Математические основы защиты информации и информационной безопасности. Отчет по лабораторной работе №1

Шифры простой замены

Юдин Герман Станиславович 1132236901

Содержание

List of Figures

List of Tables

1 Цель работы

Освоить на практике шифры простой замены.

2 Выполнение лабораторной работы

Требуется реализовать:

1. Шифр Цезаря с произвольным ключом K.
2. Шифр Атбаш.

2.1 Шифр цезаря

Шифр Цезаря (также он является шифром простой замены) — это моноалфавитная подстановка, т.е. каждой букве открытого текста ставится в соответствие одна буква шифртекста. На практике при создании шифра простойзамены в качестве шифроалфавита берется исходный алфавит, но с нарушеннымпорядком букв (алфавитная перестановка). Для запоминания нового порядкабукв перемешивание алфавита осуществляется с помощью пароля. В качествепароля могут выступать слово или несколько слов с неповторяющимися буквами. Шифровальная таблица состоит из двух строк: в первой записывается стандартный алфавит открытого текста, во второй — начиная с некоторой позицииразмещается пароль (пробелы опускаются), а далее идут в алфавитном порядке оставшиеся буквы, не вошедшие в пароль. В случае несовпадения начала пароля с началом строки процесс после ее завершения циклически продолжается с первой позиции. Ключом шифра служит пароль вместе с числом, указывающим положение начальной буквы пароля.

Чтобы реализовать программу был написал след. код на python:

Выписан алфавит с учетом возможного пробела fig. 2.1.

Figure 2.1: Алфавит

Вводится число, на которое будет произведен сдвиг. Даже если число слишком большое оно преобразуется в возможный интервал (берётся остаток от деления) fig. 2.2.

```
k = int(input("Введите число, на которое хотите сдвинуть шифр: "))

k = k % len(base)
```

Figure 2.2: Ввод k

Вводится контрольное слово, у которого берутся только уникальные символы fig. 2.3.

```
pswd = input("Введите кодовое слово: ").lower()
unique_characters = []

for character in pswd:
    if character not in unique_characters:
        unique_characters.append(character)
```

Figure 2.3: Контрольное слово

После слова добавляется весь оставшийся алфавит fig. 2.4.

```
for character in base:
    if character not in unique_characters:
        unique_characters.append(character)
```

Figure 2.4: Преобразование зашифрованного алфавита 1

Происходит перемещение k символов в начало fig. 2.5.

```
begin = unique_characters[len(unique_characters)-k:]
end = unique_characters[:len(unique_characters)-k]

cypher = begin + end
```

Figure 2.5: Преобразование зашифрованного алфавита 2

Выводятся оба алфавита. А также происходит ввод фразы, которую хотим зашифровать. Шифрование происходит за счет вычиления индекса символа и выбора символа из зашифрованного алфавита по данному символу fig. 2.6.

```
print(base)
print(cypher)

while True:
    string = input("Введите строку, которую хотите зашифровать: ").lower()
    if string == "q":
        break

    result = ""
    for character in string:
        result += cypher[base.index(character)]

    print(result)
```

Figure 2.6: Шифрование фразы

Результат работы алгоритма представлен на рисунке fig. 2.7.

```
Введите число, на которое хотите сдвинуть шифр: 3
Введите кодовое слово: пароль
['а', 'б', 'в', 'г', 'д', 'е', 'ё', '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', ' ']
['ы', 'э', 'ю', 'я', ' ', 'п', 'а', 'р', 'о', 'л', 'ь', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'м', 'н', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ']
Введите строку, которую хотите зашифровать: этот текст должен быть зашифрован шифром цезаря
чиеиьипбзиъ еврпдъэхицьоытлкжеюыдътлкжегънпоыжщ
Введите строку, которую хотите зашифровать: q

Process finished with exit code 0
```

Figure 2.7: Результат работы алгоритма

2.2 Шифр атбаш

Данный шифр является шифром сдвига на всю длину алфавита, состоящего из русских букв и пробела.

Чтобы реализовать программу был написал след. код на python:

Выписан алфавит с учетом возможного пробела fig. 2.8.

```
base = ['а', 'б', 'в', 'г', 'д', 'е', 'ё', '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', ' ']
```

Figure 2.8: Алфавит

Алфавит разворачивается fig. 2.9.

```
cypher = base.copy()
cypher.reverse()
```

Figure 2.9: Алфавит

Выводятся оба алфавита. А также происходит ввод фразы, которую хотим зашифровать. Шифрование происходит за счет вычиления индекса символа и выбора символа из зашифрованного алфавита по данному символу fig. 2.10.

```python
print(base)
print(cypher)

while True:
    string = input("Введите строку, которую хотите зашифровать: ").lower()
    if string == "q":
        break

    result = ""
    for character in string:
        result += cypher[base.index(character)]

    print(result)
```

Figure 2.10: Шифрование фразы

Результат работы алгоритма представлен на рисунке fig. 2.1.

```
['а', 'б', 'в', 'г', 'д', 'е', 'ё', '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', ' ']
[' ', 'я', 'ю', 'э', 'ь', 'ы', 'ъ', 'щ', 'ш', 'ч', 'ц', 'х', 'ф', 'у', 'т', 'с', 'р', 'п', 'о', 'н', 'м', 'л', 'к', 'й', 'и', 'з', 'ж', 'ё', 'е', 'д', 'г', 'в', 'б', 'а']
Введите строку, которую хотите зашифровать: этот текст должен быть зашифрован шифром атбаша
гнснаныхонаьсфщытаяендаш зчлпсю тазчлпсуа ня з
Введите строку, которую хотите зашифровать:
```

Figure 2.11: Результат работы алгоритма

3 Выводы

В результате выполнения работы я освоил на практике применение шифров простой замены.

4 Список литературы

1. Методические материалы курса