Лабораторная работа №1

Шифры простой замены

Дугаева Светлана Анатольевна, НФИмд-02-22

Содержание

Список таблиц

Список иллюстраций

Цель работы

Цель данной работы — изучить и программно реализовать шифры Цезаря и Атбаш.

Задание

Заданием является:

- Реализовать шифр Цезаря с произвольным ключом k;
- Реализовать шифр Атбаш.

Теоретическое введение

Шифр простой замены представляет собой замену каждой буквы в исходном слове на определенное число, которому соответствует данная буква [@золотин74криптографические]. В основе функционирования шифров простой замены лежит следующий принцип: для получения шифртекста отдельные символы или группы символов исходного алфавита заменяются символами или группами символов шифроалфавита.

Шифр Цезаря

Шифр Цезаря является моноалфавитной подстановкой, т.е. каждой букве открытого текста ставится соответствие одна буква шифротекста.

Математическая процедура шифрования описывается как

$$T_m = \left\{T^j\right\}, j = 0, 1, \cdots, m-1,$$

$$T^j(a) = (a + j) \mod m,$$

где m - длина алфавитаа, j - произвольный ключ (величина сдвига от изначальной позиции буквы), a - текущая позиция буквы в алфавите.

Для латинского алфавита длина составляет 26 символов, а формулу можно привести к виду:

$$T^k(i) = (i + k) \mod 26,$$

где i, k соответствуют a, j, а $m = 26$.

Сам же Цезарь обычно использовал подстановку T^3.

Шифр Атбаш

Шифр Атбаш является сдвигом на всю длину алфавита. Правило шифрования состоит в замене i-й буквы алфавита буквой с номером $n - i + 1$, где n — число букв в алфавите.

Выполнение лабораторной работы

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

Реализация шифра Цезаря с произвольным ключом k

Шифр Цезаря реализуем в виде функции cesar следующего вида:

```
# --- Ceasar 's Cipher ---
def cesar (text, k):
    encr = ""
    for c in text:
        c1 = c.lower()
        c_ind=ord(c1) - ord("a")
        c_sh= (c_ind+k) % 26 + ord("a")
        c_new = chr(c_sh)
        if c.islower():
            encr += c_new
        elif c.isupper():
            encr += c_new.upper()
        else:
            encr += c
    return(encr)
```

На вход она принимает исходный текст и ключ(на сколько символов производится сдвиг.

Так как в исходном тексте могут встреться как строчные, так и заглавные буквы, то нужно сначала перевести все буквы в строчные, потом зашифровать их.

Затем проверить какого регистра была изначальная буква: если строчная, то добавить её в результирующую строку, если прописная, то сделать её прописной и добавить в результирующую строку, а если она не удовлетворяет ни одну из этих условий, то ее добавляем к результату без шифровки(это будут цифры, знаки препинания, пробелы и тд.

Реализация шифра Атбаша

Шифр Атбаш реализуем в виде функции atbash следующего вида:

```
# --- Atbash 's Cipher ---
def atbash(text, a):
    encr = ""
    for c in text:
        c1 = c.lower()
        if c1 not in a:
            encr += c
            break
        c_new = a[len(a)-1-a.index(c1)]
        if c.isupper():
            c_new = c_new.upper()
        encr += c_new
    return(encr)
```

На вход она принимает исходный текст и созданный специально для этого задания алфавит. код создания алфавита представлен ниже:

```
# --- Alphabet ---
alphab = list(map(chr, range(97, 123)))
alphab.append(chr(32))
```

Так как в исходном тексте могут встреться как строчные, так и заглавные буквы, то нужно сначала перевести все буквы в строчные.

Запускаем цикл по кажому символу из исходного текста.

После этого проводим проверку присутствует ли текущий символ в алфавите, если нет, то добавляем его в результиющую строку и выходим из текущей итерации цикла, если символ присутствует в алфавите, то шифруем его.

Далее проводим проверку какого регистра был исходный символ. Если он был строчный, то ничего не меняем, а если он был прописной, тогда меняем регистр.

Добавляем полученный символ к результирующей строке. Выводим полученный резульат

Тестирование

```
# --- Tests ---
print("Шифр Цезаря:")
print("Исходный текст: Hello world!\nЗашифрованный текст: ", cesar("Hello world!", 4))

print("Шифр Атбаш:")
print("Исходный текст: Twppmaemjpx!\nЗашифрованный текст: ", atbash("Twppmaemjpx!", alpha
```

Данные тесты возвращают строку шифро-текста.

Результаты тестирования

Запустив наш программный код, получим результат, изображенный в приложении [-@fig:001].

Для шифра Цезаря с ключом $k = 4$ получаем следующий результат:

CEASAR'S CIPHER TEST

Шифр Цезаря:

Исходный текст: Hello world!

Зашифрованный текст: Lipps asvph!

Из-за простоты шифров их можно проверить вручную. В первом случае исходный текст был: “Hello world!”, ключ мы приняли равным 4.

Таким образом вместо H мы должны были получить L, вместо e - i и тд., пробел и восклицательный знак должны остаться без изменений. Так и есть, это можно увидеть на

Для шифра Атбаш получаем следующий результат:

ATBASH'S CIPHER TEST

Шифр Атбаш:

Исходный текст: Twppmaemjpx!

Зашифрованный текст: Hello world!

Во втором случае текст был: “Twppmaemjpx!” (в записи лабораторной работы изначально сообщение было “Hello world!”). У данного шифра есть особенность: если закодированное сообщение закодировать еще раз мы должны получить исходное сообщение. В записи лабораторной работы я так и сделала, получила исходное сообщение “Hello world!”. Исходя из всего вышесказанного можем сделать вывод, что оба шифра работают корректно.

Выводы

В рамках выполненной лабораторной работы мы изучили и реализовали следующие шифры простой замены: шифр Цезаря (с произвольным ключом k) и шифр Атбаш.

Приложения

```
print("Шифр Цезаря:")
print("Исходный текст: Hello world!\nЗашифрованный текст: ", cesar("Hello world!", 4))
print("\nШифр Атбаш:")
print("Исходный текст: Twppmaemjpx!\nЗашифрованный текст: ", atbash("Twppmaemjpx!", alphab))
```

```
Шифр Цезаря:
Исходный текст: Hello world!
Зашифрованный текст:  Lipps asvph!

Шифр Атбаш:
Исходный текст: Twppmaemjpx!
Зашифрованный текст:  Hello world!
```

Рис. 0.1: Вывод программы