Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Криптографические методы защиты информации

**Отчет по лабораторной работе №5:**

**«Исследование криптографических шифров на основе перестановки символов»**

2 вариант

Студент: Войцехович Е.А.

ФИТ 3 курс 10 группа

Преподаватель: Ржеутская Н.В.

Минск 2023

**Цель**: изучение и приобретение практических навыков разработки и использования приложений для реализации перестановочных шифров.

**Практическая часть**

Таблица 1

Задание

|  |  |
| --- | --- |
| Вариант | Шифр |
| 11 | 1. Маршрутная перестановка (маршрут: запись по строкам, считывание – по столбцам таблицы; параметры таблицы – по указанию преподавателя)  2. Множественная перестановка, ключевые слова – собственные имя и фамилия |

var start, stop;

function process(action) {

start = new Date().getTime();

let firstKey = document.getElementById('firstKey').value.toUpperCase();

let secondKey = document.getElementById('secondKey').value.toUpperCase();

let inputMessage = '';

inputMessage = document.getElementById('inputMessage').value;

inputMessage = inputMessage.replace(/\s/g, '').toUpperCase();

for (let i = inputMessage.length; i < firstKey.length \* secondKey.length; i++) {

inputMessage += '\_';

};

let array\_firstKey = firstKey.split('');

let array\_secondKey = secondKey.split('');

let sorted\_firstKey = alphabetize(firstKey);

let array\_sorted\_firstKey = sorted\_firstKey.split('');

console.log(sorted\_firstKey);

let sorted\_secondKey = alphabetize(secondKey);

let array\_sorted\_secondKey = sorted\_secondKey.split('');

function alphabetize(str) {

return str.split('').sort().join('');

}

let numbers\_firstKey = [];

for (const letter of array\_firstKey) {

const indexLetter = array\_sorted\_firstKey.indexOf(letter);

array\_sorted\_firstKey[indexLetter] = '\_';

if (numbers\_firstKey.indexOf(indexLetter + 1) != -1) {

numbers\_firstKey.push(indexLetter + 2);

}

else

numbers\_firstKey.push(indexLetter + 1);

}

let numbers\_secondKey = [];

for (const letter of array\_secondKey) {

const indexLetter = array\_sorted\_secondKey.indexOf(letter);

array\_sorted\_secondKey[indexLetter] = '\_';

if (numbers\_secondKey.indexOf(indexLetter + 1) != -1) {

numbers\_secondKey.push(indexLetter + 2);

}

else

numbers\_secondKey.push(indexLetter + 1);

}

let cryptResult = '';

if (action == 'encrypt') {

cryptResult = encrypt(inputMessage, numbers\_firstKey, numbers\_secondKey); // шифруем

}

else cryptResult = decrypt(inputMessage, numbers\_firstKey, numbers\_secondKey); // дешифруем

// формируем строку с результатом

let regExp = new RegExp('.{' + firstKey.length + '}', 'g');

let resultLine = cryptResult.right.join('').replace(regExp, '$& ');

let leftTableView = '';

let middleTableView = '';

let rightTableView = '';

for (let j = 0; j < secondKey.length; j++) {

// вырезаем из начала массива одну строку и оборачиваем каждую букву в теги <td></td>

// а строку в <tr></tr>

leftTableView += '<tr>' + cryptResult.left.splice(0, firstKey.length).join('').replace(/./g, '<td>$&</td>').toLowerCase() + '</tr>'

middleTableView += '<tr>' + cryptResult.middle.splice(0, firstKey.length).join('').replace(/./g, '<td>$&</td>').toLowerCase() + '</tr>'

rightTableView += '<tr>' + cryptResult.right.splice(0, firstKey.length).join('').replace(/./g, '<td>$&</td>').toLowerCase() + '</tr>';

}

leftTableView = '<table><caption>Исходное сообщение</caption>' + leftTableView + '</table>';

middleTableView = '<table><caption>Перестановка столбцов</caption>' + middleTableView + '</table>';

rightTableView = '<table><caption>Перестановка строк</caption>' + rightTableView + '</table>';

stop = new Date().getTime();

alert(stop - start);

document.getElementById('output').innerHTML = leftTableView + middleTableView + rightTableView;

document.getElementById('inputMessage').innerText = resultLine.toLowerCase();

}

function encrypt(message, firstKey, secondKey) {

let result = [];

let middle = [];

let left = [];

let colsCount = firstKey.length;

let rowsCount = secondKey.length;

for (let row = 0; row < rowsCount; row++) {

for (let col = 0; col < colsCount; col++) {

let newCol = firstKey[col] - 1;

let newRow = secondKey[row] - 1;

left[row \* colsCount + col] = message[row \* colsCount + col];

middle[row \* colsCount + newCol] = message[row \* colsCount + col];

result[newRow \* colsCount + newCol] = message[row \* colsCount + col];

}

}

return { left: left, middle: middle, right: result };

}

function decrypt(message, firstKey, secondKey) {

let result = [];

let middle = [];

let left = [];

let colsCount = firstKey.length;

let rowsCount = secondKey.length;

for (let row = 0; row < rowsCount; row++) {

for (let col = 0; col < colsCount; col++) {

let newCol = firstKey[col] - 1;

let newRow = secondKey[row] - 1;

left[row \* colsCount + col] = message[row \* colsCount + col];

middle[row \* colsCount + col] = message[row \* colsCount + newCol];

result[row \* colsCount + col] = message[newRow \* colsCount + newCol];

}

}

return { left: left, middle: middle, right: result };

}

const btnEncryptRoute = document.querySelector('#encryptRoute');

const btnDecryptRoute = document.querySelector('#decryptRoute');

btnEncryptRoute.addEventListener('click', () => {

start = new Date().getTime();

let message = document.querySelector('#text').value;

let rows = document.querySelector('#key').value;

message = message.replace(/\s/g, '');

var columns = Math.ceil(message.length / rows);

var encryptedMessage = new Array(rows);

for (var i = 0; i < rows; i++) {

encryptedMessage[i] = new Array(columns);

}

var index = 0;

for (var i = 0; i < rows; i++) {

for (var j = 0; j < columns; j++) {

if (index < message.length) {

encryptedMessage[i][j] = message.charAt(index);

index++;

} else {

encryptedMessage[i][j] = "\_";

}

}

}

var result = "";

for (var j = 0; j < columns; j++) {

for (var i = 0; i < rows; i++) {

result += encryptedMessage[i][j];

}

}

stop = new Date().getTime();

alert(stop - start);

document.getElementById('text').innerText = result;

});

btnDecryptRoute.addEventListener('click', () => {

start = new Date().getTime();

let message = document.querySelector('#text').value;

let rows = document.querySelector('#key').value;

message = message.replace(/\s/g, '');

var columns = Math.ceil(message.length / rows);

var encryptedMessage = new Array(rows);

for (var i = 0; i < rows; i++) {

encryptedMessage[i] = new Array(columns);

}

var index = 0;

for (var j = 0; j < columns; j++) {

for (var i = 0; i < rows; i++) {

if (index < message.length) {

encryptedMessage[i][j] = message.charAt(index);

index++;

} else {

encryptedMessage[i][j] = "\_";

}

}

}

var result = "";

for (var i = 0; i < rows; i++) {

for (var j = 0; j < columns; j++) {

result += encryptedMessage[i][j];

}

}

stop = new Date().getTime();

alert(stop - start);

document.getElementById('text').innerText = result;

});

Листинг 1 – Реализация шифров

**Выводы:** приобретены практические навыки перестановочного зашифрования/расшифрования и разработано приложение для автоматизации этих операций.