Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики–

Филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

(ИАТЭ НИЯУ МИФИ)

Отделение Интеллектуальных кибернетических систем

Направление подготовки: 01.03.02 Прикладная математика и информатика, профиль – Прикладная информатика

УДК 51.37

Утверждаю

Директор ИАТЭ НИЯУ МИФИ

\_\_\_\_\_\_\_\_\_\_\_\_\_Т.А. Осипова

«\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

ЛАБОРАТОРНАЯ РАБОТА № 3

по теме:

криптографическая защита по ключам

Студент гр. М-Б20 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Заграевская

Преподаватель,

к.ф.-м.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Мышев

Обнинск 2021

СОДЕРЖАНИЕ

I. Введение 3

1.1. Постановка цели 3

1.2. Постановка задачи и этапы выполнения работы 3

1.3. Основные понятия и определения 4

II. Основная часть 8

2.1. Листинг программы 8

2.2. Тестирование 11

III. Заключение 13

**Введение**

**1.1. Постановка цели**

Цель работы: разработка и реализация компьютерных технологий помехоустойчивого кодирования информации на основе синтеза традиционных методов избыточного кодирования и способов виртуализации каналов передачи и систем хранения в виде готового программного продукта; приобретение базовых знаний применения методов избыточного кодирования для создания моделей алгоритмов и процедур программных компонент таких технологий; освоение навыков применения теории кодов, исправляющих ошибки для решения практических задач защиты (физической и конфиденциальной) информации в каналах передачи и системах хранения на основе технологий виртуализации; освоение методологии анализа и оценки эффективности и степени надежности помехоустойчивой защиты разрабатываемой технологии.

**1.2. Постановка задач и этапы выполнения работы**

1. Разработка модели алгоритмов и процедур программных компонентов помехоустойчивого кодирования информационного объекта в кодировании каналов передачи и хранения согласно схеме, определенной в задании.
2. Реализация программного интерфейса ввода-вывода файловых структур с ВУ в ОЗУ и обратно с учетом особенностей периферии.
3. Реализация моделей алгоритмов и процедур технологий кодера и декодера на бинарных полях в виде программных компонент.
4. Разработка и реализация программного оконного диалогового интерфейса для ввода исходных данных, записи закодированного информационного объекта на ВУ и чтения в ОЗУ.
5. Отладка программных компонент.
6. Тестирование и контрольные примеры.
7. Расчеты и анализ. Выводы.
8. Прием-сдача программного компонента в диалоговом режиме.

**1.3.** **Основные понятия и определения**

Криптографический ключ представляет собой специальный набор данных, при помощи которого выполняется шифрование и расшифровка информации, отправляемой по сети. Криптоключи используются при определении кодов аутентичности и для проверки электронных цифровых подписей.

Успешность дешифровки будет зависеть от используемого ключа, и, если по какой-либо причине доступ к нему будет утерян, расшифровать данные будет невозможно.

Длина криптографического ключа ‒ объем информации, хранящейся в криптографических ключах, измеряется в битах.

**Кодирование для исправления ошибок: основные положения**

Все коды, исправляющие ошибки, основаны на одной общей идее: для исправления ошибок, которые могут возникнуть в процессе передачи или хранения информации, к ней добавляется некоторая избыточность. По основной схеме, используемой на практике, избыточные символы дописываются за информационными, образуя кодовую последовательность или **кодовое слово(*codeword*)*.*** На рисунке №1 показана геометрическая иллюстрация процедуры формирования **блокового кода(*bloch code*)***.* Такое кодирование называют **систематическим (*systematic*)***.*

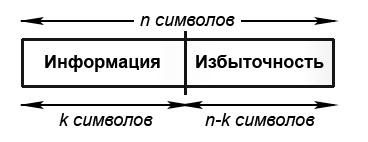


Рис.1. Систематическое блоковое кодирование для исправления ошибок

Это означает, что информационные символы всегда появляются на первых ***k*** позициях кодового слова. Символы на оставшихся **(*n - k*)** позициях являются различными функциями от информационных символов, обеспечивающими тем самым избыточность, необходимую для обнаружения или исправления ошибок. Множество всех кодовых последовательностей называют **кодом, исправляющим ошибки(*error correcting code*).**

