимают под гаммой шифра?

Гамми́рование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Гамма шифра – псевдослучайная последовательность, вырабатываемая по определенному алгоритму, используемая для шифровки открытых данных и дешифровки шифротекста.

1. Какие операции можно применять при наложении гаммы?

Операция сложения по модулю два в алгебре логики называется также "исключающее ИЛИ" или по-английски XOR. При использовании метода гаммирования ключом является последовательность, с которой производится сложение – гамма.

1. Какие требования предъявляются к криптографически стойкому генератору ПСП? Почему наиболее важна длина периода гаммы?

К криптографически стойкому генератору псевдослучайной последовательности чисел (гаммы шифра) предъявляются три основных требования:

- период гаммы должен быть достаточно большим для шифрования сообщений различной длины;

- гамма должна быть практически непредсказуемой, что означает невозможность предсказать следующий бит гаммы, даже если известны тип генератора и предшествующий кусок гаммы;

- генерирование гаммы не должно вызывать больших технических сложностей.

По окончании периода числа начнут повторяться, и их можно будет предсказать. Требуемая длина периода гаммы определяется степенью закрытости данных. Чем длиннее ключ, тем труднее его подобрать. Длина периода гаммы зависит от выбранного алгоритма получения псевдослучайных чисел.

1. Опишите линейный конгруэнтный способ генерации ПСП.

Линейный конгруэнтный метод

В большинстве языков программирования именно этот метод используется в стандартной функции получения случайных чисел. Впервые этот метод был предложен Лехмером в 1949 году. Выбирается 4 числа:

Модуль m (m>0);

Множитель a (0<=a<m);

Приращение c (0<=c<m);

Начальное значение X0 (0<= X0<m)

Последовательность получается с использование следующей рекуррентной формулы: Xn+1=(a\* Xn+c) mod m.

Этот метод даёт действительно хорошие псевдослучайные числа, но, если взять числа m,a,c произвольно, то результат нас скорее всего разочарует. При m=7, X0=1, a=2, c=4 получится следующая последовательность: 1,6,2,1,6,2,1,…

Очевидно, что эта последовательность не совсем подходит под определение случайной. Тем не менее, этот провал позволил нам сделать два важных вывода:

Числа m,a,c, X0 не должны быть случайными;

Линейный конгруэнтный метод даёт нам повторяющиеся последовательности.

1. Опишите аддитивный и мультипликативный генераторы ПСП

Аддитивные генераторы (иногда называемые запаздывающими генераторами Фиббоначи) очень эффективны, так как их результатом являются случайные слова, а не случайные биты . Сами по себе они не безопасны, но их можно использовать в качестве составных блоков для безопасных генераторов.

Основная формула для генерации ПСЧ по аддитивному методу имеет вид:

https://helpiks.org/helpiksorg/baza6/318200760833.files/image154.gif , где т — целое число.

Очевидно, что для инициализации генератора, построенного по этому методу, необходимо помимо модуля т задать два исходных члена последовательности. При https://helpiks.org/helpiksorg/baza6/318200760833.files/image156.gif https://helpiks.org/helpiksorg/baza6/318200760833.files/image158.gif ; последовательность превращается в ряд Фибоначчи. Рекомендации по выбору модуля совпадают с предыдущим случаем; длину последовательности можно оценить по приближенной формуле

https://helpiks.org/helpiksorg/baza6/318200760833.files/image160.gif.

Основная формула мультипликативного генератора для расчета значения очередного ПСЧ по значению предыдущего имеет вид:

https://helpiks.org/helpiksorg/baza6/318200760833.files/image142.gif,

где а, т — неотрицательные целые числа (их называют множитель и модуль).

1. Опишите алгоритмы шифровки и дешифровки открытого текста методом гаммирования

Процесс *шифровки* заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2. Следует отметить, что перед шифровкой открытые данные разбивают на блоки Tо(*i*) одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков Гш(*i*) аналогичной длины.

Уравнение шифровки можно записать в виде: Тш(*i*) = Гш(*i*) ⊕ То(*i*) , *i* = 1 ... *M*,

где Тш(*i*) - *i*-й блок шифротекста;

       Гш(*i*) -  *i*-й блок гаммы шифра;

       То(*i*) - *i*-й блок открытого текста;

*M* - Количество блоков открытого текста.

Процесс *дешифровки* сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение дешифровки имеет вид:  То(i) = Гш(i) ⊕ Тш(i) , i = 1 ... M.