Лабораторная работа № 2

Тема: исследование и программирование классических методов симметричных криптосистем.

Цель: изучить теоретические основы шифров перестановки и замены, составить программу, позволяющую закодировать текст и выполнить обратное преобразование.

1 Теоретический вопрос

При шифровании шифром двойной перестановки в таблицу по определённому маршруту записывается текст, затем переставляются столбцы и строки. Далее по определённому маршруту выписывается шифрограмма.

Ключом к шифру являются размер таблицы, маршруты вписывания и выписывания, порядки перестановки столбцов и строк. Если маршруты являются фиксированными величинами, то количество ключей равно n ! m ! {\displaystyle n!m!} n!m!, где n {\displaystyle n} n и m {\displaystyle m} m — количество строк и столбцов в таблице.

2 Блок схемы

На рисунке 1 представлена блок схема шифрования и дешифрования сообщения.



Рисунок 1 – Блок схема шифрования и дешифрования

3 Описание работы программы

На рисунке 2 представлен скриншот интерфейса разработанной программы.

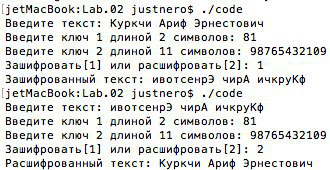


Рисунок 2 – Интерфейс программы

4 Текст программы

package main

import (

"bufio"

"fmt"

"math"

"os"

"strings"

)

func readKey(reader \*bufio.Reader, i, size int) []int {

fmt.Printf("Введите ключ %d длиной %d символов: ", i, size)

value, \_ := reader.ReadString('\n')

for len([]rune(strings.TrimSpace(value))) != size {

fmt.Printf("Убедитесь, что длина ключа %d символов: ", size)

tmp, \_ := reader.ReadString('\n')

value = tmp

}

runes := []rune(strings.TrimSpace(value))

min := 0

result := make([]int, size)

for k := 0; k < size; k = k + 1 {

index := 0

for j := 0; j < size; j = j + 1 {

if int(runes[j]) >= min && (int(runes[index]) < min || runes[j] < runes[index]) {

index = j

}

}

result[k] = index

min = int(runes[index])

runes[index] = rune(0)

}

return result

}

func main() {

reader := bufio.NewReader(os.Stdin)

fmt.Print("Введите текст: ")

text, \_ := reader.ReadString('\n')

input := []rune(strings.TrimSpace(text))

size := float64(len(input))

sizeX := 1

sizeY := int(size)

for i := float64(2); i <= math.Sqrt(size) && float64(sizeY) > i; i = i + 1 {

if math.Abs(math.Floor(size/i)-size/i) < 1e-9 {

sizeX = int(i)

sizeY = int(size) / sizeX

}

}

key1, key2 := readKey(reader, 1, sizeX), readKey(reader, 2, sizeY)

fmt.Print("Зашифровать[1] или расшифровать[2]: ")

direction, \_ := reader.ReadString('\n')

for strings.TrimSpace(direction) != "1" && strings.TrimSpace(direction) != "2" {

fmt.Print("Возможные вариант [1] или [2]: ")

tmp, \_ := reader.ReadString('\n')

direction = tmp

}

runes := make([]rune, int(size))

for i := 0; i < int(size); i = i + 1 {

if strings.TrimSpace(direction) == "1" {

runes[(key1[i/sizeY])\*sizeY+key2[i%sizeY]] = input[i]

} else {

runes[i] = input[(key1[i/sizeY])\*sizeY+key2[i%sizeY]]

}

}

if strings.TrimSpace(direction) == "1" {

fmt.Printf("Зашифрованный текст: %s\n", string(runes))

} else {

fmt.Printf("Расшифрованный текст: %s\n", string(runes))

}

}

Вывод

В ходе выполнения лабораторной работы были изучены теоретические основы шифров перестановки и замены, составлена программа, позволяющая закодировать текст, а также выполнить обратное преобразование. Был изучен шифр двойной перестановки.