**Корректирующая способность кода и его связь с кодовым расстоянием**

Двоичный код ***С***, представляющий собой множество **2*n*** символьных последовательностей длины ***n*** исправляющий ошибки, если передает по информационному каналу не все возможные векторы, может обладать свойством помехоустойчивости. Код ***С*** является подмножеством ***n -*** мерного двоичного пространства ***R*2 *=* {0,1}*n***таких, что его элементы максимально удалены друг от друга. В информационном пространстве ***V*2** расстояние определяется как число позиций, в которых два вектора не совпадают. Пусть ***https://lh3.googleusercontent.com/LDponZrCM5fb2qdva9zu3aeRnq7xYbbEWS6W3gTTVLc71LDgB9LMH3CYIWAY9ryVt6N1TOTORH3sBHOr4PYZ9UW3kIJlu073uHr90LGx8TC3R7Ilth8D6SWBhuxdtTxqZpSxFJc*1*=* (*х*1*,*0*;х*1*,*1*…;х*1*,n-*1)** и ***https://lh3.googleusercontent.com/LDponZrCM5fb2qdva9zu3aeRnq7xYbbEWS6W3gTTVLc71LDgB9LMH3CYIWAY9ryVt6N1TOTORH3sBHOr4PYZ9UW3kIJlu073uHr90LGx8TC3R7Ilth8D6SWBhuxdtTxqZpSxFJc*2 *=* (*х*2*,*0*,х*2*,*1*,…,х*2*,n-*1)**два вектора в ***V*2**, тогда хеммингово расстояние между векторами ***https://lh3.googleusercontent.com/LDponZrCM5fb2qdva9zu3aeRnq7xYbbEWS6W3gTTVLc71LDgB9LMH3CYIWAY9ryVt6N1TOTORH3sBHOr4PYZ9UW3kIJlu073uHr90LGx8TC3R7Ilth8D6SWBhuxdtTxqZpSxFJc*1**и ***https://lh3.googleusercontent.com/LDponZrCM5fb2qdva9zu3aeRnq7xYbbEWS6W3gTTVLc71LDgB9LMH3CYIWAY9ryVt6N1TOTORH3sBHOr4PYZ9UW3kIJlu073uHr90LGx8TC3R7Ilth8D6SWBhuxdtTxqZpSxFJc*2**, которое обозначим, как ***dx*(*x*1*,x*2)*,*** равно

***dx*(*x*1*,x*2) *= |{i:х*1*, i≠x*2*,i,* 0*≤i≤n}***|, (1.1)

где через **|*A*|** обычно обозначают число элементов в множестве ***А***.

Для кода ***С*** минимальное хеммингово расстояние *dmin* определяется, как минимум хеммингова расстояния по всем возможным парам различных кодовых слов:

***dmin = min {dx* (*V*1*, V*2)| *V*1*≠ V*2 *}.*** (1.2)

***V*1*, V*2 *https://lh4.googleusercontent.com/fQ0wHdkIHaIIfh3NRW1S5HiXEKPTu9yfuLPAwnnqQFesXAsigmWiV2kN_luDN69icQ6_eJatat0kQM23tl7MotERSuqmZiPWu-UEMLtCwBLyPysjgaf-fAzszjH7et_QQXG3-EwC***

**Корректирующей способностью(*error correcting capability) t*** кода ***С*** называют наибольший радиус хемминговой сферы ***St*(*V*)**для всех кодовых слов ***Vt ∊ C*** такой, что для любых различных пар ***Vi , Vj*** ∊ **C** соответствующие им хемминговы сферы не пересекаются, т.е.

***https://lh3.googleusercontent.com/3HnqUUwAK6nBcJprqEVTyjx9oH2NKO51-P7OYoV-4BbpaGjTqZtkxuHloOcgIUG8qbe4kufPoun3zvUC5IrnT6-8_oS_UI_nZgmCS6cvfL3iuJIcxCR4znju53Qy7tCakQbv2rYt = max {l| Sl*(*Vi*)*https://lh4.googleusercontent.com/8TZnENx_zlbPtlsvlN-bPBtIMLdSJkgAOgvt_yheW1ZR27wBnMwzUNlC3oH1BNkRugd8kmkdOW7fyx4W0k_gJAhbdg2Ur0HEhcPMMHKd0OxTHvYZsqMuGPve70GdxU9hMhdzteQ Sl*(*Vj*) *= 0, Vi ≠ Vj }*** (1.6)

***Vi, Vj https://lh5.googleusercontent.com/xg595OTCIU0N0JjW6L0BvfPLmBFjPc4i9MbhA3y7ZEJ4jUWOWDrWGJ9M0KjOXT9S5LgpNgDJPN_QLvM7dDm4nMJPLsA29S8unik9F87Z6vq2NeJmEJBCuIkzKsldasXlxayqa2Ql***

Это соответствует более распространенному определению

***t = [*(*dmin -* 1)*/*2*],*** (1.7)

где через ***[Х]*** обозначается целая часть ***Х***, т.е. целое число меньше или равное ***х***.

