Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.01 – «Информатика и вычислительная техника»

Дисциплина: «Безопасность и защита информации»

Профиль: «Компьютерные системы»

Семестр 6

ОТЧЕТ

по лабораторной работе №1

Тема: «Шифры перестановки и замены»

Выполнил: студент группы КС-21-1б

Чирков А.В. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

**ЦЕЛЬ РАБОТЫ**

Получить практические навыки по применению шифров перестановки и шифров простой замены.

**ЗАДАНИЕ**

Реализовать шифрование текстового сообщения, используя шифр Гронсфельда.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Шифр сложной замены, называемый шифром Гронсфельда, представляет собой модификацию шифра Цезаря числовым ключом. Для этого под буквами исходного сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. Шифртекст получают примерно, как в шифре Цезаря, но отсчитывают по алфавиту не третью букву (как это делается в шифре Цезаря), а выбирают ту букву, которая смещена по алфавиту на соответствующую цифру ключа. Например, применяя в качестве ключа группу из четырех начальных цифр числа e (основания натуральных логарифмов), а именно 2718, получаем для исходного сообщения ВОСТОЧНЫЙ ЭКСПРЕСС следующий шифртекст:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение |  | В | О | С | Т | О | Ч | Н | Ы | Й |  | Э | К | С | П | Р | Е | С | С |
| Ключ |  | 2 | 7 | 1 | 8 | 2 | 7 | 1 | 8 | 2 |  | 7 | 1 | 8 | 2 | 7 | 1 | 8 | 2 |
| Шифртекст |  | Д | Х | Т | Ь | Р | Ю | О | Г | Л |  | Д | Л | Щ | С | Ч | Ж | Щ | У |

Чтобы зашифровать первую букву сообщения В, используя первую цифру ключа 2, нужно отсчитать вторую по порядку букву от В в алфавите

|  |  |  |
| --- | --- | --- |
| В | Г | Д |
|  | 1 | 2 |

получается первая буква шифр-текста Д.

Следует отметить, что шифр Гронсфельда вскрывается относительно легко, если учесть, что в числовом ключе каждая цифра имеет только десять значений, а значит, имеется лишь десять вариантов прочтения каждой буквы шифртекста. С другой стороны, шифр Гронсфельда допускает дальнейшие модификации, улучшающие его стойкость, в частности двойное шифрование разными числовыми ключами.

Шифр Гронсфельда представляет собой, по существу, частный случай системы шифрования Вижинера.

**ХОД РАБОТЫ**

При запуске программы, на экране появляется окно с полями ввода шифруемого сообщения (Enter text), ключа (Enter key), кнопками шифровки (Encrypt) и выхода из программы (Exit), а также полем вывода зашифрованного сообщения (Encrypted Result) (рисунок 1).

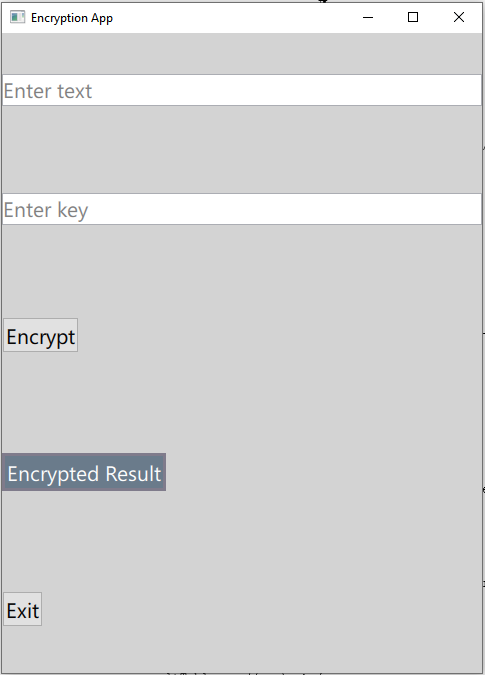


Рисунок 1 – Окно программы при запуске.

Пример работы программы представлен на рисунке 2.

