Государственное учреждение образования

«Средняя школа №10 г. Солигорска»

Исследовательская работа

**«Защита данных»**

|  |  |
| --- | --- |
|  | Выполнил  Ермаков Илья Вадимович,  учащийся 9 «А» класса  Руководитель  Горавская Виктория Дмитриевна,  учитель физики  223710 г. Солигорск  ул. Октябрьская, 63  тел. 8(0174) 33-00-93  school10\_soligorsk@tut.by |

2024

ОГЛАВЛЕНИЕ

**Введение**……………………....…………………………………………………3

**Глава 1. Теоретическая часть**

* 1. Кодирование информации… ……...…….…………………………….…..4

1.2 Коды и шифры………… ..…………………………………………………..5

**Глава 2. Практическая часть**

2.1 Принцип кодирования……………………………………………………….7

2.2 Создание кодировщика на С++…………………………………………......8

**Заключение**……………………………………………………….……………16

**Список использованных источников**……………………………………....17

**ВВЕДЕНИЕ**

На сегодняшний день информация является главной ценностью современного мира. Благодаря информации осуществляется управление практически всеми сферами жизни современного человека. Информационные ресурсы приносят значительную пользу, однако они могут быть подвержены различным негативным проявлениям, и как следствие информация нуждается в защите. С этой целью осуществляется кодирование информации. Оно бывает разных видов. Например, шифрование или стеганография. Также, кодирование данных обеспечивает компактность и удобство использования информации. Для прочтения закодированных данных применяется операция декодирования. Кодирование информации обеспечивает сохранность, надежность и компактность данных.

Таким образом, выбранная мною тема исследовательской работы, является **актуальной**.

**Гипотеза**: возможно ли самостоятельно создать кодировщик текстовых файлов?

**Объект исследования:** защита данных.

**Предмет исследования:** обеспечение информационной безопасности данных.

**Цель:** создание кодировщика текстовых файлов.

**Задачи:**

1. исследование всевозможной специальной и научно-популярной литературы, а также ознакомление с сетевыми ресурсами;
2. создание кодировщика:

а. разработка алгоритма кодировки;

б. разработка программы кодировки;

в. отладка,

г. сборка.

**Методы исследования:**

1. анализ источников литературы по данной теме;
2. поиск информации в сетевых ресурсах;
3. углубленное изучение языка программирования С++.

**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

* 1. **Кодирование информации**

**Кодирование**– это перевод информации с одного языка на другой (запись в другой системе символов, в другом алфавите). При этом обычно кодированием называют перевод информации с «человеческого» языка на формальный, например, в двоичный код, а **декодированием** – обратный переход. Один символ исходного сообщения может заменяться одним символом нового кода или несколькими символами, а может быть и наоборот – несколько символов исходного сообщения заменяются одним символом в новом коде (так, китайские иероглифы обозначают целые слова и понятия).  Кодирование может бытьравномерноеи неравномерное; при равномерном кодировании все символы кодируются кодами равной длины; при неравномерном кодировании разные символы могут кодироваться кодами разной длины, это затрудняет однозначное декодирование или даже делает его невозможным. [1].

**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

* 1. **Коды и шифры**

Существует много разных систем шифрования. К ним прибегают в военном деле, на дипломатической службе, вообще в тех случаях, когда нужно сохранить в тайне содержание переписки. Шифрование текста используется человечеством с того самого момента, как появилась первая секретная информация, то есть такая, которая должна быть недоступна тем, кому она не предназначена. Опишем несколько методов шифрования текста.

Азбука Морзе является способом знакового кодирования, представление букв алфавита, цифр, знаков препинания и других символов последовательностью сигналов: длинных (тире) и коротких точек). За единицу времени принимается длительность одной точки. Длительность тире равна трём точкам. Пауза между элементами одного знака — одна точка, между знаками в слове — 3 точки, между словами — 7 точек. В отличие от большинства шифров, азбука Морзе используется не для затруднения чтения сообщений, а наоборот, для облегчения их передачи (с помощью телеграфа). Длинные и короткие сигналы посылаются с помощью включения и выключения электрического тока [2].