Отметим, что для определения минимального кодового расстояния произвольного кода ***С*** в соответствии с выражением (1.3) необходимо вычислить все **2*k*(2*k-*1)** расстояний между различными парами кодовых слов. Это практически трудноразрешимая процедура даже для сравнительно коротких кодов.

**Порождающая и проверочная матрицы**

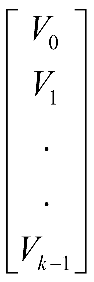
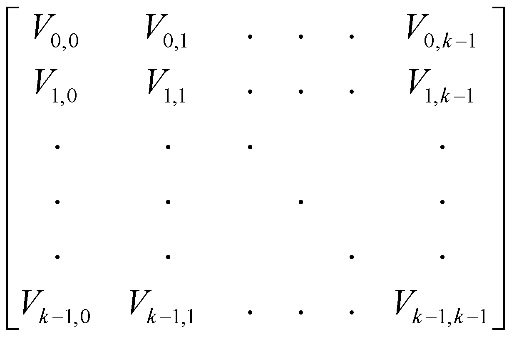
Если ***С*** - двоичный линейный ***(n, k, dmin)*** код, который образует ***k -*** мерное подпространство, имеющее базис **(*V*0*,V*1*,…,Vk-*1)** такой, что любое кодовое слово ***Vhttps://lh5.googleusercontent.com/xg595OTCIU0N0JjW6L0BvfPLmBFjPc4i9MbhA3y7ZEJ4jUWOWDrWGJ9M0KjOXT9S5LgpNgDJPN_QLvM7dDm4nMJPLsA29S8unik9F87Z6vq2NeJmEJBCuIkzKsldasXlxayqa2QC*** может быть выражено через базис:

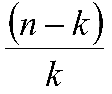
***V = U*0*V*0 *+ U*1*V*1 *+…+ Uk-*1*Vk-*1*,*** (1.8)

где ***Ui https://lh5.googleusercontent.com/xg595OTCIU0N0JjW6L0BvfPLmBFjPc4i9MbhA3y7ZEJ4jUWOWDrWGJ9M0KjOXT9S5LgpNgDJPN_QLvM7dDm4nMJPLsA29S8unik9F87Z6vq2NeJmEJBCuIkzKsldasXlxayqa2Q {*0,1*},* 0 *≤ i ≤ k***

Уравнение (1.8) может быть записано через породающую матрицу ***G*** и векторное сообщение ***U=*(*U*0*,U*1*,…,Uk-*1)**

***V=UG ,*** (1.9)

где G = =****** (1.10)

Так как ***G*** является ***k -*** мерным векторным пространством в ***V2***, то существует ****- **мерное дуальное пространство(*dual space*)** ***С┴***, которое порождается строками матрицы **Н**, называемой проверочной матрицей  (*parity chech* *matrix*), такой что **GHT =0**, где через ***HT*** обозначается транспонированная матрица ***Н***. Следует заметить, что любое кодовое слово ***Vhttps://lh5.googleusercontent.com/xg595OTCIU0N0JjW6L0BvfPLmBFjPc4i9MbhA3y7ZEJ4jUWOWDrWGJ9M0KjOXT9S5LgpNgDJPN_QLvM7dDm4nMJPLsA29S8unik9F87Z6vq2NeJmEJBCuIkzKsldasXlxayqa2QC*** удовлетворяет условию

***VHT =*0**. (1.11)

Уравнение (1.11) является фундаментальным для декодирования линейных кодов.

Линейный ***к***од ***С┴,*** который генерируется матрицей ***Н***, является линейным ***(n, n-h, d┴min)*** кодом, который называется **дуальным коду *С***.

