**Лабораторная работа № 3**

**ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ШИФРОВ НА ОСНОВЕ ПЕРЕСТАНОВКИ СИМВОЛОВ**

Цель: изучение и приобретение практических навыков разработки и использования приложений для реализации перестановочных шифров (работа рассчитана на 4 часа аудиторных занятий).

Задачи:

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости перестановочных шифров.

2. Ознакомиться с особенностями реализации и свойствами различных перестановочных шифров на основе готового программного средства (L\_LUX).

3. Разработать приложение для реализации указанных преподавателем методов перестановочного зашифрования/расшифрования.

4. Выполнить исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях.

5. Оценить скорость зашифрования/расшифрования реализованных способов шифров.

6. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Практическое задание**

Рекомендация! Перед выполнением практического задания целесообразно освежить практические навыки использования и особенности функционирования программного средства L\_LUX, реализующего перестановочные (и другие) методы зашифрования/расшифрования текстовой информации и являющегося приложением на компакт-диске к [6].

Обратим внимание на использование «горячих» клавиш для реализации некоторых операций:

• Ctrl + F3 – зашифрование на основе простой перестановки;

• Shift + F3 – расшифрование на основе простой перестановки;

• Shift + Ctrl + F1 – вывод гистограмм (частотных параметров символов) исходного и зашифрованного сообщений;

• Shift + Ctrl + F2 – вывод гистограмм (частотных параметров символов) зашифрованного и расшифрованного сообщений.

**Основное задание**

1. Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции:

• выполнять зашифрование/расшифрование текстовых доку ментов (объемом не менее 500 знаков), созданных на основе алфавита языка в соответствии с нижеследующей таблицей вариантов задания; при этом следует использовать шифры подстановки из третьего столбца данной таблицы (варианты задания в табл. 1.1);

Таблица 1.1 Варианты заданий



Задание 1: Маршрутная перестановка (маршрут змейка)

Код шифрования и дешифрования представлен в листинге 1.1.

if (mode === 'encrypt') {

const table: string[][] = Array.from({ length: ROWS }, () => Array(COLS).fill(' '));

let index = 0;

for (let i = 0; i < ROWS; i++) {

for (let j = 0; j < COLS; j++) {

if (index < text.length) {

const col = i % 2 === 0 ? j : COLS - 1 - j;

table[i][col] = text[index++];

}

}

}

result = table.flat().join('');

} else if (mode === 'decrypt') {

const table: string[][] = Array.from({ length: ROWS }, () => Array(COLS).fill(' '));

let index = 0;

for (let i = 0; i < ROWS; i++) {

for (let j = 0; j < COLS; j++) {

const col = i % 2 === 0 ? j : COLS - 1 - j;

if (index < text.length) {

table[i][col] = text[index++];

}

}

}

for (let j = 0; j < COLS; j++) {

for (let i = 0; i < ROWS; i++) {

if (table[i][j] !== ' ') {

result += table[i][j];

}

}

}

}

Листинг 1.1 Шифрование и расшифрование методом маршрутной перестановки

Результат выполнения программы представлен на рисунке 1..

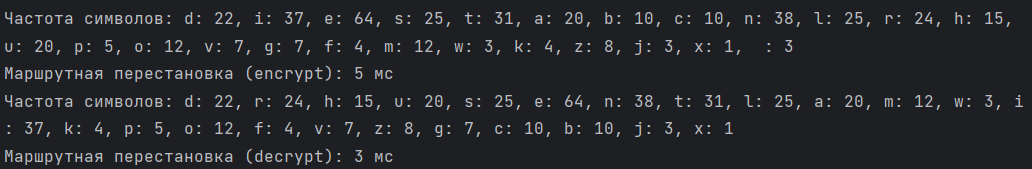


Рисунок 1. Результат работы программы шифрования методом маршрутной перестановки

Задание 2: Множественная перестановка.

