Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

АЛГОРИТМЫ ДОНАУЧНОЙ КРИПТОГРАФИИ

Отчет по лабораторной работе №1

По дисциплине

«Информационная безопасность»

Студент гр. 431-3

\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Бекиш

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель:

Ассистент кафедры АСУ

\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.В. Яблонский

(подпись)

Томск 2024

Оглавление

[1 Цель работы 3](#_Toc177212477)

[2 Задание на лабораторную работу 4](#_Toc177212478)

[3 Описание алгоритма шифрования 5](#_Toc177212479)

[4 Листинг программы 6](#_Toc177212480)

[5 Примеры работы программы 12](#_Toc177212481)

[6 Вывод 13](#_Toc177212482)

# Цель работы

Познакомиться и научиться работать с алгоритмами донаучной криптографии.

# Задание на лабораторную работу

Напишите программу, позволяющую зашифровать и расшифровать сообщения с использованием алгоритма Атбаш. Правило шифрования заключается в замене i-й буквы алфавита (i = 1,..,n) буквой с номером n - i + 1, где n - число букв алфавита. Входные и выходные данные запишите в файл типа .txt.

# Описание алгоритма шифрования

Атбаш – простой шифр постановки. Правило шифрование состоит в замене i-й буквы алфавита с буквой, которая соответствует номеру n – i + 1 алфавита, где n – мощность алфавита. Данный алгоритм можно увидеть на рисунке 3.1.

Рисунок 3.1 – Алгоритм шифрования

# Листинг программы

Листинг файла ciphers.h

#pragma once

#include <string>

#include <iostream>

#include <map>

#include <fstream>

class Ciphers {

protected:

std::string \_alphabet;

int length\_alphabet;

int \_shift;

std::string message;

std::string cipher\_message;

std::string originall\_message;

int length\_message;

virtual std::string encryption() = 0;

virtual std::string decryption() = 0;

public:

virtual std::string get\_alphabet() const = 0;

virtual int get\_length\_alphabet() const = 0;

virtual std::string get\_text\_from\_file(const std::string &name\_file) const = 0;

virtual std::string get\_message() const = 0;

virtual std::string get\_cipher\_message() const = 0;

virtual std::string get\_originall\_message() const = 0;

virtual int get\_length\_message() const = 0;

virtual void print\_hash\_table() = 0;

friend std::ostream& operator<<(std::ostream& output, const Ciphers& cipher) {

return output << "Испольуемый алфавит: " << cipher.\_alphabet << \

"\n\nНачальное сообщение: " << cipher.message << \

"\nЗашифрованное сообщение: " << cipher.cipher\_message << \

"\nРасшифрованное сообщение: " << cipher.originall\_message << '\n';

}

};

Листинг файла atbash\_cipher.cpp

#include "ciphers.h"

class Abash\_Cipher : virtual public Ciphers {

private:

std::map<char, char> hash\_table\_alphabet;

public:

Abash\_Cipher(const std::string & alphabet, std::string name\_file) {

\_alphabet = alphabet;

length\_alphabet = \_alphabet.length();

hash\_table\_alphabet = get\_hash\_table\_alphabet();

message = get\_text\_from\_file(name\_file);

length\_message = message.length();

cipher\_message = encryption();

originall\_message = decryption();

std::cout << "\t\t\t\tШифр Абаша\n" << std::endl;

}

std::string get\_alphabet() const override {

return \_alphabet;

}

int get\_length\_alphabet() const override {

return length\_alphabet;

}

std::map<char, char> get\_hash\_table\_alphabet() {

for (int i = 0; i < length\_alphabet; i ++) {

hash\_table\_alphabet.emplace(\_alphabet[i], \_alphabet[length\_alphabet - i - 1]);

}

return hash\_table\_alphabet;

}

void print\_hash\_table() override {

if (hash\_table\_alphabet.empty()) {

std::cout << "Hash table is empty." << std::endl;

return;

}

std::cout << "\t Словарь зашифрованных символов\n" << std::endl;

for (const auto &[key, value] : hash\_table\_alphabet) {

std::cout << "Первоначальная буква: " << key << " -> Зашифрованная буква: " << value << std::endl;

}

}

std::string encryption() override {

for (int i = 0; i < length\_message; i ++) {

if (isalpha(message[i])) {

if (isupper(message[i])) {

cipher\_message += hash\_table\_alphabet[message[i]];

}

else {

cipher\_message += std::tolower(hash\_table\_alphabet[std::toupper(message[i])]);

}

}

else {

cipher\_message += message[i];

}

}

return cipher\_message;

}

std::string decryption() override {

for (int i = 0; i < length\_message; i ++) {

if (isalpha(cipher\_message[i])) {

if (isupper(cipher\_message[i])) {

originall\_message += hash\_table\_alphabet[cipher\_message[i]];

}

else {

originall\_message += std::tolower(hash\_table\_alphabet[std::toupper(cipher\_message[i])]);

}

}

else {

originall\_message += cipher\_message[i];

}

}

return originall\_message;

}

std::string get\_text\_from\_file(const std::string &name\_file) const override {

std::ifstream in(name\_file);

std::string text = "", tmp;

if (in.is\_open()) {

while (std::getline(in,tmp)) {

text += tmp;

}

return text;

}

in.close();

return "Файл закрыт!!!";

}

std::string get\_message() const override {

return message;

}

std::string get\_cipher\_message() const override {

return cipher\_message;

}

std::string get\_originall\_message() const override {

return originall\_message;

}

int get\_length\_message() const override {

return length\_message;

}

};

Листинг main.cpp

#include <iostream>

#include <string>

#include "Chipers/abash\_cipher.cpp"

int main() {

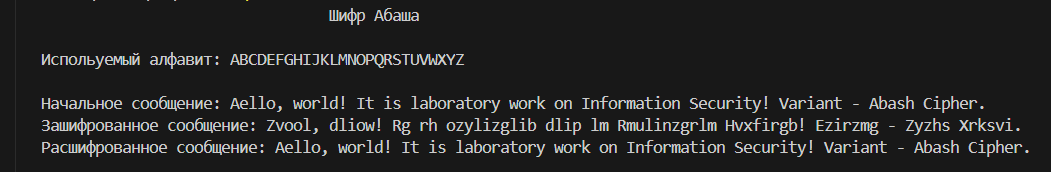
Abash\_Cipher Abash("ABCDEFGHIJKLMNOPQRSTUVWXYZ", "file.txt");

std::cout << Abash << std::endl;

return 0;

}

# Примеры работы программы

Далее подадим на вход в файл .txt два сообщения для шифрования и расшифрования. Результат работы программы можно увидеть на рисунках 5.1 – 5.2.

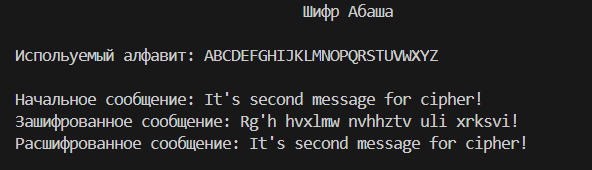
Рисунок 5.1 – Результат первого сообщения

Рисунок 5.2 – Результат второго сообщения

# Вывод

В результате выполнения лабораторной работы я познакомился и научился работать с алгоритмом донаучной криптографии на примере шифра Атбаш, а также выполнил задание в соответствие с заданным вариантом на языке C++.