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 №2
на тему

ЭЛЕМЕНТЫ КРИПТОГРАФИИ

Выполнил: студент гр.253501
Станишевский А.Д.

Проверил: ассистент кафедры информатики
Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Целью данной работы является разработка и реализация программных средств для шифрования и дешифрования текстовых файлов с использованием алгоритмов шифра Цезаря (шифра сдвига) и шифра Виженера, которые обеспечат защиту конфиденциальной информации и возможность её безопасной передачи. Основные задачи включают проектирование и реализацию алгоритмов шифрования и дешифрования, поддерживающих как русский, так и английский алфавиты, с учетом особенностей каждого метода. Необходимо разработать пользовательский интерфейс, позволяющий загружать текстовые файлы, выбирать метод шифрования, задавать параметры (сдвиг для шифра Цезаря или ключ для шифра Виженера) и отображать результаты в удобной форме.

Кроме того, требуется реализовать функциональность для обработки текстовых данных, включая корректное шифрование и дешифрование символов, игнорирование пробелов и знаков препинания, а также добавление визуальных эффектов для улучшения пользовательского опыта. Для подтверждения работоспособности программы необходимо протестировать её на различных текстовых файлах, содержащих как русские, так и английские символы, и убедиться в корректности выполнения операций шифрования и дешифрования. Также следует проверить устойчивость программы к ошибкам, таким как некорректный ввод данных или отсутствие файла, и обеспечить возможность сохранения результатов в текстовый файл.

2 ХОД РАБОТЫ

В ходе выполнения работы была реализована программа для шифрования и дешифрования текстовых файлов с использованием шифра Цезаря и шифра Виженера. На первом этапе была разработана функция для шифра Цезаря, которая поддерживает как русский, так и английский алфавиты. Функция позволяет задавать сдвиг для шифрования и дешифрования, при этом пробелы и знаки препинания остаются без изменений. Для тестирования работы шифра Цезаря был создан текстовый файл. На рисунке 2.1 изображено содержимое текстового файла. Дешифрование этого текста с тем же сдвигом успешно вернуло исходное сообщение, что подтвердило корректность работы алгоритма.

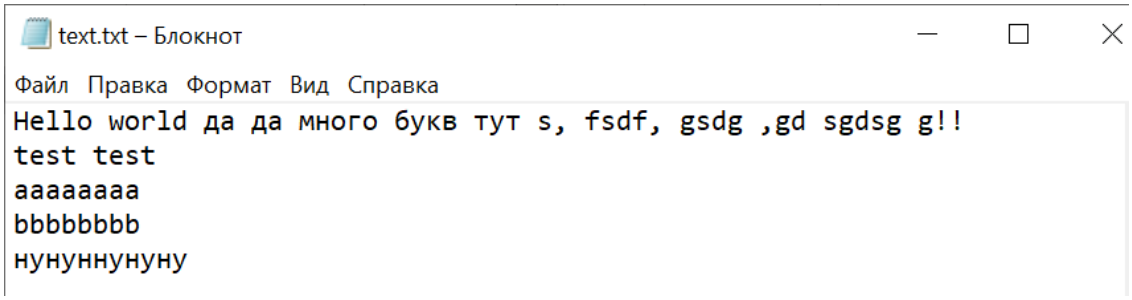

Рисунок 2.1 – Содержимое текстового файла.

На рисунке 2.2 изображено содержимое файла с результатом после выполнения программы.

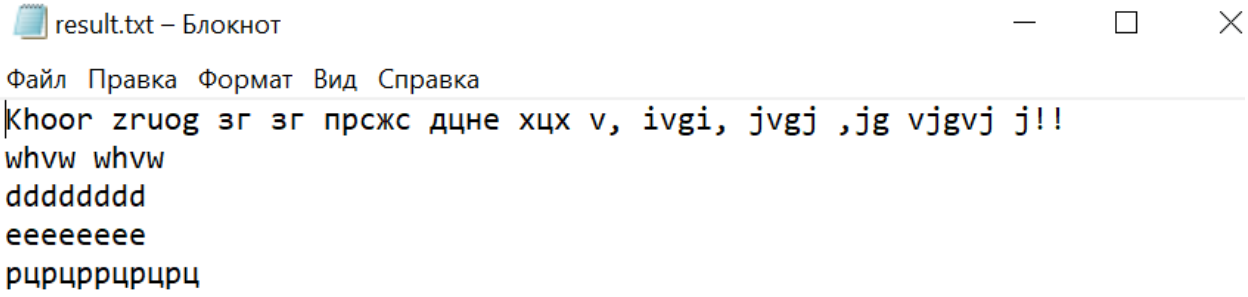

Рисунок 2.2 – Содержимое файла с результатом используя шифр Цезаря.

