Протокол самостоятельной работы

По Теории Защиты Информации

На тему: Шифр Полибия

Выполнил студент К17.1

Шишелов Владимир

***«Квадрат Полибия»***

Система Цезаря не является старейшей. Возможно, что наиболее древней из известных  является  система  греческого историка Полибия,  умершего  за 30 лет  до  рождения Цезаря. Устройство  для шифрования,  называемое  квадратом Полибия, или полибианским квадратом, представляло собой квадрат,  размер  этого  квадрата  зависит  от  мощности используемого алфавита. В каждую клетку этого квадрата вписывалась буква в порядке её следования в алфавите. В  процессе  шифрования  каждой  букве  ставилась  в  соответствие  пара чисел –  это  номер  столбца  и  номер  строки,  на  пересечении  которых располагалась  шифруемая  буква.

К каждому языку отдельно составляется таблица шифрования с одинаковым (не обязательно) количеством пронумерованных строк и столбцов. Берутся два целых числа, произведение которых ближе всего к количеству букв в языке — получаем нужное число строк и столбцов. Затем вписываем в таблицу все буквы алфавита подряд — по одной на каждую клетку. При нехватке клеток можно вписать в одну две буквы (редко употребляющиеся или схожие по употреблению).

Для латинского алфавита: в современном латинском алфавите 26 букв, следовательно таблица должна состоять из 5 строк и 5 столбцов, так как 25=5\*5 наиболее близкое к 26 число.

Для русского алфавита: число букв в русском алфавите отличается от числа букв в греческом алфавите, поэтому размер таблицы выбран другой (квадрат 6\*6=36, поскольку 36 наиболее близкое число к 33)

Возможен также другой вариант составления, предусматривающий объединение букв Е и Ё, И и Й, Ъ и Ь

Используя подобный алгоритм таблицу шифрования можно задать для любого языка. Чтобы расшифровать закрытый текст необходимо знать, таблицей шифрования какого алфавита он зашифрован.

Методы шифрования

Существует несколько вариантов шифрования данных с помощью квадрата Полибия. Рассмотрим наиболее известные из них на примере шифрования латинского текста с квадратом без использования ключа.

Метод 1. Использование соседнего снизу символа

В этом методе для шифрования в квадрате ищут исходную букву и в качестве кода берут нижнюю от нее расположенную в том же столбце, или самую верхнюю, если искомый символ расположен в последней строке.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица координат | | | | | | | | |
| Исходный текст: | P | O | L | Y | B | I | U | S |
| Шифротекст: | U | T | Q | D | G | O | Z | X |

Метод 2. Преобразование координат

Для начала необходимо выписать координаты символов исходного текста:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Координаты | | | | | | | |
| Буква: | P | O | L | Y | B | I | U | S |
| Вертикальная координата: | 5 | 4 | 1 | 4 | 2 | 4 | 5 | 3 |
| Горизонтальная координата: | 3 | 3 | 3 | 5 | 1 | 2 | 4 | 4 |

**Реализация Шифра на языке программирования С#.**

class PolybiusSquare

{

char[,] square;

string alphabet;

Method encryptMethod;

public PolybiusSquare(string alphabet = null, Method cipherMethod = Method.Method1)

{

this.alphabet = alphabet ?? "ABCDEFGHIJKLMNOPQRSTUVWXYZ ";

encryptMethod = cipherMethod;

}

//возвращает квадрат Полибия

char[,] GetSquare(string key)

{

var newAlphabet = alphabet;

//удаляем из алфавита все символы которые содержит ключ

for (int i = 0; i < key.Length; i++)

{

newAlphabet = newAlphabet.Replace(key[i].ToString(), "");

}

//добавляем пароль в начало алфавита, а в конец дополнительные знаки

//для того чтобы избежать пустых ячеек

newAlphabet = key + newAlphabet + "0123456789!@#$%^&\*)\_+-=<>?,.";

//получаем размер стороны квадрата

//округлением квадратного корня в сторону большего целого числа

var n = (int)Math.Ceiling(Math.Sqrt(alphabet.Length));

//создаем и заполняем массив

square = new char[n, n];

var index = 0;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (index < newAlphabet.Length)

{

square[i, j] = newAlphabet[index];

index++;

}

}

}

return square;

}

//поиск символа в двухмерном массиве

bool FindSymbol(char[,] symbolsTable, char symbol, out int column, out int row)

{

var l = symbolsTable.GetUpperBound(0) + 1;

for (int i = 0; i < l; i++)

{

for (int j = 0; j < l; j++)

{

if (symbolsTable[i, j] == symbol)

{

//значение найдено

row = i;

column = j;

return true;

}

}

}

//если ничего не нашли

row = -1;

column = -1;

return false;

}

public string PolibiusEncrypt(string text, string password)

{

var outputText = "";

var square = GetSquare(password);

switch (encryptMethod)

{

case Method.Method1:

for (int i = 0; i < text.Length; i++)

{

if (FindSymbol(square, text[i], out int columnIndex, out int rowIndex))

{

var newRowIndex = rowIndex == square.GetUpperBound(1)

? 0

: rowIndex + 1;

outputText += square[newRowIndex, columnIndex].ToString();

}

}

break;

case Method.Method2:

var m = text.Length;

var coordinates = new int[m \* 2];

for (int i = 0; i < m; i++)

{

if (FindSymbol(square, text[i], out int columnIndex, out int rowIndex))

{

coordinates[i] = columnIndex;

coordinates[i + m] = rowIndex;

}

}

for (int i = 0; i < m \* 2; i += 2)

{

outputText += square[coordinates[i + 1], coordinates[i]];

}

break;

}

return outputText;

}

public string PolybiusDecrypt(string text, string password)

{

var outputText = "";

var square = GetSquare(password);

var m = text.Length;

switch (encryptMethod)

{

case Method.Method1:

for (int i = 0; i < m; i++)

{

if (FindSymbol(square, text[i], out int columnIndex, out int rowIndex))

{

var newRowIndex = rowIndex == 0

? square.GetUpperBound(1)

: rowIndex - 1;

outputText += square[newRowIndex, columnIndex].ToString();

}

}

break;

case Method.Method2:

var coordinates = new int[m \* 2];

var j = 0;

for (int i = 0; i < m; i++)

{

if (FindSymbol(square, text[i], out int columnIndex, out int rowIndex))

{

coordinates[j] = columnIndex;

coordinates[j + 1] = rowIndex;

j += 2;

}

}

for (int i = 0; i < m; i++)

{

outputText += square[coordinates[i + m], coordinates[i]];

}

break;

}

return outputText;

}

}

class Program

{

static void Main(string[] args)

{

var polybius = new PolybiusSquare();

Console.Write("Введите текст: ");

var message = Console.ReadLine().ToUpper();

Console.Write("Введите пароль(без повторов букв): ");

var pass = Console.ReadLine().ToUpper();

var cipherText = polybius.PolibiusEncrypt(message, pass);

Console.WriteLine("Зашифрованный текст: {0}", cipherText);

Console.WriteLine("Расшифрованный текст: {0}",

polybius.PolybiusDecrypt(cipherText, pass));

Console.ReadLine();

}

}