Рис.1. Азбука Морзе

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом вправо на 3, А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского полководца Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами. Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и всё ещё имеет современное приложение в системе ROT13. Как и все моно алфавитные шифры, шифр Цезаря легко взламывается и не имеет почти никакого применения на практике [3].

**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

* 1. **Коды и шифры**

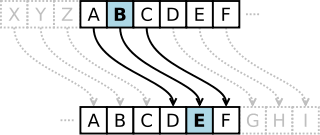


Рис. 2. Шифр Цезаря

В шифре Цезаря каждая буква алфавита сдвигается на несколько позиций. Например, в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Очень долгое время шифр Виженера считался не взламываемым. Чтобы его расшифровать, для начала угадывают длину кодового слова и применяют частотный анализ к каждой n-ной букве послания, где n — предполагаемая длина кодового слова. Если длина была угадана верно, то и сам шифр вскроется с большей или меньшей долей вероятности. Если предполагаемая длина не дает верных результатов, то пробуют другую длину кодового слова, и так далее до конца [4].

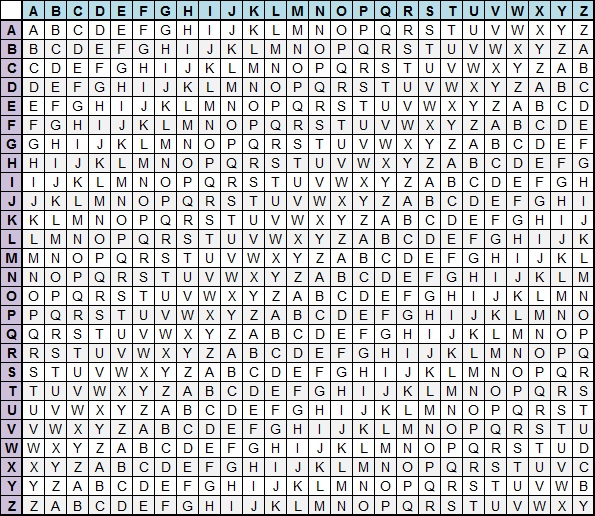


Рис. 3. Шифр Виженера

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.1 Принцип кодирования**

Теперь зная существующие принципы шифрования, можно рассмотреть мой вариант. Конкретной формулы как таковой не существует, потому что при каждом новом кодировании программа генерирует случайную формулу используя массив математических операций. Если упростить то моя программа представляет собой кодовый замок с минимум 10 000 000 вариаций пароля.

Не зная ключа декодировать зашифрованный текст будет максимально проблематично.

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

#include <cmath>

#include <ctime>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

unsigned long long a;

unsigned long long s;

void math1() { s = s \* 2; }

void math2() { s = s \* 3; }

void math3() { s = s \* 13; }

void math4() { s = s \* 5; }

void math5() { s = s \* 8; }

void math6() { s = (s + s) \* 2;}

void math7() { s = (s \* 10); }

void math8() { s = 6 \* s; }

void math9() { s = s \* 4; }

void math10() { s = (s \* 2) + (s \* 3); }

void rev\_math1() { s = s / 2; }

void rev\_math2() { s = s / 3; }

void rev\_math3() { s = s / 13; }

void rev\_math4() { s = s / 5; }

void rev\_math5() { s = s / 8; }

void rev\_math6() { s = s / 4; }

void rev\_math7() { s = (s / 10); }

void rev\_math8() { s = s / 6; }

void rev\_math9() { s = s / 4; }

void rev\_math10() { s = s / 5; }

void (\*functions[])() = {

math1,

math2,

math3,

math4,

math5,

math6,

math7,

math8,

math9,

math10

};

void (\*Rev\_functions[])() = {

rev\_math1,

rev\_math2,

rev\_math3,

rev\_math4,

rev\_math5,

rev\_math6,

rev\_math7,

rev\_math8,

rev\_math9,

rev\_math10

};