Для удобства пользователя был разработан интерфейс, позволяющий загружать текстовые файлы, выбирать метод шифрования, задавать параметры (сдвиг для шифра Цезаря или ключ для шифра Виженера) и отображать результаты на экране.

На рисунке 2.3 изображено дешифрование результирующего файла.

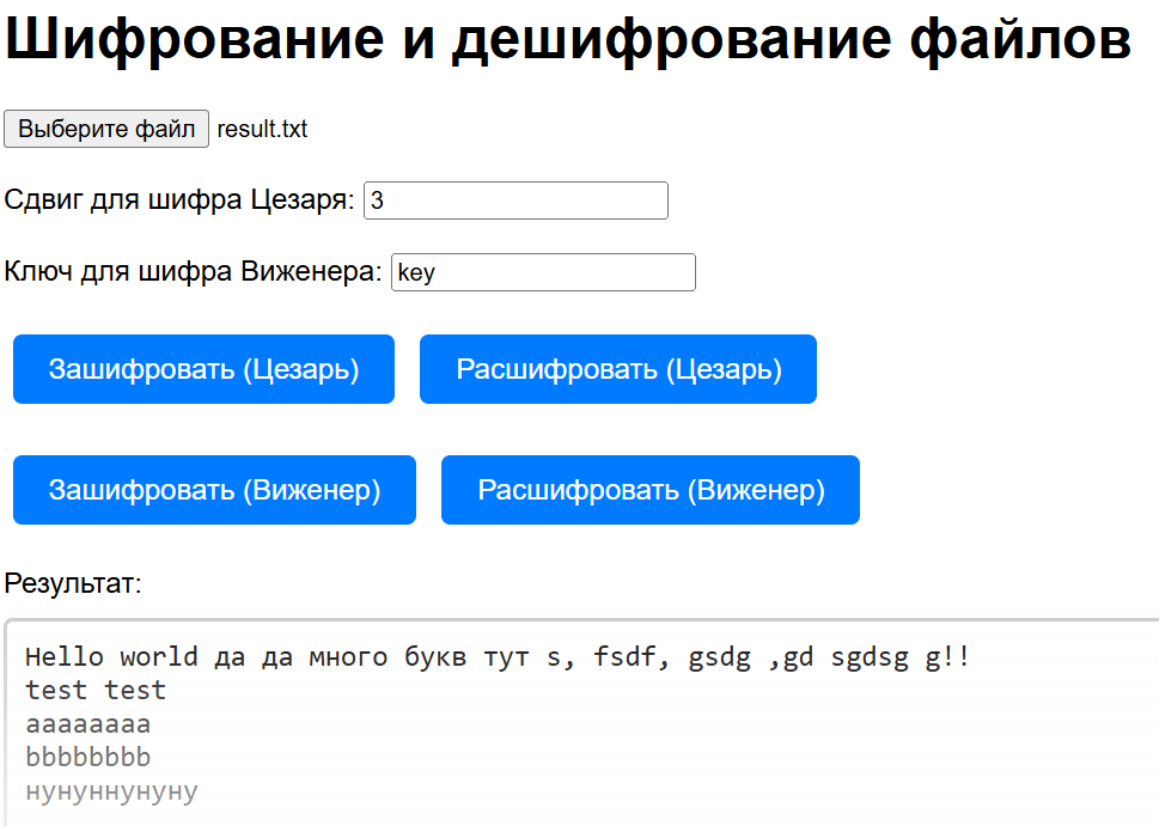

Рисунок 2.3 – Дешифрование шифра Цезаря.

Далее был реализован шифр Виженера, который также поддерживает русский и английский алфавиты. В отличие от шифра Цезаря, шифр Виженера использует ключ для шифрования и дешифрования. Для тестирования был использован тот же файл с текстом. На рисунке 2.4 изображено содержимое полученного текстового файла.