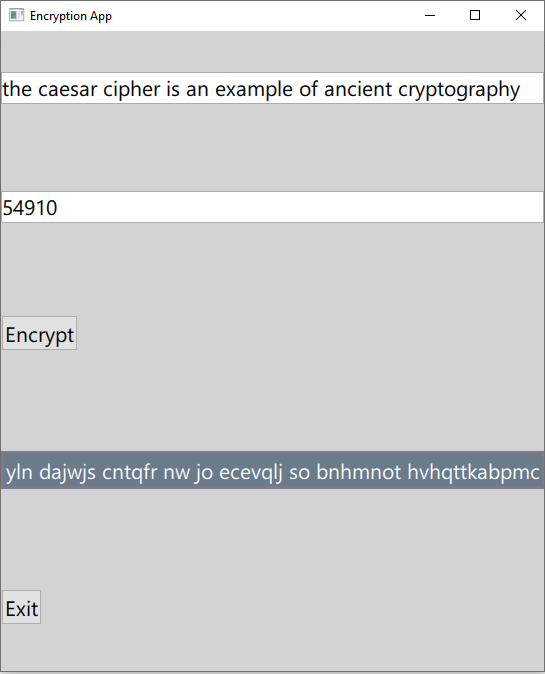


Рисунок 2 – Пример работы программы.

В связи с ограничениями работы с кириллическими символами в базовой версии фреймворка QT, была реализована шифрация сообщений состоящих только из строчных латинских букв. При вводе других символов алгоритм их не обрабатывает (таблица 1).

Таблица 1 – шифр Гронсфельда

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение | t | h | e |  | c | a | e | s | a | r |  | c | i | p | h | e | r |
| Ключ | 5 | 4 | 9 |  | 1 | 0 | 5 | 4 | 9 | 1 |  | 0 | 5 | 4 | 9 | 1 | 0 |
| Шифртекст | y | l | n |  | d | a | j | w | j | s |  | c | n | t | q | f | r |

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

main.cpp

#include <QGuiApplication>

#include <QQmlApplicationEngine>

#include <QQmlContext>

#include "MyCppObject.h"

int main(int argc, char\* argv[]) {

QCoreApplication::setAttribute(Qt::AA\_EnableHighDpiScaling);

QGuiApplication app(argc, argv);

QQmlApplicationEngine engine;

MyCppObject myCppObject;

engine.rootContext()->setContextProperty("myCppObject", &myCppObject);

const QUrl url(QStringLiteral("main.qml"));

QObject::connect(&engine, &QQmlApplicationEngine::objectCreated,

&app, [url](QObject\* obj, const QUrl& objUrl) {

if (!obj && url == objUrl)

QCoreApplication::exit(-1);

}, Qt::QueuedConnection);

engine.load(url);

return app.exec();

}

main.qml

import QtQuick 2.15

import QtQuick.Controls 2.15

import QtQuick.Layouts 1.15

ApplicationWindow {

visible: true

width: 480

height: 640

color: "lightgray"

title: qsTr("Encryption App")

function encryptGronsfeld(text, key)

{

var result = "";

var keyIndex = 0;

for (var i = 0; i < text.length; ++i)

{

var curChar = text[i];

if (!((curChar >= "a" && curChar <= "z")))

{

result += curChar;

continue;

}

var keyDigit = parseInt(key[keyIndex]); //parseInt = str to int

var encryptedCharCode = (curChar.charCodeAt(0) + keyDigit); //curChar.charCodeAt(0) - char код 1 символа строки curChar

if (encryptedCharCode > "z".charCodeAt(0))

{

encryptedCharCode -= 26; // избегаем выпадения из алфавита

}

var encryptedChar = String.fromCharCode(encryptedCharCode);

result += encryptedChar;

keyIndex = (keyIndex + 1) % key.length; // переход к след. цифре

}

return result;

}

ColumnLayout

{

anchors.fill: parent

TextField

{

id: inputField

placeholderText: qsTr("Enter text")

font.pointSize: 16

Layout.preferredWidth: parent.width / 2

Layout.fillWidth: true

}

TextField

{

id: keyField

placeholderText: qsTr("Enter key")

font.pointSize: 16

Layout.preferredWidth: parent.width / 2

Layout.fillWidth: true

}

Button

{

text: "Encrypt"

font.pointSize: 16

onClicked: { encryptedText.text = encryptGronsfeld(inputField.text, keyField.text); }

}

Rectangle

{

width: encryptedText.width + 10

height: encryptedText.height + 10

color: "#6A7B8B"

border.color: "#7F7B8B"

border.width: 3

Text

{

id: encryptedText

text: "Encrypted Result"

font.pointSize: 16

color: "white"

anchors.centerIn: parent

}

}

Button

{

text: "Exit"

font.pointSize: 16

onClicked: Qt.quit()

}

}

}