Отчет по лабораторной работе номер 2

По предмету мат. основы защиты информации

Дидусь Кирилл Валерьевич

Содержание

# 1 Цель работы

* Изучение шрифров перестановки

# 2 Задание

* Программно реализовать шифр виженера
* Программно реализовать шифр вертикальной перестановки

# 3 Теоретическое введение

Шифр перестано́вки — это метод симметричного шифрования, в котором элементы исходного открытого текста меняют местами. Элементами текста могут быть отдельные символы (самый распространённый случай), пары букв, тройки букв, комбинирование этих случаев и так далее. Типичными примерами перестановки являются анаграммы. В классической криптографии шифры перестановки можно разделить на два класса:

* Шифры одинарной (простой) перестановки — при шифровании символы открытого текста перемещаются с исходных позиций в новые один раз.
* Шифры множественной (сложной) перестановки — при шифровании символы открытого текста перемещаются с исходных позиций в новые несколько раз.

Широкое распространение получила разновидность маршрутной перестановки — вертикальная перестановка. В этом шифре также используется прямоугольная таблица, в которую сообщение записывается по строкам слева направо. Выписывается шифрограмма по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом.

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал Джовани Баттиста Белласо (итал. Giovan Battista Bellaso) в книге La cifra del. Sig. Giovan Battista Bellasо в 1553 году, однако в XIX веке получил имя Блеза Виженера, французского дипломата. Метод прост для понимания и реализации, но является недоступным для простых методов криптоанализа.

Хотя шифр легко понять и реализовать, на протяжении трех столетий он противостоял всем попыткам его сломать; чем и заработал имя le chiffre indéchiffrable (фр. неразгаданный шифр). Многие люди пытались реализовать схемы шифрования, которые по сути являлись шифрами Виженера.

# 4 Выполнение лабораторной работы

Был разработ код для программной реализации шифров из теории лабораторной работы.

1. Шифр Виженера (рис. 1)



Рис. 1: Шифр Виженера

1. Шифр вертикальлной перестановки (рис. 2)

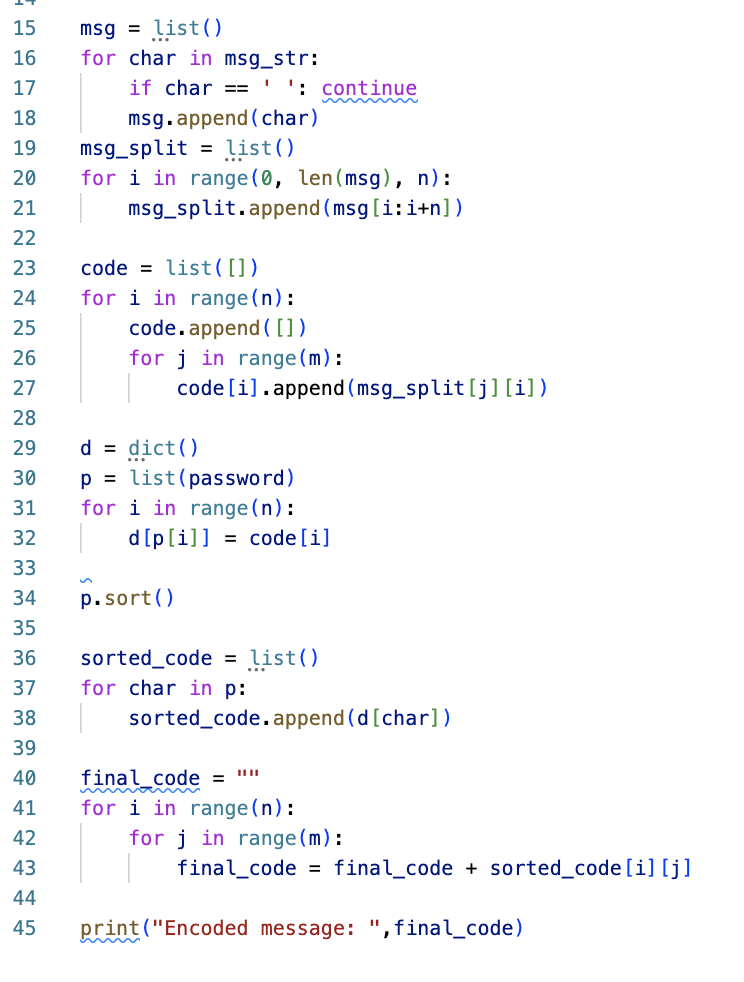


Рис. 2: Шифр вертикальлной перестановки

1. Шифрование с помощью решеток (рис. 3)



Рис. 3: Шифр вертикальлной перестановки

# 5 Выводы

Мы изучили шифры перестановки.