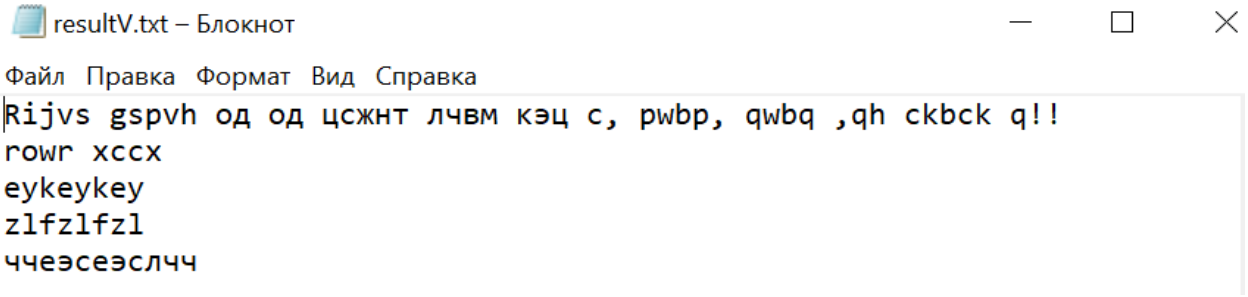

Рисунок 2.4 – Содержимое файла с результатом используя шифр Виженера.

На рисунке 2.5 изображено дешифрование результирующего файла.

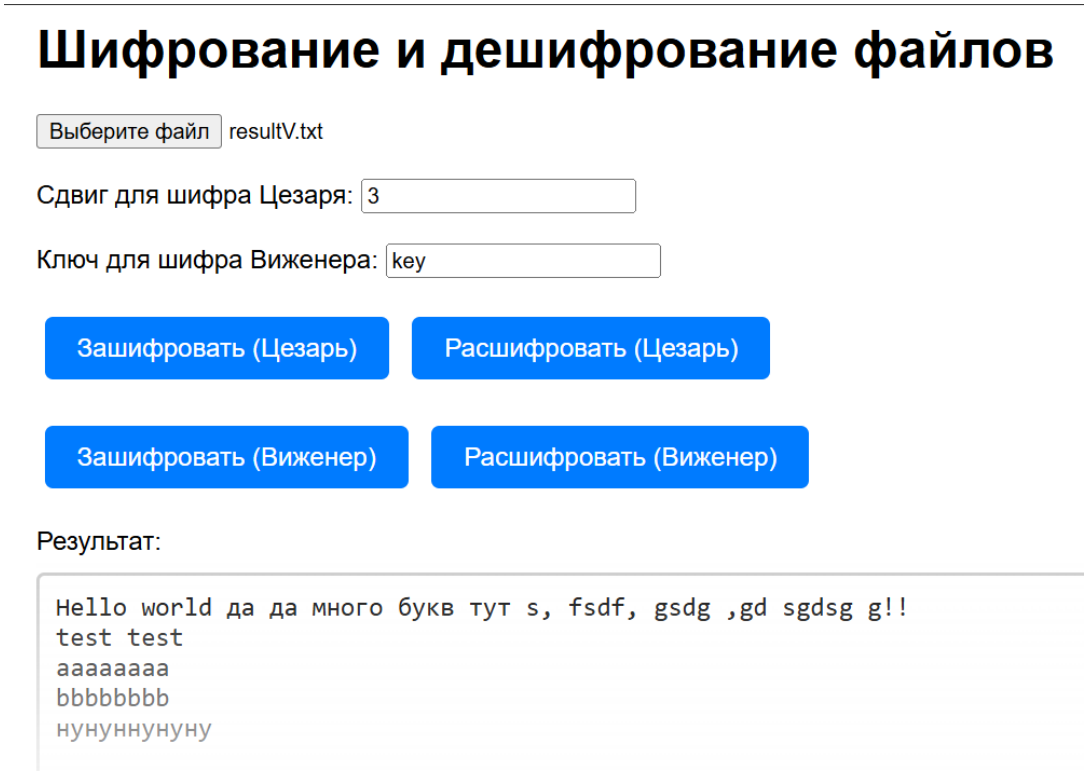

Рисунок 2.5 – Дешифрование шифра Виженера.