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

string chA[] = {

"",

" ",

"'",

"\n",

"A",

"B",

"C",

"D",

"E",

"F",

"G",

"H",

"I",

"J",

"K",

"L",

"M",

"N",

"O",

"P",

"Q",

"R",

"S",

"T",

"U",

"V",

"W",

"X",

"Y",

"Z",

"a",

"b",

"c",

"d",

"e",

"f",

"g",

"h",

"i",

"j",

"k",

"l",

"m",

"n",

"o",

"p",

"q",

"r",

"s",

"t",

"u",

"v",

"w",

"x",

"y",

"z",

"а",

"б",

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

"в",

"г",

"д",

"е",

"ё",

"ж",

"з",

"и",

"й",

"к",

"л",

"м",

"н",

"о",

"п",

"р",

"с",

"т",

"у",

"ф",

"х",

"ц",

"ч",

"ш",

"щ",

"ъ",

"ы",

"ь",

"э",

"ю",

"я",

"А",

"Б",

"В",

"Г",

"Д",

"Е",

"Ё",

"Ж",

"З",

"И",

"Й",

"К",

"Л",

"М",

"Н",

"О",

"П",

"Р",

"С",

"Т",

"У",

"Ф",

"Х",

"Ц",

"Ч",

"Ш",

"Щ",

"Ъ",

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

"Ы",

"Ь",

"Э",

"Ю",

"Я",

"(",

")",

"{",

"}",

"[",

"]",

"/",

":",

";",

"+",

"-",

"\*",

"@",

"!",

"#",

"№",

"?",

"^",

"$",

"~",

"`",

"=",

"<",

">",

"1",

"2",

"3",

"4",

"5",

"6",

"7",

"8",

"9",

"0",

".",

","

};

string chAR[] = {

"1",

"2",

"3",

"4",

"5",

"6",

"7",

"8",

"9",

"0"

};

int sim;

int simbals;

int simbals\_rev;

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

string cey;

string cey\_operation;

string in\_true;

string obj\_in\_true;

string obj\_out\_true;

int timer = 0;

string buffer;

int fun = 0;

string simb\_get;

int main() {

srand(time(0));

int pr1 = rand() % 10;

int pr2 = rand() % 10;

int pr3 = rand() % 10;

int pr4 = rand() % 10;

int pr5 = rand() % 10;

cey += to\_string(pr5) + "x";

cey += to\_string(pr4) + "x";

cey += to\_string(pr3) + "x";

cey += to\_string(pr2) + "x";

cey += to\_string(pr1) + "x";

simbals = 0;

setlocale(LC\_ALL, "ru");

string pathToFile;

cout << "выберите функцию \n" << "кодировка : 1 \n" << "декодировка : 2 \n";

cin >> fun;

cout << "укажите путь к txt файлу \n" << "пример: D:/text.txt \n";

cin >> pathToFile;

fstream fout;

ofstream fin;

fout.open(pathToFile);

char ch2;

char ch;

if (fun == 1) {

while (fout.get(ch)) {

simbals++;

simb\_get.append(1, ch);

if (!fout.is\_open()) {

cout << "файл не найден, тушите свет";

}

else {

for (int i = 0; i < 155; i++) {

if (simb\_get == chA[i]) {

s = i;

cout << a << " " << s << "\n";

(\*functions[pr1])();

cout << s << "\n";

(\*functions[pr2])();

cout << s << "\n";

(\*functions[pr3])();

cout << s << "\n";

(\*functions[pr4])();

cout << s << "\n";

(\*functions[pr5])();

cout << s << "\n \n";

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

obj\_out\_true += to\_string(s) + " ";

}

}

simb\_get.erase();

}

}

}

if (fun == 2) {

cout << "введите ключ" << "\n";

int cey\_1 = 0;

int cey\_2 = 0;

int cey\_3 = 0;

int cey\_4 = 0;

int cey\_5 = 0;

for(int i = 0; i < 5;i++){

getline(cin,cey\_operation, 'x');

cout << cey\_operation;

if (cey\_1 == 0) {

cey\_1 = atoi(cey\_operation.c\_str());

cout << cey\_1 << "\n";

}

else {

if (cey\_2 == 0) {

cey\_2 = atoi(cey\_operation.c\_str());

cout << cey\_2 << "\n";