# Основная часть

# 2.1. Листинг программы

#include <iostream>

#include <locale.h>

#include <fstream>

#include <string>

using namespace std;

void EncryptionWithKeys(string text, fstream& fout);

string ToBinaryString(string text);

string ToBinaryString(int n);

string ToNormText(string text);

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Данные берутся из fin.txt, результат записывается в fout.txt" << endl;

ifstream fin("fin.txt");

fstream fout("fout.txt", ios\_base::out | ios::binary | ios\_base::trunc);

string str, text;

getline(fin, str);

text += str;

while (!fin.eof()) {

getline(fin, str);

text += '\n' + str; //Для определения конца строки

}

EncryptionWithKeys(text, fout);

fin.close();

fout.close();

return 0;

}

void EncryptionWithKeys(string text, fstream& fout) {

cout << "Введите ключ" << endl;

string key;

cin >> key;

string keyBin;

keyBin = ToBinaryString(key);

string textBin = ToBinaryString(text);

cout << "Текст: ";

for (int i = 0; i < textBin.size(); i++) {

if (i % 4 == 0) cout << " ";

cout << textBin[i];

}

cout << endl;

cout << "Ключ: ";

for (int i = 0; i < keyBin.size(); i++) {

if (i % 4 == 0) cout << " ";

cout << keyBin[i];

}

cout << endl;

string resBin = "";

for (int i = 0; i < textBin.size();) {

for (int j = 0; j < keyBin.size() && i < textBin.size(); j++, i++) { // конъюнкция или побитовое и

if (textBin[i] == '1' && keyBin[j] == '1')

resBin += "1";

else

resBin += "0";

}

}

cout << "Результат: ";

for (int i = 0; i < resBin.size(); i++) {

if (i % 4 == 0) cout << " ";

cout << resBin[i];

}

}

string ToBinaryString(string text) {

string textBin = "";

int letter;

for (int i = 0; i < text.size(); i++) {

letter = text[i] - '0';

textBin += ToBinaryString(letter);

}

return textBin;

}

string ToBinaryString(int n) {

string res;

if (n == numeric\_limits<int>::digits, '0');

string buffer;

buffer.reserve(numeric\_limits<int>::digits + 1); //+1 для минуса

bool negative = (n < 0);

if (negative)

n = -n;

do {

buffer += char('0' + n % 2);

n = n / 2;

} while (n > 0);

res = string(buffer.crbegin(), buffer.crend());

while (res.size() < 4)

res = "0" + res;

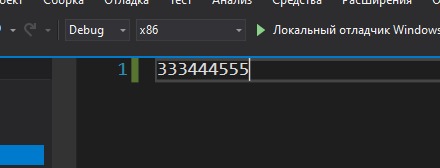
return res;

}

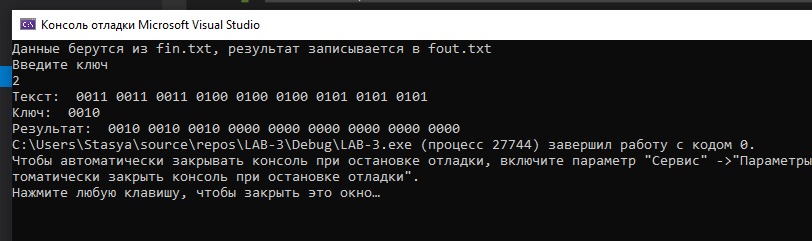
# 2.2. Тестирование

В целях тестирования программы было проведено 3 теста, результаты которых представлены ниже.

Входные данные:



Выходные данные:



Программа отработала верно.

**ЗАКЛЮЧЕНИЕ**

Результатом данной лабораторно работы явилось создание программы на языке программирования С++, реализующей следующие функции:

* ввод-вывод информации, имеющий организацию в виде файловых структур, с ВУ в ОЗУ и обратно, который позволяет определить и описать ее тип, как бинарное поле или строка;
* реализация преобразования информационного объекта в виртуальном адресном пространстве бинарного поля ОЗУ;
* реализация помехоустойчивого кодирования исходного файла с его записью на ВУ индивидуального компьютера или передачей по сети;
* реализация декодирования «запрещенного» объекта и выявление возможных ошибок;
* реализация процедуры оптимизации выбора параметров с целью повышения надежности и эффективности помехоустойчивого кодирования;
* реализация контроль правильности кодирования и декодирования по схеме кода.

Через решение поставленной задачи были изучены и апробированы методы сжатия информации на основе методов кодирования и способов виртуализации в каналах передачи и хранения в виде готового программного продукта, приобретены базовые знания построения моделей алгоритмов и процедур программных компонент таких технологий, сформированы навыки интеллектуального анализа информационных объектов на уровне их «физического и виртуального» представления, оценки эффективности и конкурентоспособности технологии продукта.