Министерство науки и высшего образования РФ

ФГБОУ ВО Тульский государственный университет

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Отчет по выполнению лабораторной работы № 4

Вариант №5

Выполнил: ст. гр. 230791 \_\_\_\_\_\_\_\_\_\_\_ Конак А.С.

(подпись)

Проверил: доц. каф. ИБ \_\_\_\_\_\_\_\_\_\_\_ Басалова Г.В.

(подпись)

Тула, 2021 г

Лабораторная работа 4. Блочные алгоритмы симметричного шифрования.

Цель и задачи работы

Получение практических навыков криптографического преобразования информации с использованием блочных алгоритмов симметричного шифрования типа ГОСТ 34.12-2018 «Магма».

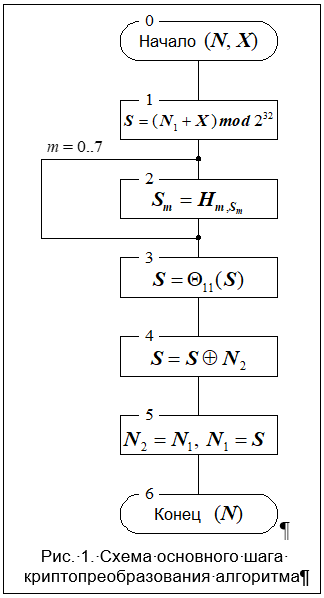
Задание на работу

1. Разработать функции, реализующие базовые циклы зашифрования/расшифрования, аналогичные циклам 32-З и 32-Р ГОСТ 34.12-2018 «Магма».

2. Разработать программы шифрования/расшифрования одного полного блока данных с использованием созданных функций в соответствии с вариантом задания. Размер блока: 128; размер ключа: 16; число основных шагов: 64.

Ход работы

Схема алгоритма шифрования:



Опишем работу алгоритма:

1) На вход функции, выполняющей основное криптопреобразование, подается два числа (две половины исходного блока данных) и ключ.

2) Старшие разряды складываются с ключом.

3) Полученное число разбивается на части. И эти части заменяются в соответствии с таблицей замен. Затем они собираются обратно в число.

4) Полученное число циклически сдвигается на 11 влево.

5) Результат прошлого шага складывается с младшими разрядами.

6) На место младших разрядов записываются старшие. А на место старших – то, что мы получили в результате одной итерации цикла.

Текст программы:

#include<iostream>

#include <vector>

#include<cmath>

using namespace std;

typedef unsigned long long int b64;

int replacementTable[263]

{

252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35,

197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241, 187,

211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206,

204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183,

189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34,

185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78,

51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38,

65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149,

20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1,

142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212,

59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137,

225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97,

93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158,

178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195,

32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166,

116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182,

252, 238, 221, 17, 207, 110, 49

};

vector<b64> encrypt(b64 L, b64 R, b64 key)

{

b64 Z; b64 buff; char zamena; b64 sdvig;

for (int i = 0; i < 64; i++)

{

buff = 0; zamena = 0; Z = 0;

Z = L + key;

for (int j = 0; j < 8; j++)

{

zamena = Z & 255;

buff = buff | ((b64)replacementTable[(int)zamena - 1 + j] << (8 \* j));

Z = Z >> 8;

zamena = 0;

}

sdvig = buff;

sdvig = sdvig << 11;

buff = buff >> 53;

buff = sdvig | buff;

buff ^= R;

R = L;

L = buff;

}

return { L,R };

}

int main()

{

b64 a = 14648;

b64 b = 2;

unsigned short int keyN = 12435;

b64 key=0;

for (int i = 0; i < 4; i++)

{

key = key | keyN;

key = key << 16;

}

vector<b64> res1 = encrypt(a, b, key);

for (int i = 0; i < 2; i++)

{

cout << res1[i] << endl;

}

cout << endl;

vector<b64> res2 = encrypt(res1[1], res1[0], key);

for (int i = 0; i < 2; i++)

{

cout << res2[i] << endl;

}

return 0;

}

Вывод

В ходе выполнения работы по варианту была разработана программа для шифровки одного блока данных.