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

Познакомиться и научиться работать с алгоритмами донаучной криптографии.

2 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.

3 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

4 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;
}
```

5 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.

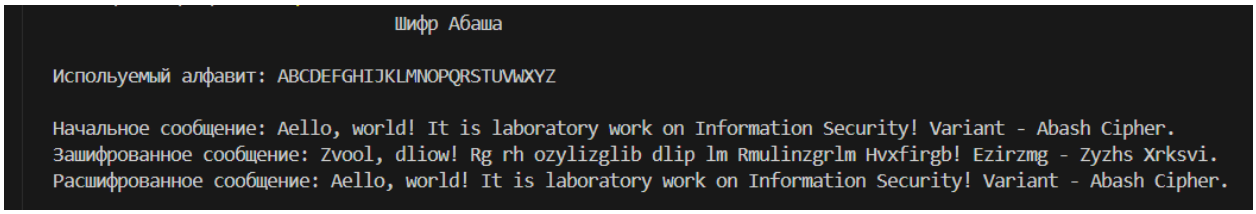

Рисунок 5.1 – Результат первого сообщения

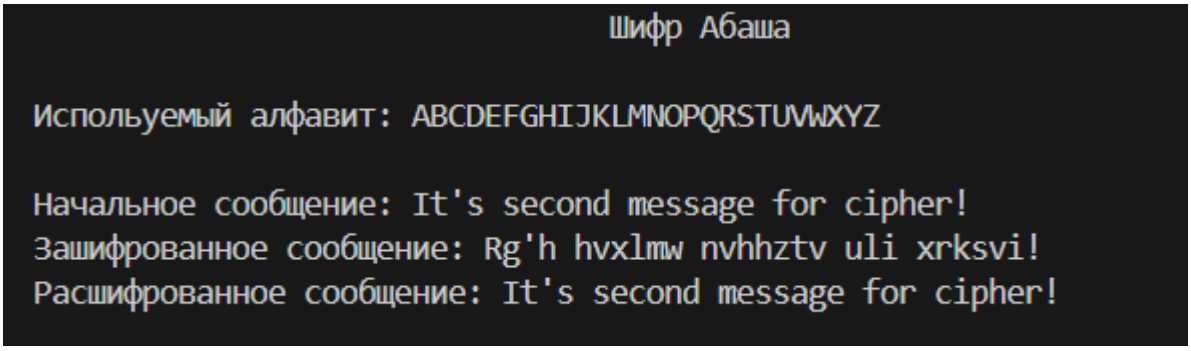

Рисунок 5.2 – Результат второго сообщения

6 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++.