Код шифрования и дешифрования представлен в листинге 1.2.

if (mode === 'encrypt') {

const table: string[][] = Array.from({ length: ROWS }, () => Array(COLS).fill(' '));

let index = 0;

for (let i = 0; i < ROWS; i++) {

for (let j = 0; j < COLS; j++) {

if (index < text.length) {

table[i][j] = text[index++];

}

}

}

const permutedTable: string[][] = rowOrder.map((row) => colOrder.map((col) => table[row][col]));

result = permutedTable.flat().join('');

} else if (mode === 'decrypt') {

const permutedTable: string[][] = Array.from({ length: ROWS }, () => Array(COLS).fill(' '));

let index = 0;

for (let row of rowOrder) {

for (let col of colOrder) {

if (index < text.length) {

permutedTable[row][col] = text[index++];

}

}

}

result = permutedTable.map((row) => row.join('')).join('');

result = result.replace(/\\*+$/g, '');

}

Листинг 1.2 Шифрование и расшифрование методом множественной перестановки

Результат выполнения программы представлен на рисунке 1..

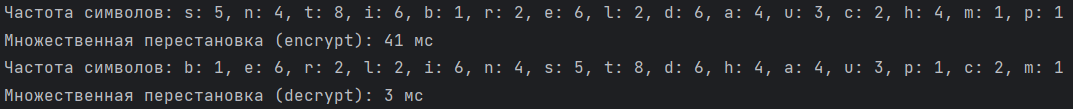


Рисунок 1. Результат работы программы шифрования методом множественной перестановки

• формировать гистограммы частот появления символов для исходного и зашифрованного сообщений;

• оценивать время выполнения операций зашифрования/расшифрования (напоминание: во многих языках программирования есть встроенные методы для замеров времени; при отсутствии такового в используемом языке можно воспользоваться разностью двух дат (например, в миллисекундах: время после выполнения программы – время до начала выполнения пре образования)).

При анализе полученных гистограмм можно сопоставить полученные данные с аналогичными результатами выполнения лабораторной работы № 2 из [2] и лабораторной работы № 2 настоящего пособия.

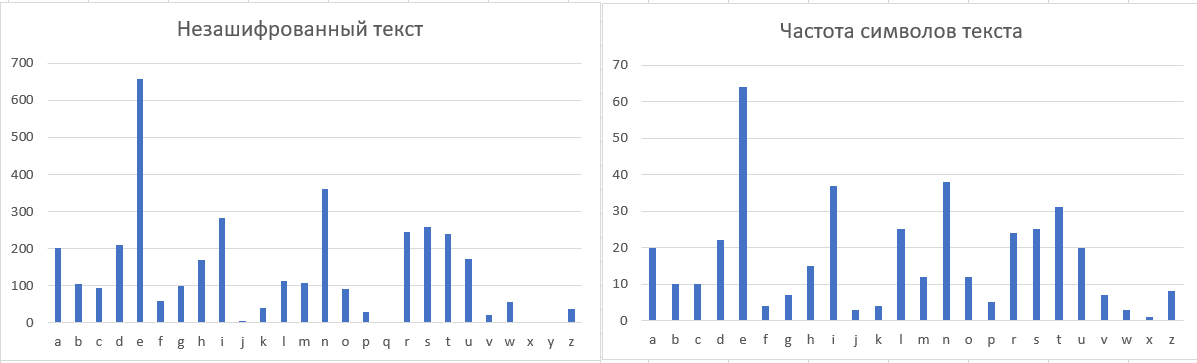


Рисунок 1. Сопоставление гистограмм частоты символов

Если указанный в таблице язык исходного текста не известен разработчику программного средства, можно взять документ на требуемом языке и воспользоваться доступным электронным переводчиком (возникающие при этом отдельные семантические неточности не следует считать существенным недостатком выполняемого анализа).

3. Результаты оформить в виде отчета по установленным правилам.