Отчет по лабораторной работе номер 2

По предмету мат. основы защиты информации

Дидусь Кирилл Валерьевич

Содержание

Список иллюстраций

Список таблиц

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

Bellaso 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. 4.1)

```python
# 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)

```

Рис. 4.1: Шифр Виженера

2. Шифр вертикальлной перестановки (рис. 4.2)

```
15  msg = list()
16  for char in msg_str:
17      if char == ' ': continue
18      msg.append(char)
19  msg_split = list()
20  for i in range(0, len(msg), n):
21      msg_split.append(msg[i:i+n])
22
23  code = list([])
24  for i in range(n):
25      code.append([])
26      for j in range(m):
27          code[i].append(msg_split[j][i])
28
29  d = dict()
30  p = list(password)
31  for i in range(n):
32      d[p[i]] = code[i]
33
34  p.sort()
35
36  sorted_code = list()
37  for char in p:
38      sorted_code.append(d[char])
39
40  final_code = ""
41  for i in range(n):
42      for j in range(m):
43          final_code = final_code + sorted_code[i][j]
44
45  print("Encoded message: ",final_code)
```

Рис. 4.2: Шифр вертикальлной перестановки

2. Шифрование с помощью решеток (рис. 4.3)

```
64  def encrypt(k,msg):
65      print("encrypting...")
66      m = np.arange(1,(k**2)+1)
67      m = m.reshape(k,k)
68      m_upper = np.hstack((m,rotate_clockwise(m,1)))
69      m_bottom = np.hstack((rotate_clockwise(m,3),rotate_clockwise(m,2)))
70      m_final = np.vstack((m_upper,m_bottom))
71      index_arr = find_net(k,m_final)
72
73      msg = list(msg)
74
75      arr = [[]]
76      for g in range(k**2-1):
77          arr.append([])
78
79      count = 0
80      for char in msg:
81          if char == ' ':
82              continue
83          if count == k**2:
84              index_arr = rotate_index(index_arr,k)
85              count = 0
86          i,j = index_arr[count]
87          count += 1
88          arr[i].insert(j,char)
89      final_code = ""
90      for i in range(len(arr)):
91          for j in range(len(arr)):
92              final_code = final_code + arr[i][j]
93      print("Encoded message: ",final_code)
94
95
```

Рис. 4.3: 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