**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Технологии и методы программирования»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

**Выполнили:**

Ахраров Али, студент группы номер N3350

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Ищенко Алексей Петрович

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Содержание

[Введение 3](#_Toc185718485)

[1 Краткое описание алгоритмов шифрования 4](#_Toc185718486)

[1.1 Шифровка последовательностей нулей и единиц (ШПНиЕ) 4](#_Toc185718487)

[1.2 Шифр Виженера для русских букв 4](#_Toc185718488)

[2 Определение ограничений на решаемую задачу 5](#_Toc185718489)

[3 Реализация алгоритмов в виде программного кода 6](#_Toc185718490)

[3.1 Шифровка последовательностей нулей и единиц (ШПНиЕ) 6](#_Toc185718491)

[3.2 Шифр Виженера для русских букв 7](#_Toc185718492)

[4 Реализация и описание графического интерфейса 10](#_Toc185718493)

Введение

Задание - Написать программу, реализующую два исторических примера алгоритмов шифрования. Продумать интерфейс, руководство пользователя и описание работы алгоритмов (их историей и криптоустойчивостью), для демонстрации алгоритмов пользователю со встроенными примерами текстов (шифруем и дешифруем), а также с возможностью ввода произвольного текста с его шифровкой дешифровкой.

* + 1. Необходимо реализовать следующие этапы для каждого своего алгоритма:
    2. Краткое описание алгоритма шифрования.
    3. Определение ограничений на решаемую задачу.
    4. Реализация данного алгоритма в виде программного кода.
    5. Реализация и описание графического интерфейса.
    6. Оформление отчета (включает описание выше и нижеизложенных пунктов со скриншотами работы программы и результатов)

