Отчёт по лабораторной работе №2

Шифр простой замены

Студент: Гонсалес Ананина Луис Антонио, 1032175329
Группа: НФИмд-02-21
Преподаватель: Кулябов Дмитрий Сергеевич,
д-р.ф.-м.н., проф.

Москва 2021

Содержание

List of Figures

1 Цель работы

Цель данной лабораторной работы- изучить теорию и реализовать рассмотренные шифры програмно.

2 Теоретические сведения

Шифры перестановки

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 [1].

Маршрутное шифрование

Пусть m и n – некоторые натуральные (т.е. целые положительные) числа, каждое больше 1. Открытый текст последовательно разбивается на части (блоки) с длиной, равной 6 произведений mn (если в последнем блоке не хватает букв, можно дописать до нужной длины произвольный их набор). Блок вписывается построчно в таблицу размерности m×n (т.е. m строк и n столбцов). Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Этот маршрут вместе с числами m и n составляет ключ шифра.

Чаще всего буквы выписывают по столбцам, которые упорядочиваются в соответствии с паролем: под таблицей подписывается слово, состоящее из n неповторяющихся букв, и столбцы таблицы нумеруются по алфавитному порядку букв пароля. Например, для шифрования открытого текста, выражающего один из главных принципов криптологии: нельзя недооценивать противника, добавим к его 29 буквам еще одну, скажем а, возьмем m=5, n=6, впишем текст в таблицу 5×6 и выберем в качестве пароля слово п а р о л ь:

нельзя недооц ениват ьпроти вникаа пароль

Выписывая теперь буквы по столбцам в соответствии с алфавитным порядком

букв в пароле, получаем следующую криптограмму: ЕЕНПНЗОАТАЬОВОКННЕЬ-ВЛДИРИЯЦТИА (истинные пробелы в криптографии не выставляются) [2].

Шифрование с помощью решеток

Выбирается натуральное число k > 1, и квадрат размерности k×k построчно заполняется числами 1, 2, …, k. Для примера возьмем k = 2.

Квадрат поворачивается по часовой стрелке на 90° и размещается вплотную к предыдущему квадрату. Аналогичные действия совершаются еще два раза, так чтобы в результате из четырех малых квадратов образовался один большой с длиной стороны 2k.

Далее из большого квадрата вырезаются клетки с числами от 1 до k2, для каждого числа одна клетка. Процесс шифрования происходит следующим образом. Сделанная решетка (квадрат с прорезями) накладывается на чистый квадрат 2k×2k и в прорези по строчкам (т.е. слева направо и сверху вниз) вписываются первые буквы открытого текста. Затем решетка поворачивается на 90° по часовой стрелке и накладывается на частично заполненный квадрат, вписывание продолжается.

После третьего поворота, наложения и вписывания все клетки квадрата будут заполнены. Правило выбора прорезей гарантирует, что при заполнении квадрата буква на букву никогда не попадет. Из заполненного квадрата буквы можно выписать по столбцам, выбрав подходящий пароль. Например, с использованием изображенной выше решетки и пароля ш и ф р открытый текст договор подписали переводится в криптограмму за пять шагов:

Итоговая криптограмма: ОВОРДЛГПАПИОСДОИ.

Таблица Виженера

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколь-

ко позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет такой вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

Первый символ исходного текста («A») зашифрован последовательностью L, которая является первым символом ключа. Первый символ зашифрованного текста («L») находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ зашифрованного текста («X») получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом [3].

