

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированных систем управления (АСУ)

Алгоритмы донаучной криптографии
Лабораторная работа №1
по дисциплине
«Информационная безопасность»

Студент гр. 430-2

_____ А.А. Лузинсан

«_____» _____ 2023 г.

Руководитель

Профессор кафедры АСУ, д.т.н.

_____ А.Н. Горитов

«_____» _____ 2023 г.

Томск 2023

Оглавление

Введение.....	3
1 Ход работы.....	4
1.1 Реализация алгоритма.....	4
2 Тестирование.....	9
Заключение.....	10

Введение

Цель: познакомиться и научиться работать с алгоритмами донаучной криптографии.

Задание по варианту №2: напишите программу, позволяющую зашифровать и расшифровать сообщения с использованием шифра сдвига. Входные и выходные данные запишите в файл типа .txt.

1 ХОД РАБОТЫ

Шифр сдвига, иначе известный как код Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиции левее или правее него в алфавите.

Например, имеется исходный текст: АБВГДЕЖЗ.

В шифре со сдвигом вправо на 3, зашифрованный текст будет иметь вид: ГДЕЖЗИЙК.

1.1 Реализация алгоритма

В реализации программы использовались шаблоны алфавитов, считываемые из файла. Во всех остальных случаях алгоритм сохраняет исходный символ.

Содержимое файла представлено листинге 1.1. Алгоритм проходит в цикле по символам кодируемой строки и фиксирует, к какому алфавиту принадлежит рассматриваемый символ. Это осуществляется за счёт циклического прохода по шаблонам, написанным в виде регулярных выражений и определения ключа рассматриваемого алфавита-кортежа. Далее определяется нормированное значение сдвига на случай непреднамеренной (или преднамеренной) ошибки пользователя, то есть обрабатывается случай, когда пользователь указал значение сдвига большее, чем мощность самого алфавита. И наконец инициализируется индекс нового символа в текущем алфавите. В случае, если рассматриваемый символ не принадлежит ни одному из указанных алфавитов, то возвращается сам символ.

Алгоритмом обеспечивается как положительный, так и отрицательный сдвиг.

Листинг 1.1 — Содержимое файла по лабораторной работе

```
import dearpygui.dearpygui as dpg
import time
import re

def set_path(sender, app_data):
    dpg.set_value('file', value=app_data['file_path_name'])

with dpg.file_dialog(directory_selector=False, show=False,
                    callback=set_path, tag="file_dialog",
                    width=700, height=400, modal=True):
    dpg.add_file_extension(".txt", color=(0, 255, 0, 255), custom_text="[Text]")

def decrypting(input_data, shift, area_alphabetic):
    decrypted_data = []
    for row in input_data:
        decrypted_data.append(".join(encrypting_row(row, -shift, area_alphabetic)))
    return decrypted_data

def get_input_data():
    if dpg.get_value('input_method') == 'File':
        file_path = dpg.get_value('file')
        file = open(file_path, 'r', encoding="utf-8")
        input_data = file.readlines()
        is_multiline = '\n'
    else:
        input_data = dpg.get_value('Manually')
        is_multiline = " "
        input_data = [row.replace('ë', 'e').replace('Ë', 'E').replace('\n', " ") for row in
input_data]
    return input_data, is_multiline

def get_alphabet_area():
    falphabets = open('alphabetic.txt', 'r', encoding="utf-8")
    alphabet_area = falphabets.readlines()
    alphabets = {}
    for item in alphabet_area:
```

```

        item = item[:3]
        alphabets[item] = tuple(chr(symbol)
                                for symbol in range(ord(item[0]), ord(item[-1]) + 1))
    falphabets.close()
    return alphabets

def encrypting_row(row, shift, area_alphabetic):
    combined_pattern = [f'[{pattern}]' for pattern in area_alphabetic.keys()]
    sign = -1 if shift < 0 else 1
    shift = abs(shift)
    enc_row = []
    for symbol in row:
        index_alphabetic = ".join([pattern[1:-1]
                                    for pattern in combined_pattern
                                    if re.search(pattern, symbol)])
        if index_alphabetic:
            alphabetic = area_alphabetic[index_alphabetic]
            shift_for_symbol = (shift % (len(alphabetic) - 1)) * sign
            index = (alphabetic.index(symbol) + shift_for_symbol) % (len(alphabetic))
            enc_row.append(alphabetic[index])
        else:
            enc_row.append(symbol)
    return enc_row

def encrypting(sender, app_data, user_data):
    dpg.show_item('Caesar\'s cipher')
    input_data, is_multiline = get_input_data()

    dpg.set_value('input data', value=is_multiline.join(input_data))
    shift = dpg.get_value('shift')
    encrypted_data = []
    num_dots = 1
    dpg.configure_item('dots', color=(0, 0, 255, 255))
    area_alphabetic = get_alphabet_area()
    if type(input_data) is list:
        for row in input_data:
            time.sleep(0.1)

