Вторая лабораторная работа. Шифры перестановки

НФИмд-01-23

Асеинова Елизавета Валерьевна

Содержание

Список таблиц

Список иллюстраций

1. Цель работы

Цель данной работы - ознакомиться с шифрами перестановки, а также научиться применять их на практике.

2. Задание

1. Реализовать маршрутное шифрование
2. Реализовать шифрование с помощью решеток
3. Реализовать шифрование с использованием таблицы Вижинера

3. Теоретическое введение

Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов. Способ, каким при шифровании переставляются буквы открытого текста и является ключом шифра. Важным требования является равенство длин ключа исходного текста.

4. Выполнение лабораторной работы

1. Произведено ознакомление с шифрами перестановки по методическим материалам курса

2. Прописан код для маршрутного шифрования на языке программирования Python.

```
alphabet = 'абвгдеёжзиклмнопрстуфхцчшщъыьэюя'
def shifr(text, key, m, n):
  global alphabet
  text_n = text.replace(' ','')
  if len(text_n)< (m*n):
    text_n += alphabet[:m*n - len(text_n)]
  ch = iter(text_n)
  matr = [[next(ch) for y in range(m)]for x in range(n)]
  passw = [alphabet.index(x) for x in key]
  passw_sort = sorted(passw)
  result = ''
  for char in passw_sort:
    for x in range(n):
      result += matr[x][passw.index(char)]
  return result
```

Рис. 4.1.: Маршрутное шифрование

3. Выводим на экран результат применения.

```
[8] print(shifr('нельзя недооценивать противника', 'пароль', 6, 5))

    еенпнзоатаьовокннеьвлдирияцтиа
```

Рис. 4.2.: Результат применения 1

4. Прописан код для шифрования с помощью решеток на языке программирования Python.

```
[10] import numpy as np
     k = 2
     k_2 = [x+1 for x in range(k**2)]
     matr = [[0 for x in range(2*k)] for y in range(2*k)]
     matr = np.array(matr)
     for x in range(k**2):
       c=0
       for x in range(k):
         for y in range(k):
           matr[x][y] = k_2[c]
           c+=1
       matr = np.rot90(matr)
     mv ={ k: 0 for k in k_2}
     mv_2 = {1:2, 2:4, 3:3, 4:3}
     for x in range(k**2):
       for y in range(k**2):
         mv[matr[x][y]]+=1
         if mv[matr[x][y]]!= mv_2[matr[x][y]]:
           matr[x][y] = -1
         else:
           matr[x][y] = 0
```

Рис. 4.3.: Шифрование с помощью решеток

5. Выводим на экран результат применения.

```
[13] text = 'договорподписали'
     key = 'шифр'

     ct = 0
     t = iter(text)
     matr2 = [['0' for y in range(k**2)] for x in range(k**2)]
     for v in range(4):
       for x in range(k**2):
         for y in range(k**2):
           if matr[x][y]==0:
             matr2[x][y] = text[ct]
             ct+=1
       matr = np.rot90(matr, -1)
     passw = [alphabet.index(x) for x in key]
     passw_sort = sorted(passw)
     result = ''
     for char in passw_sort:
       for x in range(k**2):
         result+=matr2[x][passw.index(char)]
     print(result)

     овордлгпаписдои
```

Рис. 4.4.: Результат применения 2

6. Прописан код для шифрования с использованием таблицы Вижинера на

языке программирования Python.

```python
def make_key(m, key):
  key.replace(' ', '')
  m.replace(' ', '')
  key = list(key)
  if len(m) == len(key):
    return(key)
  else:
    for i in range(len(m) - len(key)):
      key.append(key[i%len(key)])
  return(''.join(key))

def vigion(m, key):
  v = []
  m.replace(' ', '')
  for i in range(len(m)):
    x = (ord(m[i]) + ord(key[i])) % 26
    x += ord('A')
    v.append(chr(x))
  return (''.join(v))
```

Рис. 4.5.: Таблица Вижинера

7. Выводим на экран результат применения.

```python
m = 'cryptography is a serious science'
key = 'math'
print(vigion(m, make_key(m, key)))
```

```
ADDIRALKYBMRLUXGYZXXPUTNQZXVGQSVC
```

Рис. 4.6.: Результат применения 3

5. Выводы

В рамках данной лабораторной работы было произведено ознакомление с шифрами перестановки. Шифры были реализованы на языке программирования Python.

6. Список литературы

1. Методические материалы курса