Лабораторная работа №2

Шифры перестановки

Доборщук В.В.

1 октября 2022

Российский университет дружбы народов, Москва, Россия



Информация

Докладчик

- Доборщук Владимир Владимирович
- студент группы НФИмд-02-22, студ. билет 1132223451
- учебный ассистент кафедры прикладной информатики и теории вероятностей
- Российский университет дружбы народов
- · 1132223451@rudn.ru



<u> Цели и задачи</u>

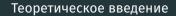
Цели и задачи

Цель работы — изучить и программно реализовать шифры перестановки.

Задачами являются:

• Реализовать все описанные в лабораторной работе шифры.

Теоретическое введение



Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов.

Выполнение лабораторной работы

Выполнение лабораторной работы

Для реализации шифров мы будем использовать Python, так как его синтаксис позволяет быстро реализовать необходимые нам алгоритмы.

```
In [1]: import numpy as np
In [2]: def get_alphabet(option="english"):
    if option == "english":
        return list(map(chr, range(ord("a"), ord("z")+1)))
    elif option == "russlan":
        return list(map(chr, range(ord("a"), ord("x")+1)))
```

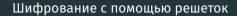
Рис. 1: Библиотеки и дополнительные функции

Также реализовали функции получения алфавитов (английского и русского).

Маршрутное шифрование

Маршрутное шифрование реализовали в соответствии с описанной в лабораторной работе процедурой. Успешно протестировали на приведенном в работе отрывке.

```
In [3]: def marchroute cipher(message: str. key: str):
            alphabet russian = get alphabet("russian")
            alphabet english = get alphabet()
            columns size = len(kev)
            message cleared = list(filter(lambda s: s.lower() in alphabet russian or s in alphabet english, message))
            message matrix = [
                [letter for letter in message cleared[i:i+columns size]]
                for i in range(0, len(message cleared), columns size)
            if len(message matrix[-1]) < columns size:</pre>
                message matrix[-1] = message matrix[-1] +
                    [message matrix[-1][-1]]*(columns size-len(message matrix[-1]))
            message password dict = { value : np.array(message matrix)[:,k] for k, value in enumerate(list(key)) }
            ciphered message = ''.join([''.join(message_password_dict[k]).upper()
                                        for k in sorted(message password dict.kevs())])
            return ciphered message
In [4]: m test = "нельзя недооценивать противника"
        k test = "пароль"
In [5]: result = marchroute cipher(m test, k test)
        print(f'Pesvльтат шифрования: \
                \n{m test} * [{k test}]\n-> {result, len(result)}')
        Результат шифрования:
        нельзя недооценивать противника * [пароль]
        -> ('ЕЕНПНЗОАТАЬОВОКННЕЬВЛЛИРИЯЦТИА', 30)
```



Данный вид шифрования не удалось реализовать.

Таблица Виженера

Маршрутное шифрование реализовали в соответствии с описанной в лабораторной работе процедурой. Успешно протестировали на приведенном в работе отрывке (с учетом, что русский алфавит немного изменен).

```
In [9]: def vigenere table(message: str, kev: str, differ alphabet=False):
             alphabet_russian = get_alphabet("russian")
             if differ alphabet:
                 alphabet russian.remove('b')
                 alphabet russian[alphabet russian.index('b')] = 'b'
             alphabet english = get alphabet()
             def find letter for pair(letters pair: tuple):
                 if letters pair[0].lower() in alphabet russian:
                     orig letter index = alphabet russian.index(letters pair[1].lower())
                     key letter index = alphabet russian.index(letters pair[0].lower())
                     shift = orig letter index + key letter index
                     if shift > len(alphabet russian):
                         return alphabet russian[shift - len(alphabet russian)]
                     return alphabet russian[shift]
             message cleared = list(filter(lambda s: s.lower() in alphabet russian or s in alphabet english, message))
             row_length = len(message_cleared)
             full key = (list(key) * row length)[:row length]
             message key zip = list(zip(full key, message cleared))
             return ''.join(list(map(find_letter_for_pair, message_key_zip))).upper()
In [10]: m test = "криптография - серьезная наука"
         k test = "математика"
In [11]: result = vigenere table(m test, k test, True)
```



Выводы

В рамках выполненной лабораторной работы мы изучили и реализовали следующие шифры перестановки: маршрутное шифрование и таблицу Виженера. Реализовать шифрование с помощью решеток не удалось.