}

else {

if (cey\_3 == 0) {

cey\_3 = atoi(cey\_operation.c\_str());

cout << cey\_3 << "\n";

}

else {

if (cey\_4 == 0) {

cey\_4 = atoi(cey\_operation.c\_str());

cout << cey\_4 << "\n";

}

else {

if (cey\_5 == 0) {

cey\_5 = atoi(cey\_operation.c\_str());

cout << cey\_5 << "\n";

}

}

}

}

}

}

int timer\_2 = 0;

int sim\_rev = 1;

while (fout.get(ch2)) {

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

simbals++;

if (!fout.is\_open()) {

cout << "файл не найден, тушите свет";

}

else {

int codOutput;

simb\_get += ch2;

if (simb\_get != " ") {

buffer += simb\_get;

simb\_get = "";

}

else {

atoi(buffer.c\_str());

s = stoi(buffer);

cout << cey\_operation;

(\*Rev\_functions[cey\_5])();

cout << s << "\n";

(\*Rev\_functions[cey\_4])();

cout << s << "\n";

(\*Rev\_functions[cey\_3])();

cout << s << "\n";

(\*Rev\_functions[cey\_2])();

cout << s << "\n";

(\*Rev\_functions[cey\_1])();

cout << s << "\n \n";

obj\_out\_true += chA[s];

buffer = "";

}

}

}

}

fout.close();

fin.open(pathToFile);

fin << obj\_out\_true;

cout << "\n" << obj\_out\_true;

cout << "\n" << "Ermakov S.W." << "\n";

if (simbals != 0) {

if (fun == 1) {

cout << "\n" << "кодирование завершено!" << "\n";

cout << "ваш ключ: " << cey << endl;

}

else {

cout << "\n" << "декодирование завершено!" << endl;

}

}

else {

**ГЛАВА 2. ДЕМОНСТРАЦИЯ ПРОЕКТА**

**2.2 Создание кодировщика на С++**

cout << "\n" << "операция не удалась, попробуйте ещё раз" << "\n" << endl;

cey = "";

main();

}

fin.close();

system("pause");

}

**ЗАКЛЮЧЕНИЕ**

Вся жизнь человека, так или иначе, связана с получением, накоплением и обработкой информации. Чтобы человек ни делал: читает ли он книгу, смотрит ли он телевизор, разговаривает, он постоянно и непрерывно получает и обрабатывает информацию. Для любой операции над информацией, даже такой простой, как сохранение, она должна быть как-то представлена записана и зафиксирована. Этот процесс имеет специальное название – кодирование информации. Современному человеку невозможно обойтись без кодирования информации. Чтобы мы понимали все из того, что слышим, видим, чувствуем необходимо знать различные способы кодирования информации.

Таким образом, в результате анализа научно-методической литературы и ознакомления с сетевыми ресурсами, был разработан и создан кодировщик текстовых документов.

Гипотеза оказалась правдивой.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Кодирование информации [Электронный ресурс] / Кодирование информации. – Режим доступа: <https://foxford.ru/wiki/informatika/kodirovanie-informatsii4>. – Дата доступа: 31.08.2023.

2. Как называется азбука, представляющая собой способ знакового кодирования с помощью сигналов: тире и точек? [Электронный ресурс] / Азбука Морзе. – Режим доступа: <https://ru.quizzclub.com/games/welcome/kak-nazyvaetsya-azbuka-predstavlyayushhaya-soboj-sposob-znakovogo-kodirovaniya-s-pomoshhyu-signalov-tire-i-tochek/answer/1977029/>. – Дата доступа: 01.09.2023.

3. Цифр Цезаря [Электронный ресурс] / Цифр Цезаря. – Режим доступа: <https://kmb.cybber.ru/crypto/caezar_cipher/main.html>. – Дата доступа: 08.09.2023.

4. Шифр Цезаря и Виженера C# [Электронный ресурс] / Шифр Цезаря. – Режим доступа:<https://studlearn.com/works/details/shifr-tsezarya-i-vizhenera-c-136>. – Дата доступа: 08.09.2023.