Исходный текст: ATTACKATDAWN Ключ: LEMONLEMONLE Зашифрованный текст: LXFOPVEFRNHR

3 Выполнение работы

In [1]:
```
import numpy as np
import sys
```

In [2]:
```
def matr(lists):
    for i in lists:
        for j in i:
            print(j,end=" ")
        print()
```

In [5]:
```
#1. Маршрутное шифрование
def marshifr():
    text=input("Text: ").replace(' ','')
    n=int(input("n= "))
    m=int(input("m= "))
    password=input("Password: ")
    lists=[['a' for i in range(0,n)]for j in range(m)]
    it=0
    for i in range(m):
        for j in range(n):
            if it < len(text):
                lists[i][j]=text[it]
                it+=1
    lis=list()
    for i in range(n):
        lis.append(password[i])
    lists.append(lis)
    print(lists)
    result=""
    sortpass=sorted(lists[len(lists)-1])
    for i in sortpass:
        print(i," = ",lists[len(lists)-1].index(i))
        for j in range(len(lists)):
            if j ==len(lists)-1:
                continue
            result +=lists[j][lists[len(lists)-1].index(i)]
    print(result)
```

In [7]:
```
marshifr()
```

```
Text: недооценивать противника
n= 6
m= 5
Password: пароль
[['н', 'е', 'д', 'о', 'о', 'ц'], ['е', 'н', 'и', 'в', 'а', 'т'], ['ь', 'п', 'р', 'о', 'т', 'и'], ['в', 'н', 'и', 'к', 'а',
'а'], ['а', 'а', 'а', 'а', 'а', 'а'], ['п', 'а', 'р', 'о', 'л', 'ь']]
а  =  1
л  =  4
о  =  3
п  =  0
р  =  2
ь  =  5
енпнаоатааовоканеьвадириацтиаа
```

Figure 3.1: Lab02

In [11]:
```
#2. Шифрование с помощью решеток
def rotate90(matri):
    return[list(reversed(col))for col in zip(*matri)]
def deleting(largelist,inn,k):
    for i in range(k*2):
        for j in range(k*2):
            if largelist[i][j]==inn:
                largelist[i][j]= " "
                return

def reshotki():
    k=int(input("k= "))
    s=1
    lists=[[i for i in range(k)]for i in range(k)]
    for i in range(k):
        for j in range(k):
            lists[i][j] = s
            s+=1
    print(lists)
    lists1=rotate90(lists)
    lists2=rotate90(lists1)
    lists3=rotate90(lists2)
    largelist=[[1 for i in range(2*k)]for i in range(2*k)]
    for i in range(k):
        for j in range(k):
            largelist[i][j]=lists[i][j]
    i1=0
    j1=0
    for i in range(0,k):
        for j in range(k,k*2):
            largelist[i][j]=lists1[i1][j1]
            j1+=1
        j1=0
        i1+=1
    i1=0
    j1=0
    for i in range(k,k*2):
        for j in range(k,k*2):
            largelist[i][j]=lists2[i1][j1]
            j1+=1
        j1=0
        i1+=1
    i1=0
    j1=0
    for i in range(k,k*2):
        for j in range(k,k*2):
            largelist[i][j]=lists3[i1][j1]
            j1+=1
        j1=0
        i1+=1
    matr(largelist)
    text="договорподписали"
    largelist_a=[[" "for i in range(2*k)]for i in range(2*k)]
    s=0
    li=[i for i in range(1,k**2+1)]
    for inn in li:
        deleting(largelist,inn,k)
    ind=0
    for i in range(k*2):
        for j in range(k*2):
            if largelist[i][j]==largelist_a[i][j] and len(text) >0:
                largelist_a[i][j]=text[0]
                text=text[1:]
    largelist=rotate90(largelist)
    for i in range(k*2):
        for j in range(k*2):
```

