

Лабораторная работа №2

Шифры перестановки

Доборщук В.В.

1 октября 2022

Российский университет дружбы народов, Москва, Россия

Информация

- 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



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

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 == "russian":
            return list(map(chr, range(ord("а"), ord("я")+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(key)

        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:
            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.keys())])

        return ciphered_message
```

```
In [4]: m_test = "нельзя недооценивать противника"
        k_test = "пароль"
```

```
In [5]: result = marchroute_cipher(m_test, k_test)
        print(f'Результат шифрования: \
              \n{m_test} * [{k_test}]\n-> {result, len(result)}')
```

```
Результат шифрования:
нельзя недооценивать противника * [пароль]
-> ('ЕЕНПНЗОАТЬАОВКННЕНЬВЛДИРИЦТИА', 30)
```

Рис. 2: Маршрутное шифрование и его тестирование

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, key: str, differ_alphabet=False):
        alphabet_russian = get_alphabet("russian")
        if differ_alphabet:
            alphabet_russian.remove('ё')
            alphabet_russian[alphabet_russian.index('ё')] = 'ь'
            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.