```

```

        encrypted_data.append(".join(encrypting_row(row, shift, area_alphabetic)))
        dpg.set_value('dots', value='.' * num_dots)
        num_dots += 1
    else:
        encrypted_data = is_multiline.join(encrypting_row(input_data, shift))

    dpg.set_value('encrypted', value=is_multiline.join(encrypted_data))
    fout = open('output.txt', 'a', encoding="utf-8")
    fout.writelines(is_multiline.join(encrypted_data) + '\n')
    fout.close()
    # region test
    test = decrypting(encrypted_data, shift, area_alphabetic)
    dpg.set_value('test', value=is_multiline.join(test))
    if ".join(input_data) == ".join(test):
        dpg.configure_item('dots', default_value='True', color=(0, 255, 0, 255))
    else:
        dpg.configure_item('dots', default_value='False', color=(255, 0, 0, 255))
    # endregion

def switch_method(sender, method):
    dpg.hide_item('Caesar\'s cipher')
    dpg.show_item('shift')
    dpg.show_item('continue')
    if method == 'File':
        dpg.hide_item('Manually')
        dpg.show_item(method)
    else:
        dpg.hide_item('File')
        dpg.show_item(method)

with dpg.window(label="Лабораторная работа #1", tag='lr1', show=False,
width=500, height=700, pos=(100, 100)):
    dpg.add_radio_button(tag='input_method',
        items=["Manually", "File"],
        callback=switch_method,
        horizontal=True)
    dpg.add_input_int(tag='shift', label='Set Shift', default_value=1, show=False)

```

```

dpg.add_input_text(label='Input Text', tag='Manually', show=False,
                    default_value='абвәюя')
with dpg.group(horizontal=True, show=False, tag='File'):
    dpg.add_input_text(tag='file',
                       default_value='/home/luzinsan/Documents/TUSUR_learn/3_курс
/7_semester/ИБ/Лабораторные/1/test.txt')
    dpg.add_button(label='Select Path Manually', callback=lambda:
dpg.show_item("file_dialog"))
    dpg.add_button(label='Continue: Caesar\'s cipher', callback=encrypting,
show=False, tag='continue')
with dpg.group(tag='Caesar\'s cipher', show=False):
    dpg.add_text(tag='input data', label='Input Data', show_label=True)
    dpg.add_separator()
    dpg.add_text(tag='encrypted', label='Encrypted Message', show_label=True)
    dpg.add_separator()
    dpg.add_text(tag='test', label='Test', show_label=True)
    dpg.add_separator()
    dpg.add_text(tag='dots', color=(0, 0, 255, 255))

```


2 ТЕСТИРОВАНИЕ

Программа поддерживает файловый ввод исходного текста, либо же ввод вручную, а также вывод результата в выходной файл output.txt. Результат работы для файлового ввода представлен на рисунке 2.1.

Пример работы программы на данных, введённых вручную, представлен на рисунке 2.2.