# 6 Листинг программ

## 6.1 шифр Виженера

# Vigenere  
  
def generateKey(string, key):   
 key = list(key)   
 if len(string) == len(key):   
 return(key)   
 else:   
 for i in range(len(string) -len(key)):   
 key.append(key[i % len(key)])   
 return("" . join(key))   
   
def encryption(string, key):   
 encrypt\_text = []   
 for i in range(len(string)):   
 x = (ord(string[i]) +ord(key[i])) % 26  
 x += ord('A')   
 encrypt\_text.append(chr(x))   
 return("" . join(encrypt\_text))   
  
if \_\_name\_\_ == "\_\_main\_\_":   
 string = input("Enter the message: ")  
 keyword = input("Enter the keyword: ")  
 key = generateKey(string, keyword)   
 encrypt\_text = encryption(string,key)   
 print("Encrypted message:", encrypt\_text)

## 6.2 шифрование с помощью решеток

from distutils.file\_util import move\_file  
from email import message  
from operator import index  
import numpy as np  
import random  
  
# Function to rotate the matrix  
# degree clockwise  
from contextlib import nullcontext  
  
  
def rotate\_clockwise(M,n):  
 #print("starting rotate\_clockwise...")  
 m\_r = M.copy()  
 for i in range(0,n):  
 N = len(m\_r[0])  
 for i in range(N // 2):  
 for j in range(i, N - i - 1):  
 temp = m\_r[i][j]  
 m\_r[i][j] = m\_r[N - 1 - j][i]  
 m\_r[N - 1 - j][i] = m\_r[N - 1 - i][N - 1 - j]  
 m\_r[N - 1 - i][N - 1 - j] = m\_r[j][N - 1 - i]  
 m\_r[j][N - 1 - i] = temp  
 return m\_r  
  
def find\_net(k,m):  
 #print("starting find\_net...")  
 m\_net = m.copy()  
 rand\_index = random.randint(0,3)  
 index\_arr = np.array([],dtype=np.int64)  
 for n in range(1,k\*\*2+1):  
 flag = 0  
 occurance = 0  
 for i in range(len(m\_net[0])):  
 for j in range(len(m\_net[0])):  
 if (m\_net[i,j] == n):  
 if (occurance == rand\_index):  
 index\_arr = np.append([i,j],index\_arr)  
 flag = 1  
 break  
 occurance +=1  
 if flag == 1:  
 break  
 index\_arr = index\_arr.reshape(k\*\*2,2)  
 index\_arr = index\_arr[np.lexsort(index\_arr.T[::-1])]  
 return index\_arr  
  
def rotate\_index(index\_arr,k):  
 #print("starting rotate\_index...")  
 arr = index\_arr.copy()  
 arr = (np.flip(arr))  
 new\_index = np.array([],dtype=np.int64)  
 N = k\*\*2  
 for n in range(k\*\*2):  
 j,i = arr[n]  
 temp = i  
 i = j  
 j = N - temp - 1  
 new\_index = np.append(new\_index,[i,j])  
 new\_index = new\_index.reshape(k\*\*2,2)  
 new\_index = new\_index[np.lexsort(new\_index.T[::-1])]  
 return new\_index  
  
def encrypt(k,msg):  
 print("encrypting...")  
 m = np.arange(1,(k\*\*2)+1)  
 m = m.reshape(k,k)  
 m\_upper = np.hstack((m,rotate\_clockwise(m,1)))  
 m\_bottom = np.hstack((rotate\_clockwise(m,3),rotate\_clockwise(m,2)))  
 m\_final = np.vstack((m\_upper,m\_bottom))  
 index\_arr = find\_net(k,m\_final)  
   
 msg = list(msg)  
   
 arr = [[]]  
 for q in range(k\*\*2-1):  
 arr.append([])  
  
 count = 0  
 for char in msg:  
 if char == ' ':  
 continue  
 if count == k\*\*2:  
 index\_arr = rotate\_index(index\_arr,k)  
 count = 0  
 i,j = index\_arr[count]  
 count += 1  
 arr[i].insert(j,char)  
 final\_code = ""  
 for i in range(len(arr)):  
 for j in range(len(arr)):  
 final\_code = final\_code + arr[i][j]  
 print("Encoded message: ",final\_code)  
   
  
msg = input("введите сообщение:", )  
size = 2  
count = 0  
for char in msg:  
 if char == ' ':  
 continue  
 count+=1  
while(count > ((2\*size)\*(2\*size))):  
 size+=1  
  
encrypt(size,msg)

## 6.3 шифр вертикальной перестановки

from numpy import sort  
  
  
msg\_str = "договор подписали"  
m = 4 #кол-во блоков  
n = 4 # длина блоков  
password = "шифр"  
def get\_input():  
 msg\_str = input("введите сообщение:", )  
 m = input("введите кол-во блоков:",)  
 n = input("введите длину блоков:",)  
 return(msg\_str,m,n)  
  
msg = list()  
for char in msg\_str:  
 if char == ' ': continue  
 msg.append(char)  
msg\_split = list()  
for i in range(0, len(msg), n):  
 msg\_split.append(msg[i:i+n])  
  
code = list([])  
for i in range(n):  
 code.append([])  
 for j in range(m):  
 code[i].append(msg\_split[j][i])  
  
d = dict()  
p = list(password)  
for i in range(n):  
 d[p[i]] = code[i]  
   
p.sort()  
  
sorted\_code = list()  
for char in p:  
 sorted\_code.append(d[char])  
  
final\_code = ""  
for i in range(n):  
 for j in range(m):  
 final\_code = final\_code + sorted\_code[i][j]  
  
print("Encoded message: ",final\_code)

# Список литературы

* ТУИС РУДН
* Википедия