Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

Отчёт

к лабораторной работе

на тему

**ШИФР ЦЕЗАРЯ. ШИФР ВИЖЕНЕРА**

Студент: гр.153504

Ключинский В. Д.

Проверил: Лещенко Е. А.

Минск 2024

СОДЕРЖАНИЕ

[**1 ПОСТАНОВКА ЗАДАЧИ** 3](#_Toc157538333)

[**2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ** 4](#_Toc157538334)

[**3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ** 7](#_Toc157538335)

[**ВЫВОД** 10](#_Toc157538336)

# **1 ПОСТАНОВКА ЗАДАЧИ**

Целью выполнения лабораторной работы является изучение теоретических сведений по алгоритмам шифрования Цезаря и Виженера, реализация программного средства, читающие данные из файла и шифрующие(дешифрующие) их при помощи шифра Цезаря (шифра сдвига, кода Цезаря) и шифра Виженера.

# **2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ**

**Шифр Цезаря**

Шифр Цезаря, также известный, как шифр сдвига, код Цезаря или сдвиг Цезаря – один из самых простых и наиболее широко известных методов шифрования.

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 4 А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского императора 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 все ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.



Рисунок 1. Шифр Цезаря

***Математическая модель***

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

y=(x+k)\ \mod\ n

x=(y-k+n)\ \mod\ n,

где ~x – символ открытого текста, ~y – символ шифрованного текста, ~n – мощность алфавита, а ~k – ключ.

С точки зрения математики шифр Цезаря является частным случаем аффинного шифра.

***Пример***

Шифрование с использованием ключа k = 3. Буква «Е» «сдвигается» на три буквы вперёд и становится буквой «З». Твёрдый знак, перемещённый на три буквы вперёд, становится буквой «Э», и так далее:

Исходный алфавит:

***АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ***

Шифрованный:

***ГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБ***

Оригинальный текст:

*Съешь же ещё этих мягких французских булок, да выпей чаю.*

Шифрованный текст получается путём замены каждой буквы оригинального текста соответствующей буквой шифрованного алфавита:

*Фэзыя йз зьи ахлш пвёнлш чугрщцкфнлш дцосн, жг еютзм ъгб.*

**Шифр Виженера**

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. Например, предположим, что исходный текст имеет такой вид: ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: LEMONLEMONLE

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN; Ключ: LEMONLEMONLE; Зашифрованный текст: LXFOPVEFRNHR.

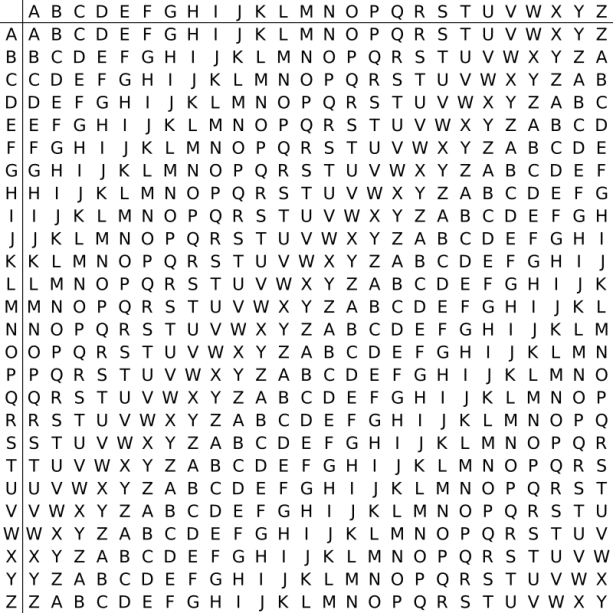


Рисунок 2 Квадрат Виженера

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если n — количество букв в алфавите, m j — буквы открытого текста, k j — буквы ключа, то шифрование Виженера можно записать следующим образом:c j = m j + k j ( mod n )

И расшифровывание:

m j = c j − k j ( mod n )

В компьютере такая операция соответствует сложению кодов ASCII символов сообщения и ключа по некоторому модулю. Кажется, что если таблица будет более сложной, чем циклическое смещение строк, то шифр станет надежнее. Это действительно так, если ее менять чаще, например, от слова, к слову. Но составление таких таблиц, представляющих собой латинские квадраты, где любая буква встречается в строке или столбце один раз, трудоемко и его стоит делать лишь на ЭВМ. Для ручного же многоалфавитного шифра полагаются лишь на длину и сложность ключа, используя приведенную таблицу, которую можно не держать в тайне, а это упрощает шифрование и расшифровывание.

