

Лабораторная № 1

Шифры перестановки

Баранов Иван Юрьевич

2022 Москва

RUDN University, Moscow, Russian Federation

Содержание

1) Цель работы.....	3
2) Ход работы	4
2.1) Описание сущности шифра перестановки.....	5
3) Описание реализации.....	6
3.1) Маршрутное шифрование	6
3.2)Шифрование с помощью решеток	7
3.3) Шифр Виженера.....	8
4) Вывод.....	9

ЦЕЛЬ РАБОТЫ

- Ознакомиться с шифрами перестановки;
- Реализовать маршрутное шифрование;
- Реализовать шифрование с помощью решеток;
- Реализовать шифр Виженера.

Ход работы

- Для реализации алгоритмов использовались средства языка
- Python.
- Были реализованы шифраторы, рассматриваемых алгоритмов.

Описание сущности шифра перестановки

Шифр 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. 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.

Описание реализации

1 Задача – Маршрутное шифрование

1 рис.

1 задача

```
In [198]: text = 'нельзя недооценивать противника'
text_1 = 'пароль'

def columns_replace(text, text_1):
    text = text.replace(' ', '')
    size = len(text_1)
    if len(text) % size != 0:
        text += 'a'

    def chunks(lst, n):
        for i in range(0, len(lst), n):
            yield lst[i:i + n]

    text = list(text)
    text = chunks(text, 6)
    text = list(text)
    text_1 = list(text_1)
    text.append(text_1)
    arr = np.array(text)
    arr = arr.T
    dict_ = {}
    for i in arr:
        key = i[-1]
        dict_[key] = list(i[:-1])
    od = collections.OrderedDict(sorted(dict_.items()))
    final_text = ''
    for i in od.values():
        char = ''.join(i)
        final_text += char

    return final_text
fin = columns_replace(text, text_1)
fin

Out[198]: 'еенпнзоаъаьовкнньевлдирияцтиа'
```

2 Задача - шифрование с помощью решеток

2 рис.

```
In [194]: def get_matrix():
    cipher_grille = [['.', '.', '.', 'x'],
                     ['.', '.', '.', '.'],
                     ['.', 'x', '.', 'x'],
                     ['.', '.', 'x', '.']]

    ciphered_password = 'договор подписали'
    ciphered_password = ciphered_password.replace(' ', '').lower()
    ciphers = []
    ciphers.append(cipher_grille)
    for i in range(3):
        cipher_grille = [[cipher_grille[i][j] for i in reversed(range(4))] for j in range(4)]
        ciphers.append(cipher_grille)
    ciphers
    out = ''
    s = []
    count = 0
    for cipher in ciphers:
        s_1 = []
        for i, item in enumerate(cipher):
            if 'x' in item:
                index = item.index('x')
                temp = item
                temp[index] = ciphered_password[count]
                count += 1
            if 'x' in temp:
                index_ = temp.index('x')
                temp[index_] = ciphered_password[count]
                count += 1
            s_1.append(temp)
        else:
            s_1.append(temp)
        else:
            s_1.append(item)
        s.append(s_1)

    return s
```

3 рис.

```
def get_char(num):
    text = ''
    bag = []
    count_ = 0
    dict_ = {}
    s = get_matrix()
    for i in s:
        work = i[num]
        count_ += 1
        for j in work:
            if j.isalpha():
                index = work.index(j)
                dict_[index] = j
    od = collections.OrderedDict(sorted(dict_.items()))
    od = [i[1] for i in od.items()]
    od = ''.join(od)
    text += od
    return text

def get_all_text():
    all_text = ''
    for num in range(len(s[0])):
        text = get_char(num)
        all_text += text
    return all_text

def chunks(lst, n):
    for i in range(0, len(lst), n):
        yield lst[i:i + n]
```

4 рис.

```
In [196]: def encoding():
    all_text = get_all_text()
    all_text = list(all_text)
    all_list = list(chunks(all_text, len(s[0])))
    password = list('шифр')
    all_list.append(password)
    arr = np.array(all_list)
    arr = arr.T
    dict_ = {}
    for i in arr:
        key = i[-1]
        dict_[key] = list(i[:-1])
    od = collections.OrderedDict(sorted(dict_.items()))
    final_text = ''
    for i in od.values():
        char = ''.join(i)
        final_text += char
    return final_text
```

```
In [197]: final_text = encoding()
final_text
```

```
Out[197]: 'овордлгпапиосдои'
```

3 задача – шифр Виженера

5 рис.

3 задача

```
In [193]: from itertools import cycle

alp = 'абвгдеёжзийклмнопрстуфхцчщъыьэюя'

def encode_vijn(text, keytext):
    text = text.replace(' ', '').lower()
    keytext = keytext.lower().replace(' ', '')
    f = lambda arg: alp[(alp.index(arg[0]) + alp.index(arg[1])) % 33]
    return ''.join(map(f, zip(text, cycle(keytext))))

def decode_vijn(coded_text, keytext):
    f = lambda arg: alp[(alp.index(arg[0]) - alp.index(arg[1])) % 33]
    return ''.join(map(f, zip(coded_text, cycle(keytext))))

text = 'клад зарыт в саду'
keytext = 'зима'
sd = encode_vijn(text, keytext)
sd
```

```
Out[193]: 'тфидлизыькюаль'
```

Вывод

- Ознакомились с шифрами перестановки;
- Реализовали маршрутное шифрование;
- Реализовали шифрование с помощью решеток;
- Реализовать шифр Виженера.