#### Министерство науки и высшего образования Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение науки и высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа № 07

по дисциплине

"Администрирование и безопасность программно-информационных систем"

Тема: "Шифрование с помощью аналитических преобразований"

Выполнил:

ст. гр. ПРИ-117

В.Г. Хлызова

Принял:

И.В. Спирин

Владимир, 2020 г.

## ЦЕЛЬ РАБОТЫ

Освоить на практике применение алгоритма шифрования с помощью аналитических преобразований.

ОПИСАНИЕ АЛГОРИТМА

Для примера зашифруем слово “тюлень”, используя в качестве ключа следующую матрицу:

Заменим буквы алфавита цифрами, соответствующими их порядковому номеру в алфавите:



Полученный текст делим на матрицы по три элемента и умножаем каждую матрицу на ключ:



Зашифрованный текст: 380 -3 405 451 -30 205

Для расшифровки нашего текста сначала найдем обратную матрицу для ключа:



Транспонируем матрицу:



Находим алгебраические дополнения:





Находим обратную матрицу путем умножения матрицы алгебраических дополнений на 1/определитель:



Текст зашифрованного сообщения делим на матрицы по три элемента и умножаем каждую матрицу на обратную матрицу ключа, возвращаем буквы расшифрованного сообщения:



ЗАДАНИЕ

Зашифровать описанным методом следующее выражение: «Помехоустойчивое кодирование – это кодирование с возможностью восстановления потерянных или ошибочно принятых данных». В качестве ключа использовать следующую матрицу:

<?php

    //умножение матрицы символов для шифрования на ключ

    function multiplication($key, $matrix){

        $result=[];

        for($i=0; $i<3; $i++){

            $sum=0;

            for($j=0; $j<3; $j++){

               $sum+=$key[$i][$j]\*$matrix[$j];

            }

            array\_push($result,$sum);

        }

        return $result;

    }

    //определитель матрицы два на два

    function det\_matrix\_2x2($matrix){

        return $matrix[0]\*$matrix[3]-$matrix[1]\*$matrix[2];

    }

    //определитель матрицы три на три

    function det\_matrix\_3x3($matrix){

        $det=$matrix[0][0]\*det\_matrix\_2x2([$matrix[1][1], $matrix[1][2],

                                            $matrix[2][1],$matrix[2][2]]);

        $det-=$matrix[0][1]\*det\_matrix\_2x2([$matrix[1][0], $matrix[1][2],

                                            $matrix[2][0],$matrix[2][2]]);

        $det+=$matrix[0][2]\*det\_matrix\_2x2([$matrix[1][0], $matrix[1][1],

                                            $matrix[2][0],$matrix[2][1]]);

        return $det;

    }

    //функция для нахождения обратной матрицы

    function inverse\_matrix($key){

        $transposed\_matrix=[];

        for($j=0; $j<3; $j++){

            $matrix=[];

            for($i=0; $i<3; $i++){

                array\_push($matrix, $key[$i][$j]);

            }

            array\_push($transposed\_matrix, $matrix);

        }

       $det=det\_matrix\_3x3($key);

       $inverse\_matrix=[];

       for($i=0; $i<3; $i++){

            $matrix=[];

            for($j=0; $j<3; $j++)

            {

                $matrix2=[];

                for($i1=0; $i1<3; $i1++)

                    for($j1=0; $j1<3; $j1++)

                        if(($i1!=$i)and($j1!=$j)){

                            array\_push($matrix2, $transposed\_matrix[$i1][$j1]);

                        }

                        $item=(det\_matrix\_2x2($matrix2)\*((-1)\*\*($i+$j)))/$det;

                        array\_push($matrix, $item);

            }

            array\_push($inverse\_matrix, $matrix);

        }

        return $inverse\_matrix;

    }

    $key = [[1, 7, 12],

            [8, -5, 0],

            [7, 8, 2]];

    $alphabet=[ 'а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и',

                'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т',

                'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь',

                'э', 'ю', 'я', ' '];

    echo "\n выберите действия\n   1: зашифровать сообщение\n   2: расшифровать сообщение\n";

    $action=readline(" ");

    switch ($action){

        case 1:

            //получаем сообщение, переводим все буквы в нижний регистр и удаляем знаки препинания

            $msg = readline("  введите текст сообщения: ");