# **3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**

В ходе выполнения лабораторной работы была разработана программа, читающая данные из файла и шифрующие при помощи шифра Цезаря и Виженера соответственно. Блок-схема алгоритма для шифра Цезаря представлена ниже на (Рисунок 1).



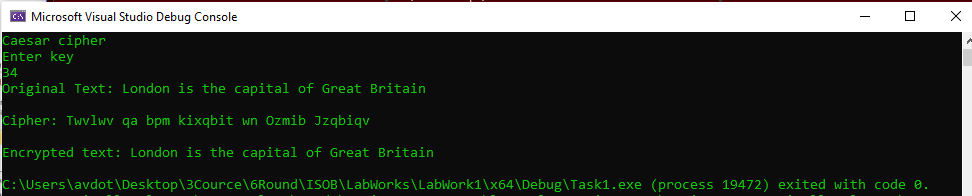
Рисунок 1 – Блок-схема алгоритма для шифра Цезаря

Блок-схема алгоритма для шифра Виженера представлена ниже на (Рисунок 2).



Рисунок 2 – Блок-схема алгоритма для шифра Виженера

Входные данные, записанные в файл для шифрования при помощи шифра Цезаря представлены на (Рисунок 3).



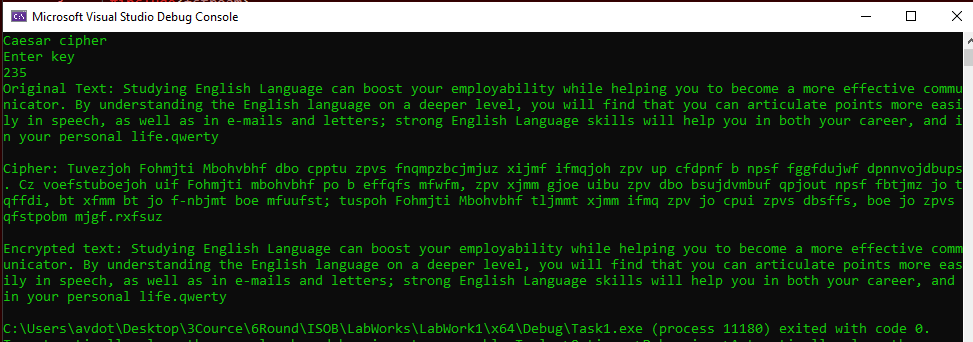
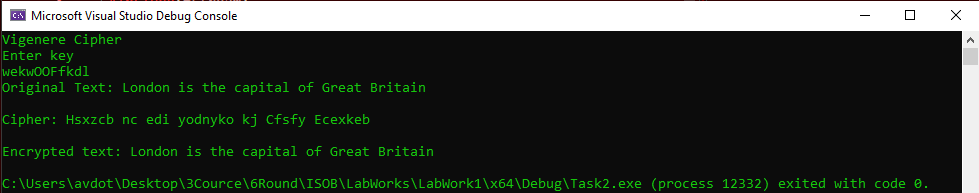


Рисунок 3 – результат выполнения шифра Цезаря



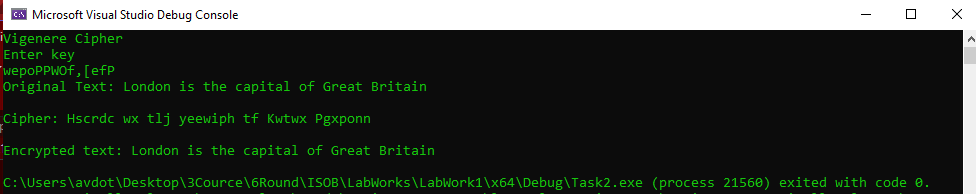


Рисунок 4 – результат выполнения шифра Виженера

# **ВЫВОД**

В ходе выполнения данной лабораторной работы были изучены теоретические сведения по шифрованию, на примере шифра Цезаря и шифра Виженера, которые позволили реализовать программное средство шифрования и дешифрования текстовых файлов.

