Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.01 – «Информатика и вычислительная техника»

Дисциплина: «Защита информации»

Профиль: «Автоматизированные системы обработки информации и управления»

Семестр 7

ОТЧЕТ

по лабораторной работе №1

Тема: «Шифры перестановки и замены»

Выполнил: студент группы АСУ-14-1б

Калмыков В. А. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2017

**ЦЕЛЬ РАБОТЫ**

Получить практические навыки по применению шифров перестановки и шифров простой замены.

**ЗАДАНИЕ**

Реализовать шифрование текстового сообщения, используя систему шифрования Вижинера.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Широко распространена разновидность шифра маршрутной перестановки, называемая «шифром вертикальной перестановки» (ШВП). В нем снова используется прямоугольник, в который сообщение вписывается обычным способом (по строкам слева направо). Выписываются буквы по вертикали, а столбцы при этом берутся в порядке, определяемом ключом. Пусть, например, этот ключ таков: (5,4,1,7,2,6,3), и с его помощью надо зашифровать сообщение:

ВОТПРИМЕРШИФРАВЕРТИКАЛЬНОЙПЕРЕСТАНОВКИ

Впишем сообщение в прямоугольник, столбцы которого пронумерованы в соответствии с ключом:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 1 | 4 | 7 | 2 | 6 | 3 |
| В | О | Т | П | Р | И | М |
| Е | Р | Ш | И | Ф | Р | А |
| В | Е | Р | Т | И | К | А |
| Л | Ь | Н | О | Й | П | Е |
| Р | Е | С | Т | А | Н | О |
| В | К | И | - | - | - | - |

Теперь, выбирая столбцы в порядке, заданном ключом, и выписывая последовательно буквы каждого из них сверху вниз, получаем такую криптограмму:

ОРЕЬЕКРФИЙА-МААЕО-ТШРНСИВЕВЛРВИРКПН-ПИТОТ-

Число ключей ШВП не более m!, где m - число столбцов таблицы. Как правило, m гораздо меньше, чем длина текста n (сообщение укладывается в несколько строк по m букв), а, значит, и m! много меньше n!.

В случае, когда ключ ШВП не рекомендуется записывать, его можно извлекать из какого-то легко запоминающегося слова или предложения. Для этого существует много способов. Наиболее распространенный состоит в том, чтобы приписывать буквам числа в соответствии с обычным алфавитным порядком букв. Например, пусть ключевым словом будет ПЕРЕСТАНОВКА. Присутствующая в нем буква А получает номер 1. Если какая-то буква входит несколько раз, то ее появления нумеруются последовательно слева направо. Поэтому второе вхождение буквы А получает номер 2. Поскольку буквы Б в этом слове нет, то буква В получает номер 3 и так далее. Процесс продолжается до тех пор, пока все буквы не получат номера. Таким образом, мы получаем следующий ключ:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| П | Е | Р | Е | С | Т | А | Н | О | В | К | А |
| 9 | 4 | 10 | 5 | 11 | 12 | 1 | 7 | 8 | 3 | 6 | 2 |

**ХОД РАБОТЫ**

На рисунке 1 представлена форма, в первое поле которой вводится шифруемая фраза, а во второе числовой шифр. Числа должны быть отделены между собой пробелом.