Таким образом, в ходе работы были разработаны и реализованы программные средства для шифрования и дешифрования текстовых файлов с использованием шифра Цезаря и шифра Виженера. В процессе выполнения проекта были достигнуты следующие ключевые результаты: успешная реализация алгоритмов шифрования, которые поддерживают как русский, так и английский алфавиты, а также корректно обрабатывают пробелы и знаки препинания, оставляя их без изменений. Разработанная программа позволяет загружать текстовые файлы, выбирать метод шифрования, задавать параметры (сдвиг для шифра Цезаря или ключ для шифра Виженера) и отображать результаты в удобной форме.

ЗАКЛЮЧЕНИЕ

В процессе выполнения работы были разработаны и успешно реализованы программные средства для шифрования и дешифрования текстовых файлов с использованием шифра Цезаря и шифра Виженера. Сначала была определена структура программы, включающая функции для шифрования и дешифрования текста, а также пользовательский интерфейс для удобного взаимодействия с приложением. Программа поддерживает как русский, так и английский алфавиты, при этом пробелы и знаки препинания остаются без изменений, что обеспечивает корректную обработку текста.

После завершения настройки программы было проведено тщательное тестирование на различных текстовых файлах, включая файлы с русскими и английскими символами. Эти тесты позволили проверить корректность работы алгоритмов, а также выявить и устранить потенциальные проблемы, такие как некорректный ввод данных или отсутствие файла. В результате тестирования было подтверждено, что программа корректно шифрует и дешифрует текст, а также устойчива к нештатным ситуациям.

В итоге, благодаря выполненной работе была успешно создана функциональная и стабильная программа, обеспечивающая надежное шифрование и дешифрование текстовых файлов. Все поставленные перед проектом задачи были успешно реализованы, и программа продемонстрировала свою работоспособность как в аспекте обработки текста, так и в плане удобства использования, что открывает возможности для дальнейшего расширения и оптимизации функциональности.

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг программного кода

```
// Цезарь
function caesarCipher(text, shift, decrypt = false) {
    const shiftAmount = decrypt ? (shift * -1) : shift;
    return text.replace(/[a-zA-Za-яA-Я]/g, (char) => {
        let base, alphabetSize;
        if (/[a-zA-Z]/.test(char)) {
            base = char < 'a' ? 'A'.charCodeAt(0) : 'a'.charCodeAt(0);
            alphabetSize = 26;
        } else {
            base = char < 'a' ? 'A'.charCodeAt(0) : 'a'.charCodeAt(0);
            alphabetSize = 32;
        }
        const charCode = char.charCodeAt(0);
        const shiftedCharCode = base + (charCode - base + shiftAmount +
alphabetSize) % alphabetSize;
        return String.fromCharCode(shiftedCharCode);
    });
}
// Виженер
function vigenereCipher(text, key, decrypt = false) {
    const keyLength = key.length;
    let keyIndex = 0;
    return text.split('').map((char) => {
        if (char === '\n') {
            keyIndex = 0;
            return char;
        }
        let base, alphabetSize;
        if (/[a-zA-Z]/.test(char)) {
            base = char < 'a' ? 'A'.charCodeAt(0) : 'a'.charCodeAt(0);
            alphabetSize = 26;
        } else if (/[a-яA-Я]/.test(char)) {
            base = char < 'a' ? 'A'.charCodeAt(0) : 'a'.charCodeAt(0);
            alphabetSize = 32;
        } else {
            return char;
        }
        const keyChar = key[keyIndex % keyLength].toLowerCase();
        let keyShift;
        if (/[a-z]/.test(keyChar)) {
            keyShift = keyChar.charCodeAt(0) - 'a'.charCodeAt(0);
        } else if (/[a-я]/.test(keyChar)) {
            keyShift = keyChar.charCodeAt(0) - 'a'.charCodeAt(0);
        } else {
            return char;
        }
        const shiftAmount = decrypt ? (alphabetSize - keyShift) % alphabetSize
: keyShift;
        const charCode = char.charCodeAt(0);
        const shiftedCharCode = base + (charCode - base + shiftAmount +
alphabetSize) % alphabetSize;
        keyIndex++;
        return String.fromCharCode(shiftedCharCode);
    }).join('');
}
```