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Листинг кода**

**Task1.cpp**

#include <iostream>

#include<string>

#include<fstream>

#include<Windows.h>

using namespace std;

const int letter\_count = 26, first\_lower = 97, first\_upper = 65;

const char\* filename = "C:\\Users\\avdot\\Desktop\\3Cource\\6Round\\ISOB\\LabWorks\\LabWork1\\orig\_text.txt";

string encrypt\_text(string& text, int key) {

string ans = "";

int size = text.size();

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

char& sym = text[i];

if (isalpha(sym)) {

if (isupper(sym)) {

ans += (int(sym) + key - first\_upper) % letter\_count + first\_upper;

}

else {

ans += (int(sym) + key - first\_lower) % letter\_count + first\_lower;

}

}

else {

ans += sym;

}

}

return ans;

}

string decrypt\_text(string& ciphered\_text, int key) {

string decrypted\_text = "";

for (char& now : ciphered\_text) {

if (isalpha(now)) {

int shift = int(now) - key;

if (isupper(now)) {

if (shift < int('A')) {

shift += letter\_count;

}

}

else {

if (shift < int('a')) {

shift += letter\_count;

}

}

decrypted\_text += char(shift);

}

else {

decrypted\_text += now;

}

}

return decrypted\_text;

}

string read\_from\_file(string&& filename) {

ifstream inputFile(filename);

string content="";

string str;

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

std::cerr << "Failed to read" << '\n';

return "";

}

while (!inputFile.eof()) {

getline(inputFile, str);

content+=str;

}

inputFile.close();

return content;

}

int main()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 10);

string original\_text = read\_from\_file(filename);

int key;

cout << "Enter key\n";

cin >> key;

if (key < 0) {

key = -key;

}

key %= 26;

cout << "Original Text: " << original\_text << "\n\n";

string cipher = encrypt\_text(original\_text, key);

cout <<"Cipher: "<< cipher << "\n\n";

string decrypted = decrypt\_text(cipher, key);

cout << "Encrypted text: "<<decrypted << '\n';

return 0;

}

**Task2.cpp**

#include <iostream>

#include<string>

#include<fstream>

#include<iomanip>

#include<Windows.h>

using namespace std;

const int letter\_count = 26, first\_lower = 97, first\_upper = 65;

const char\* filename = "C:\\Users\\avdot\\Desktop\\3Cource\\6Round\\ISOB\\LabWorks\\LabWork1\\orig\_text.txt";

string encrypt\_text(string& text, string& key\_word)

{

string cipher = "";

int text\_size = text.size(), key\_word\_size = key\_word.size();

for (int i = 0; i < text\_size; ++i) {

char& sym = text[i];

int key\_word\_index = i % key\_word\_size;

int shift = 0;

if (65 <= key\_word[key\_word\_index] && key\_word[key\_word\_index] <= 90) {

shift = int(key\_word[key\_word\_index]) - 65;

}

else if (97 <= key\_word[key\_word\_index] && key\_word[key\_word\_index] <= 122) {

shift = int(key\_word[key\_word\_index]) - 97;

}

if (isalpha(sym)) {

if (isupper(sym)) {

cipher += char(int(sym)-65+shift)%letter\_count + 65;

}

else {

cipher += char(int(sym)-97 + shift) % letter\_count + 97;

}

}

else {

cipher += sym;

}

}

return cipher;

}

string decrypt\_text(string& cipher, string& key\_word)

{

string decrypt\_text= "";

int cipher\_size = cipher.size(), key\_word\_size = key\_word.size();

for (int i = 0; i < cipher\_size; ++i) {

char& sym = cipher[i];

int key\_word\_index = i % key\_word\_size;

int shift = 0;

if (65 <= key\_word[key\_word\_index] && key\_word[key\_word\_index] <= 90) {

shift = int(key\_word[key\_word\_index]) - 65;

}

else if (97 <= key\_word[key\_word\_index] && key\_word[key\_word\_index] <= 122) {

shift = int(key\_word[key\_word\_index]) - 97;

}

if (isalpha(sym)) {

if (isupper(sym)) {

decrypt\_text += char(int(sym) - 65 + 26 - shift) % letter\_count + 65;

}

else {

decrypt\_text += char(int(sym) - 97 +26- shift) % letter\_count + 97;

}

}

else {

decrypt\_text += sym;

}

}

return decrypt\_text;

}

string read\_from\_file(string&& filename) {

ifstream inputFile(filename);

string content = "";

string str;

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

std::cerr << "Failed to read" << '\n';

return "";

}

while (!inputFile.eof()) {

getline(inputFile, str);

content += str;

}

inputFile.close();

return content;

}

int main()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 10);

cout << setw(10) << "Vigenere Cipher" << '\n';

string original\_text = read\_from\_file(filename);

string key\_word = "QWefeo";

cout << "Enter key\n";

getline(cin, key\_word);

cout << "Original Text: " << original\_text << "\n\n";

string cipher = encrypt\_text(original\_text, key\_word);

cout << "Cipher: " << cipher << "\n\n";

string decrypted = decrypt\_text(cipher, key\_word);

cout << "Encrypted text: " << decrypted << '\n';

return 0;

}