Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Пенза 2020

Лабораторная работа № 2

**«Разработка программы шифрования/дешифрирования двоичных**

**файлов с использованием перестановочного шифра»**

## Отчёт по дисциплине

**«Защита информации в сети Интернет»**

**Выполнили с-ты гр. 18ВВ1:**

Гаевский Илья

Воеков Иван

Немцев Александр

**Приняли:**

к.т.н. Дубравин А.В.  
Карамышева Н.С.

**Цель работы.**

Научиться разрабатывать программы шифрования/дешифрования двоичных файлов с использованием перестановочного шифра.

**Задание.**

Разработать программу, выполняющую шифрование и расшифровывание произвольного двоичного файла с использованием перестановочного шифра используя в качестве ключа последовательность, соответствующую номеру варианта. Выполнить проверку путем двоичного сравнения исходного файла и фала, полученного после расшифровывания.

Размер блока: 4 байта.

Таблица 1 - Вариант задания.

|  |  |
| --- | --- |
| № Варианта | Последовательность перестановки |
| 3 | 6 29 17 3 26 13 7 0 8 15 12 19 21 5 28 16 23 24 1 2 25 30 31 10 9 14 27 18 22 4 11 20 |

**Листинг.**

#include <fstream>

#include <conio.h>

#include <vector>

#include <iostream>

using namespace std;

void encryption(int key[]) {

int val = 0, c = 0, d = 0, encr = 0;

ifstream ifs("Test.txt", ios::binary);

size\_t size = ifs.seekg(0, ios::end).tellg();

size\_t block = 1024;

ifs.seekg(0);

c = 4 - size % 4;

ofstream fout("Result.txt", ios\_base::trunc | ios::binary);

for (int i = 0; i <= size / 4; i++) {

char \*buf = new char[block];

ifs.read(buf, block);//Считывает заданное число байт из файла в память

for (int j = 0; j < (block / 4); j++) {

int \*link = (j + (int\*)buf);

encr = 0;

for (int i1 = 0; i1 < 32; i1++) {

val = 1 & (\*link >> (31 - key[i1]));

if (val != 0) {

encr = encr | (1 << (31 - i1));

}

val = 0;

}

char buffer[4];

for (int i2 = 1; i2 < 5; i2++) {

buffer[i2 - 1] = encr >> (32 - (i2 \* 8));

}

for (int i3 = 0; i3 < 4; i3++) {

if (d < size + c) {

fout << buffer[3 - i3];

}

d++;

}

}

delete[] buf;

}

fout << c;

}

void decryption(int key[]) {

int val = 0, encr = 0, c = 0;

char minus;

ifstream ifs("Result.txt", ios::binary);

size\_t size = ifs.seekg(0, ios::end).tellg();//функция которая позволяет искать произвольную позицию в файл

size\_t block = 1024;

ifs.seekg(size - 1);//ищем курсор в конце файла

ifs.get(minus);

int intminus = minus - '0';

ifs.seekg(0);

ofstream fout("New.txt", ios\_base::trunc | ios::binary);

for (int i = 0; i <= size / 4; i++) {

char \* buf = new char[size];

ifs.read(buf, size);

for (int j = 0; j < (block / 4); j++) {

int \*link = (j + (int\*)buf);

encr = 0;

for (int i = 0; i < 32; i++) {

val = 1 & (\*link >> (31 - i));

if (val != 0) {

encr = encr | (1 << (31 - key[i]));

}

val = 0;

}

char buffer[4];

for (int i = 1; i < 5; i++) {

buffer[i - 1] = encr >> (32 - (i \* 8));

}

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

if (c < size - intminus - 1) {

fout << buffer[3 - i];

}

c++;

}

}

delete[] buf;

}

}

void menu()

{

char ch\_key;

int key[] = { 6, 29, 17, 3, 26, 13, 7, 0, 8, 15, 12, 19, 21, 5, 28, 16, 23, 24, 1, 2, 25, 30, 31, 10, 9 , 14, 27, 18, 22, 4, 11, 20};

cout << "1 - Encrypt file.\n";

cout << "2 - Decrypt file.\n";

cout << "ESC - Exit.\n";

switch ((ch\_key = \_getch()))

{

case '1':

cout << "File Encryption...\n";

encryption(key);

cout << "Done.\n";

system("pause");

break;

case '2':

cout << "Decryption file...\n";

decryption(key);

cout << "Done.\n";

system("pause");

break;

case 27:

cout << "Exit the program....\n";

exit(0);

break;

default:

break;

}

}

int main()

{

//setlocale(LC\_ALL, "rus");

while (true)

{

system("cls");

menu();

}

system("pause");

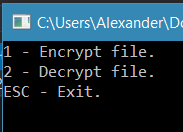
}

**Результат работы.**

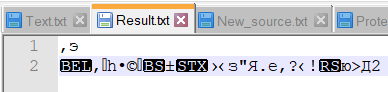
Результат работы программы показан на рисунках 1-5.



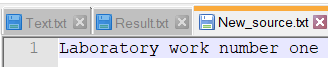
**Рисунок 1 - Исходный файл.**



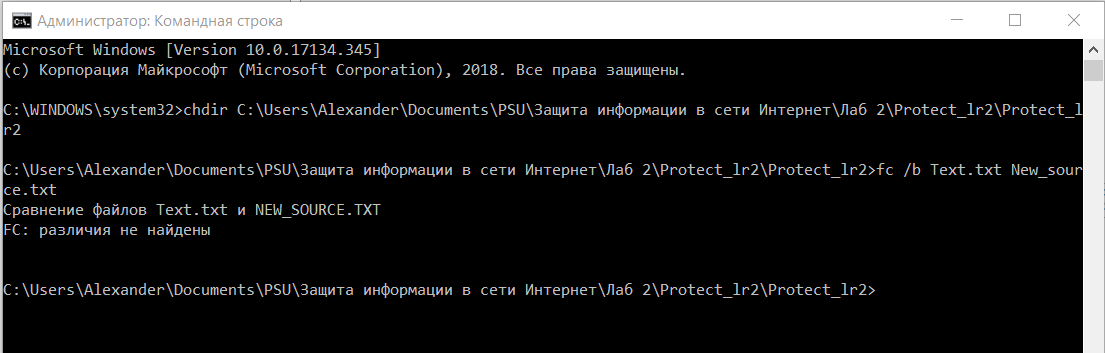
**Рисунок 2 - Меню программы.**



**Рисунок 3 - Результат после шифрования.**



**Рисунок 4 - Результат после дешифрования.**



**Рисунок 5 - Результат проверки файлов.**

**Вывод.**

Научились разрабатывать программы шифрования/дешифрования двоичных файлов с использованием перестановочного шифра.