Figure 3.2: Lab02

```
    for inn in li:
        deleting(largelist,inn,k)
    ind=0
    for i in range(k*2):
        for j in range(k*2):
            if largelist[i][j]==largelist_a[i][j] and len(text) >0:
                largelist_a[i][j]=text[0]
                text=text[1:]
    largelist=rotate90(largelist)
    for i in range(k*2):
        for j in range(k*2):
            if largelist[i][j]==largelist_a[i][j] and len(text) >0:
                largelist_a[i][j]=text[0]
                text=text[1:]
    if len(text)>0:
        largelist=rotate90(largelist)
        for i in range(k*2):
            for j in range(k*2):
                if largelist[i][j]==largelist_a[i][j] and len(text) >0:
                    largelist_a[i][j]=text[0]
                    text=text[1:]
    if len(text)>0:
        largelist=rotate90(largelist)
        for i in range(k*2):
            for j in range(k*2):
                if largelist[i][j]==largelist_a[i][j] and len(text) >0:
                    largelist_a[i][j]=text[0]
                    text=text[1:]
    matr(largelist_a)
    stri=input("Password: ")
    if len(stri)> k * 2:
        stri=stri[:k*2]
    elif len(stri) < k*2:
        while len(stri) != k*2:
            stri +="z"
    largelist_a.append(list(stri))
    matr(largelist_a)
    result=""
    spisok= sorted(largelist_a[len(largelist_a)-1])
    for i in spisok:
        print(i, " = ",largelist_a[len(largelist_a)-1].index(i))
        for j in range(len(largelist_a)):
            if j ==len(largelist_a)-1:
                continue
            result += largelist_a[j][largelist_a[len(largelist_a)-1].index(i)]
    print(result.replace(" ",""))
```

In [13]:
```
reshotki()
```

```
k= 4
[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]
1 2 3 4 13 9 5 1
5 6 7 8 14 10 6 2
9 10 11 12 15 11 7 3
13 14 15 16 16 12 8 4
1 1 1 1 4 8 12 16
1 1 1 1 3 7 11 15
1 1 1 1 2 6 10 14
1 1 1 1 1 5 9 13
д о г о в о р
  п о д п и
    с а л
      и
```

Figure 3.3: Lab02

```
Password: шифр
д о г о в о р
  п о д п и
    с а л
      и



ш и ф р z z z z
z  =  4
z  =  4
z  =  4
z  =  4
и  =  1
р  =  3
ф  =  2
ш  =  0
вплвплвплвплоподаигосд
```

In [22]:
```
#3 Таблица Виженера
alpha=[chr(i) for i in range(1072, 1072+32)]
alpha
```

Out[22]:
```
['а',
 'б',
 'в',
 'г',
 'д',
 'е',
 'ж',
 'з',
 'и',
 'й',
 'к',
 'л',
 'м',
 'н',
 'о',
 'п',
 'р',
 'с',
 'т',
 'у',
 'ф',
 'х',
 'ц',
 'ч',
 'ш',
 'щ',
 'ъ',
 'ы',
 'ь',
 'э',
 'ю',
 'я']
```

In [24]:
```
alpha.remove('ъ')
```

In [25]:
```
len(alpha)
```

Out[25]:
```
31
```

Figure 3.4: Lab02

```
In [24]: alpha.remove('ь')

In [25]: len(alpha)

Out[25]: 31

In [26]: newalpha=[]
         for i in range(len(alpha)):
             current=alpha[i:]+alpha[:i]
             newalpha.append(current)
         newalpha=np.array(newalpha)

In [27]: message= 'криптография серьезная наука'.replace(' ','')
         message

Out[27]: 'криптографиясерьезнаянаука'

In [28]: password= 'математика'

In [29]: while len(password)<len(message):
             password+=password
         password=password[:len(message)]
         password

Out[29]: 'математикаматематикаматема'

In [30]: result=''
         for i,j in zip(message,password):
             x=[idx for idx,k in enumerate(newalpha[0,:])if k==i][0]
             y=[idx for idx,k in enumerate(newalpha[:,0])if k==j][0]
             result+=newalpha[x,y]

In [31]: result

Out[31]: 'црыфяохшкффядкэьчпчалнтшца'

In [ ]:
```

Figure 3.5: Lab02

4 Выводы

В итоге в данной лабораторной работы я изучил теорию и реализовал рассмотренные шифры програмно.

Список литературы

1. Шифры перестановки [Электронный ресурс]. Википедия, 2021. URL: https://ru.wikipedia.org/wiki/Перестановочный_шифр.
2. Шифры [Электронный ресурс]. Википедия, 2021. URL: https://it.rfei.ru/course/~k017/~7mdCpor7/~c5kOtaHY.
3. Таблица Виженера [Электронный ресурс]. Википедия, 2021. URL: https://ru.wikipedia.org/wiki/Шифр_Виженера#Описание.