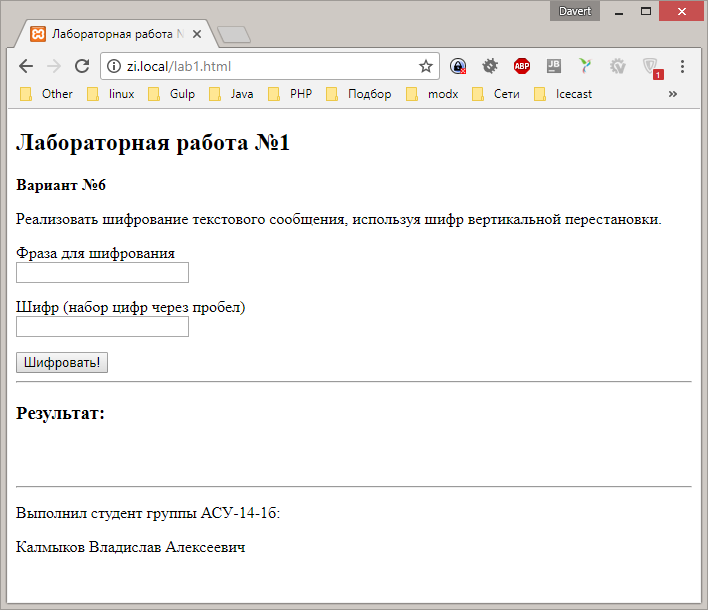


Рисунок 1 – Форма для шифрования.

Пример работы программы представлен на рисунке 2.

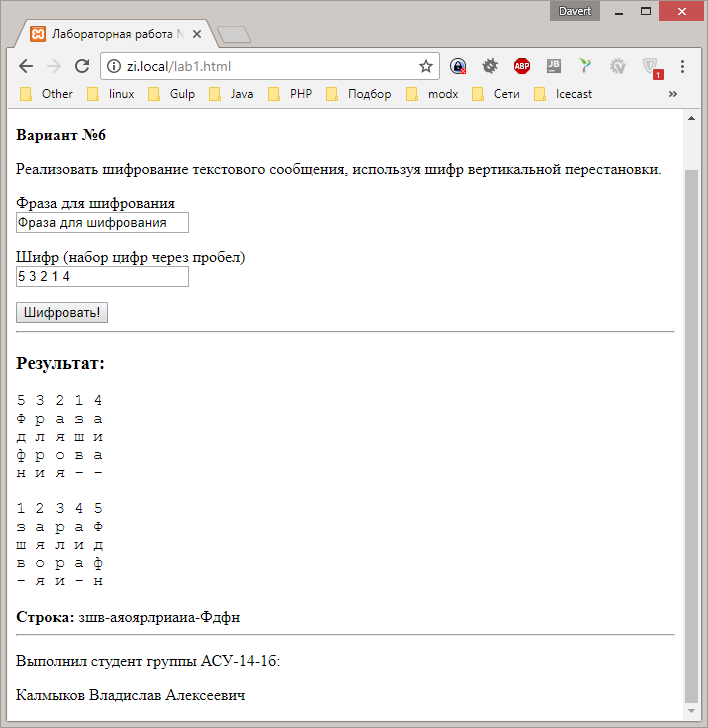


Рисунок 2 – Пример работы программы.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

'use strict';

(() => {

document.getElementById('encrypt').addEventListener('click', function (e) {

e.preventDefault();

let text = document.getElementsByName('text')[0].value.replace(/\s\*/gi, ''),

codeText = document.getElementsByName('code')[0].value,

codeArr = codeText.split(' '),

data = [codeArr],

resultTable = [],

resultTableStr = '',

resultStr = '';

for (let i = 0, j = 0; i < text.length; j++) {

let temp = [];

while (temp.length < codeArr.length) {

temp.push(text[i] !== undefined ? text[i++] : '-');

}

data.push(temp);

}

data.map((arr) => {

resultTable.push(new Array(arr.length));

resultTableStr += `${arr.join(' ')}\n`;

});

document.getElementsByClassName('result\_\_table-begin')[0].innerText = resultTableStr;

for (let i = 0; i < codeArr.length; i++) {

let col = codeArr[i] - 1;

for (let row = 0; row < resultTable.length; row++) {

resultTable[row][col] = data[row][i];

}

}

resultTableStr = '';

resultTable.map((arr) => {

resultTableStr += `${arr.join(' ')}\n`;

});

for(let col = 0; col < resultTable[0].length; col++) {

for(let row = 1; row < resultTable.length; row++) {

resultStr += resultTable[row][col];

}

}

document.getElementsByClassName('result\_\_table-end')[0].innerText = resultTableStr;

document.getElementsByClassName('result\_\_string')[0].innerHTML = `<strong>Строка:</strong> ${resultStr}`;

});

})();