Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по курсу **«Защита информации в сети Internet»**

по лабораторной работе №2

на тему «**Разработка программы шифрования/дешифрования двоичных файлов с использованием перестановочного шифра**»

Выполнили  
студенты группы 18ВВ1:  
 Кривозубов А. К.

Махров А. А.

Приняли:  
к.т.н., доцент Дубравин А.В.

Пенза 2020

**Цель работы:** Изучение шифрования и расшифровывания произвольного двоичного файла с использованием перестановочного шифра используя, в качестве ключа последовательность, соответствующую номеру варианта.

**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.

|  |  |
| --- | --- |
| Номер варианта | Последовательность перестановки |
| 6 | 24 6 25 31 27 19 2 26 11 3 18 30 7 5 10 16 1 17 28 20 12 0 23 4 29 13 8 14 22 9 21 15 |

**Листинг программы:**

#include <iostream>

#include <fstream>

#include <vector>

#include <conio.h>

using namespace std;

bool filecompare(ifstream &first, ifstream &second)

{

char ch1 = 0, ch2;

while (ch1 != EOF)

{

ch1 = first.get(); ch2 = second.get();

if (ch1 != ch2)

return false;

}

return true;

}

void encryption(int key[]) {

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

ifstream ifs("Text1.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);

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 - key[i]));

if (val != 0) {

encr = encr | (1 << (31 - 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 (d < size + c) {

fout << buffer[3 - i];

}

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();

size\_t block = 1024;

ifs.seekg(size - 1);

ifs.get(minus);

int intminus = minus - '0';

ifs.seekg(0);

ofstream fout("Text2.txt", ios\_base::trunc | ios::binary);

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

char \* buf = new char[block];

ifs.read(buf, block);

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;

}

//---Сравнение---

ifstream first ("Text1.txt", ios::binary);

ifstream second ("Text2.txt", ios::binary);

if (filecompare(first, second)) {

cout << "Файлы одинаковы!" << endl;

}

else {

cout << "Файлы не совпали!" << endl;

}

}

void menu() {

char ch\_key;

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

cout << "1 - Зашифровать файл" << endl;

cout << "2 - Расшифровать файл" << endl;

cout << "ESC - Выйти" << endl;

switch (ch\_key = \_getch())

{

case '1':

cout << "Шифрование файла..." << endl;

encryption(key);

cout << "Файл зашифрован" << endl;

system("pause");

break;

case '2':

cout << "Дешифрование файла..." << endl;

decryption(key);

cout << "Файл расшифрован" << endl;

system("pause");

break;

case 27:

cout << "Exit from program\n";

exit(0);

break;

default:

break;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

while (true) {

system("cls");

menu();

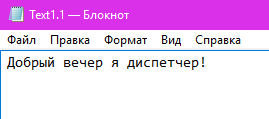
}

system("pause");

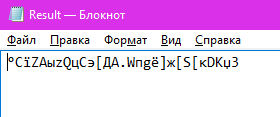
}

**Результат работы программы:**

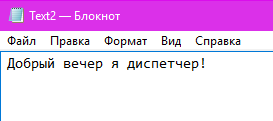
Исходный файл:

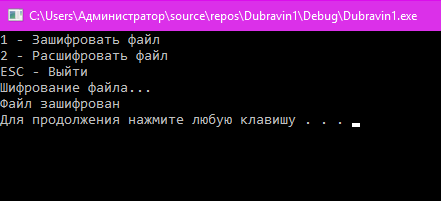


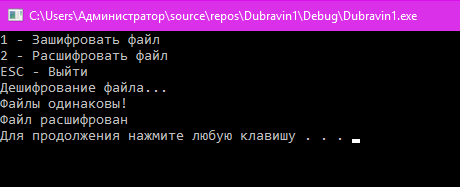
Закодированный файл:

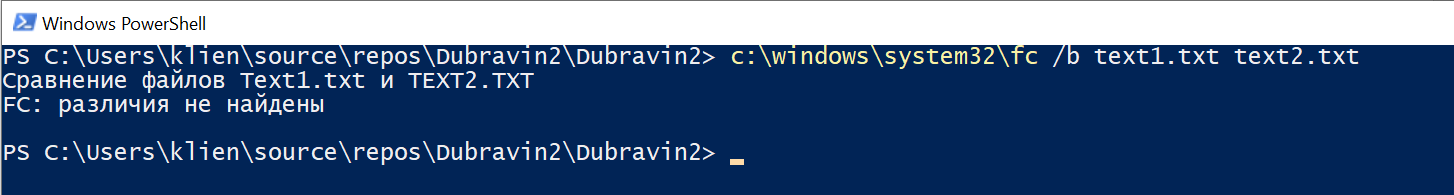


Раскодированный файл:







****

**Вывод:** Мы изучили шифрование и расшифровывание произвольного двоичного файла с использованием перестановочного шифра используя, в качестве ключа последовательность, соответствующую номеру варианта.