Вариант 2

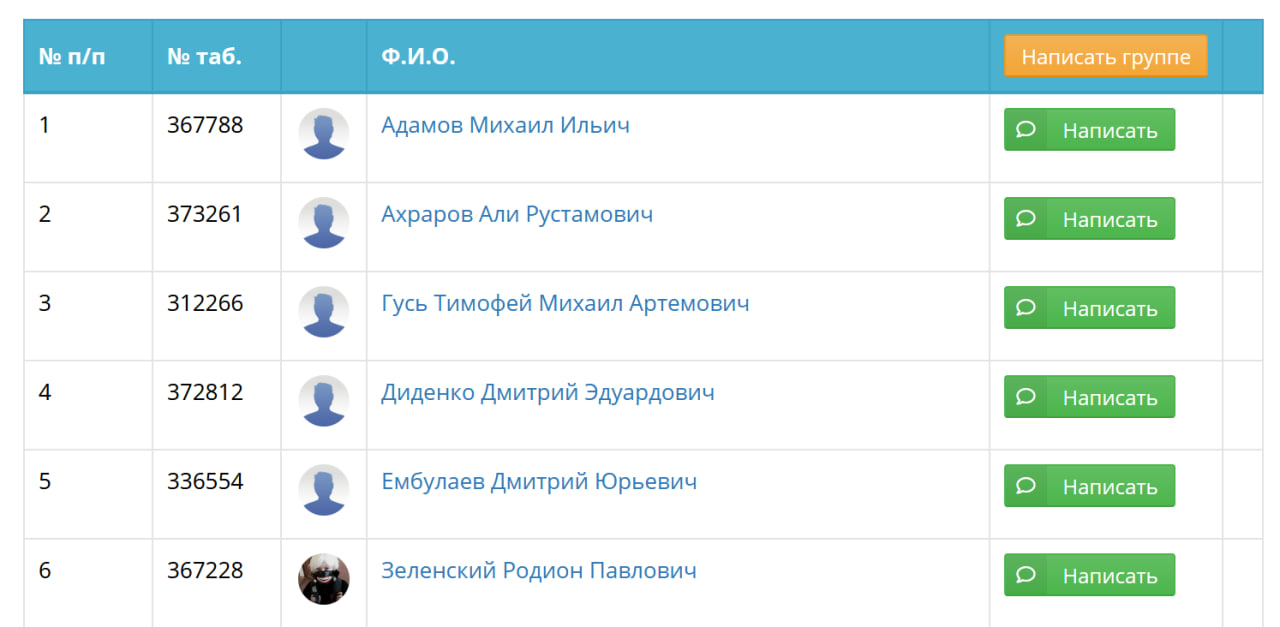


Рис.1 – Скриншот группы в ИСУ

# Краткое описание алгоритмов шифрования

### 1.1 Шифровка последовательностей нулей и единиц (ШПНиЕ)

**Описание алгоритма:** Шифровка последовательностей нулей и единиц представляет собой метод преобразования текста в бинарный формат и обратно. Один из примеров такого шифра — **Бинарный шифр**, где каждый символ текста преобразуется в его двоичное представление.

**История:** Бинарные шифры использовались в ранних вычислительных системах и в системах связи для передачи данных. Они просты в реализации, но криптоустойчивость таких методов невысока, особенно при использовании стандартных преобразований без дополнительных ключей или сложных преобразований.

**Криптоустойчивость:** Бинарные шифры легко поддаются анализу и взлому, поскольку преобразование символов в двоичный код является прямым и предсказуемым. Для повышения устойчивости можно использовать дополнительные методы, такие как XOR-шифрование с ключом.

## Шифр Виженера для русских букв

**Описание алгоритма:** Шифр Виженера — это метод полиалфавитного шифрования, использующий ключевое слово для шифрования текста. Для русского алфавита алгоритм адаптируется с учетом специфики кириллицы.

**История:** Шифр Виженера был изобретен в 16 веке и долгое время считался устойчивым к криптоанализу, пока не были обнаружены методы его взлома. Он остается популярным учебным примером полиалфавитных шифров.

**Криптоустойчивость:** Криптоустойчивость шифра Виженера выше, чем у моноалфавитных шифров, но все же метод уязвим к современным методам криптоанализа, таким как метод Касиски или индекс совпадений.

# Определение ограничений на решаемую задачу

1. **Язык программирования:** Рекомендуется использовать Python из-за его простоты и наличия библиотек для создания графических интерфейсов.
2. **Алфавит:** Для шифра Виженера использовать русский алфавит. Для ШПНиЕ — поддержка как русского, так и английского текста.
3. **Интерфейс:** Простой и интуитивно понятный GUI с возможностью выбора алгоритма, ввода текста, ввода ключа (для Виженера), отображения результата, а также опцией шифрования и дешифрования.
4. **Функциональность:**

* Шифрование и дешифрование текстов.
* Встроенные примеры для демонстрации работы алгоритмов.
* Возможность ввода собственного текста.
* Обработка как русского, так и английского текста (где применимо).

# Реализация алгоритмов в виде программного кода

## Шифровка последовательностей нулей и единиц (ШПНиЕ)

**Пример реализации:**

#include <iostream>

#include <bitset>

#include <string>

#include <sstream>

// Преобразование текста в бинарный вид

std::string textToBinary(const std::string& text) {

std::ostringstream binaryStream;

for (char c : text) {

binaryStream << std::bitset<8>(static\_cast<unsigned char>(c)) << ' ';

}

std::string binaryStr = binaryStream.str();

if (!binaryStr.empty()) {

binaryStr.pop\_back(); // Удаление последнего пробела

}

return binaryStr;

}

// Преобразование бинарного вида обратно в текст

std::string binaryToText(const std::string& binaryStr) {

std::ostringstream textStream;

std::istringstream iss(binaryStr);

std::string byte;

while (iss >> byte) {

char c = static\_cast<char>(std::bitset<8>(byte).to\_ulong());

textStream << c;

}

return textStream.str();

}

int main() {

std::string input;

std::cout << "Введите текст для шифрования: ";

std::getline(std::cin, input);

std::string encrypted = textToBinary(input);

std::cout << "Зашифрованный текст: " << encrypted << std::endl;

std::string decrypted = binaryToText(encrypted);

std::cout << "Расшифрованный текст: " << decrypted << std::endl;

return 0;

}

**Описание:**

* textToBinary: Преобразует каждый символ текста в его 8-битное двоичное представление.
* binaryToText: Преобразует строку двоичных чисел обратно в текст.