            $msg=mb\_strtolower($msg);

            $array\_of\_characters = array(".", ",", "-", ":", "«", "»");

            foreach($array\_of\_characters as $character)

                $msg=str\_replace($character, '',$msg);

            //разбиваем сообщение на массив символов

            $msg = preg\_split("//u", $msg, -1, PREG\_SPLIT\_NO\_EMPTY);

            //Заменим буквы алфавита цифрами, соответствующими их порядковому номеру в алфавите

            $index\_number\_msg=[];

            foreach($msg as $m){

                array\_push($index\_number\_msg, array\_search($m, $alphabet));

            }

            //кол-во элементов массива олжно быть равным 3, для последующего шифрования

            $first\_count=count($index\_number\_msg);

            while(count($index\_number\_msg)%3!=0){

                array\_push($index\_number\_msg, 99);

            }

            //разбиваем имеющийся массив на матрицы по три элемента

            $index\_number\_msg=array\_chunk($index\_number\_msg, 3);

            //перемножаем матрицу-ключ и полученные из сообщения буквы

            $encrypted\_message=[];

            foreach($index\_number\_msg as $item){

                $encrypted\_message=array\_merge($encrypted\_message, multiplication($key, $item));

            }

            echo "\n\n  зашифрованное сообщение: ";

            for($i=0;$i<$first\_count;$i++) echo $encrypted\_message[$i]." ";

            break;

        case 2:

            $msg = readline('введите текст сообщения: ');

            //получаем массив символов из введенной строки

            $msg = preg\_split("/[\s,]+/",  $msg);

           ///кол-во элементов массива олжно быть равным 3, для последующего шифрования

            $first\_count=count($msg);

            while(count($msg)%3!=0){

                array\_push($msg, 99);

            }

            //делим элементы массива-сообщения на группы по три символа

            $index\_number\_msg=array\_chunk($msg, 3);

            //получаем обратную матрицу для расшифровки

            $inverse\_matrix=inverse\_matrix($key);

            //перемножаем матрицу-ключ и полученные из сообщения буквы

            $encrypted\_message=[];

            foreach($index\_number\_msg as $item){

                $matrix=[];

                foreach(multiplication($inverse\_matrix, $item) as $m){

                    array\_push($matrix, round($m));

                }

                $encrypted\_message=array\_merge($encrypted\_message, $matrix);

            }

            //возвращаем буквы

            $msg=[];

            foreach($encrypted\_message as $message){

               if($message!=99) array\_push($msg, $alphabet[$message]);

            }

            echo "\n  расшифрованное сообщение: ";

            for($i=0; $i<count($msg) ;$i++) echo $msg[$i];

            break;

        default:

            echo "\n был введен не верный символ";

        }

?>

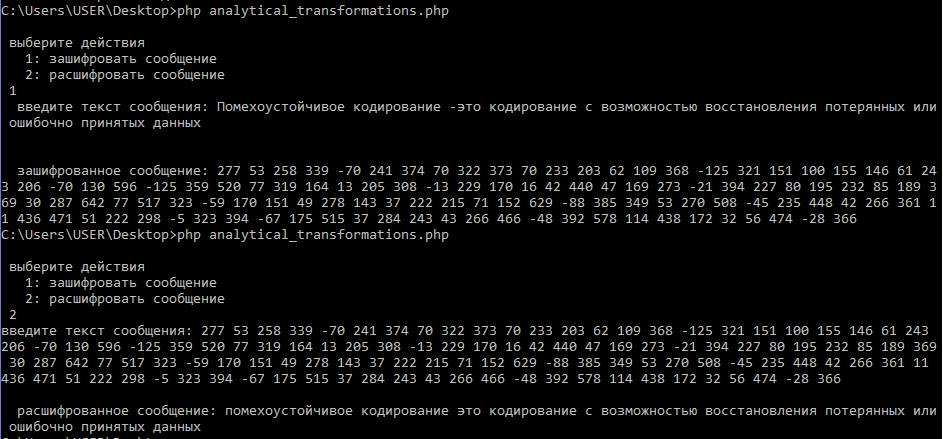


Рисунок 1. Зашифровываем и расшифровываем сообщение.

ВЫВОД

В ходе лабораторной работы было выполнено шифрование с помощью аналитических преобразований.