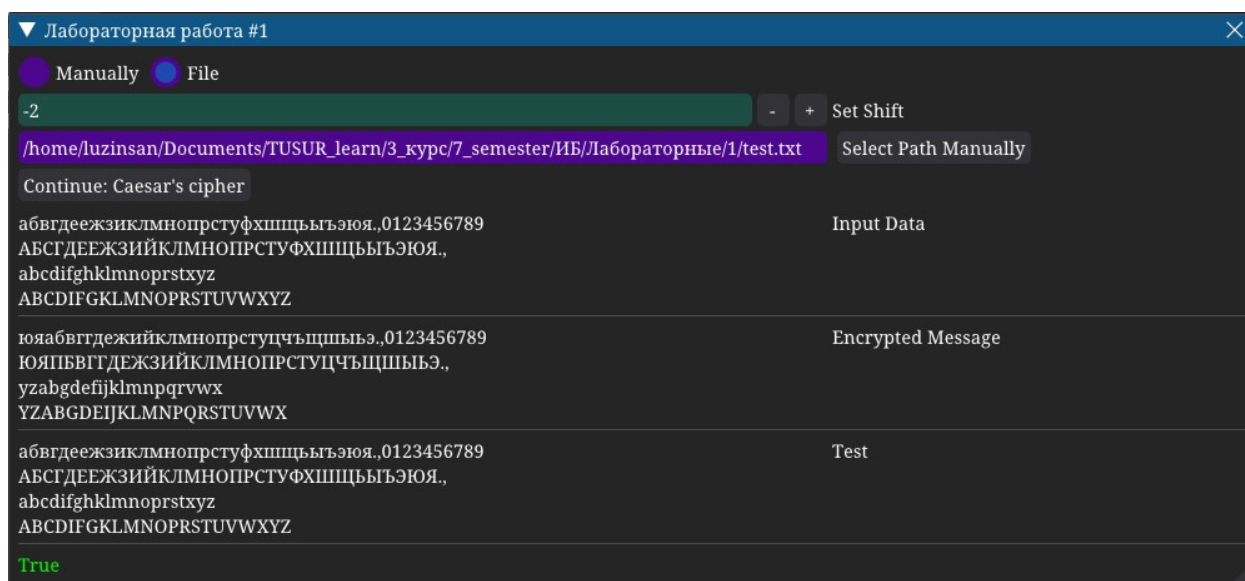


Рисунок 2.1 — Кодирование текста из файла

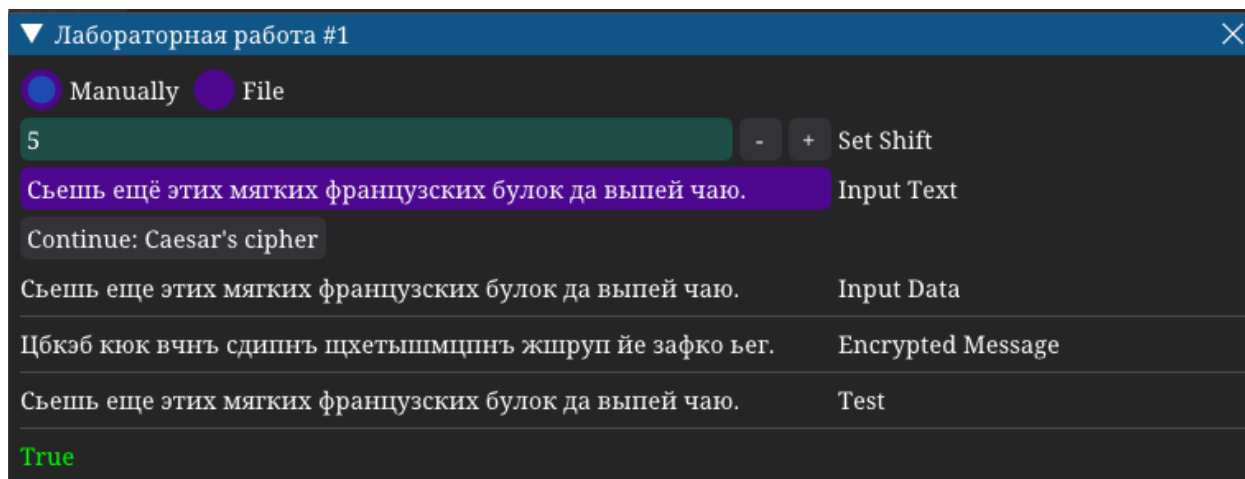


Рисунок 2.2 — Кодирование текста, введённого вручную

Заключение

В результате выполнения лабораторной работы я познакомилась и научилась работать с алгоритмом донаучной криптографии на примере шифра сдвига, а также выполнила задание в соответствии с заданным вариантом на языке Python.