## Шифр Виженера для русских букв

**Пример реализации:**

#include <iostream>

#include <string>

#include <algorithm>

// Функция для генерации ключа до длины текста

std::string generateKey(const std::string& text, const std::string& key) {

std::string extendedKey = key;

while (extendedKey.size() < text.size()) {

extendedKey += key;

}

extendedKey = extendedKey.substr(0, text.size());

return extendedKey;

}

// Функция для шифрования текста с использованием шифра Виженера

std::string vigenereEncrypt(const std::string& text, const std::string& key) {

std::string russianAlphabet = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

std::string upperText = text;

std::string upperKey = key;

std::transform(upperText.begin(), upperText.end(), upperText.begin(), ::toupper);

std::transform(upperKey.begin(), upperKey.end(), upperKey.begin(), ::toupper);

std::string extendedKey = generateKey(upperText, upperKey);

std::string encrypted;

for (size\_t i = 0; i < upperText.size(); ++i) {

char currentChar = upperText[i];

char keyChar = extendedKey[i];

size\_t pos = russianAlphabet.find(currentChar);

size\_t keyPos = russianAlphabet.find(keyChar);

if (pos != std::string::npos && keyPos != std::string::npos) {

encrypted += russianAlphabet[(pos + keyPos) % russianAlphabet.size()];

} else {

encrypted += currentChar;

}

}

return encrypted;

}

// Функция для дешифрования текста с использованием шифра Виженера

std::string vigenereDecrypt(const std::string& encryptedText, const std::string& key) {

std::string russianAlphabet = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

std::string upperEncrypted = encryptedText;

std::string upperKey = key;

std::transform(upperEncrypted.begin(), upperEncrypted.end(), upperEncrypted.begin(), ::toupper);

std::transform(upperKey.begin(), upperKey.end(), upperKey.begin(), ::toupper);

std::string extendedKey = generateKey(upperEncrypted, upperKey);

std::string decrypted;

for (size\_t i = 0; i < upperEncrypted.size(); ++i) {

char currentChar = upperEncrypted[i];

char keyChar = extendedKey[i];

size\_t pos = russianAlphabet.find(currentChar);

size\_t keyPos = russianAlphabet.find(keyChar);

if (pos != std::string::npos && keyPos != std::string::npos) {

decrypted += russianAlphabet[(pos + russianAlphabet.size() - keyPos) % russianAlphabet.size()];

} else {

decrypted += currentChar;

}

}

return decrypted;

}

int main() {

std::string text, key;

std::cout << "Введите текст для шифрования (русские буквы): ";

std::getline(std::cin, text);

std::cout << "Введите ключевое слово: ";

std::getline(std::cin, key);

std::string encrypted = vigenereEncrypt(text, key);

std::cout << "Зашифрованный текст: " << encrypted << std::endl;

std::string decrypted = vigenereDecrypt(encrypted, key);

std::cout << "Расшифрованный текст: " << decrypted << std::endl;

return 0;

}

**Описание:**

* generateKey: Расширяет ключевое слово до длины текста.
* vigenereEncrypt: Шифрует текст с использованием ключа.
* vigenereDecrypt: Дешифрует текст с использованием ключа.

# Реализация и описание графического интерфейса

Для создания графического интерфейса на C++ был использован **Qt** — мощный и широко используемый фреймворк для разработки кроссплатформенных приложений.