

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет Информатика и вычислительная техника

Кафедра Кибербезопасность информационных систем

**Лабораторная работа № 1**

на тему «**Реализация стеганографических методов сокрытия информации в MS Word**»

Выполнил обучающийся гр. ВКБ51:

Автайкин Алексей Яковлевич

Проверила:

доцент, к.т.н.

Сафарьян Ольга Александровна

Ростов-на-Дону

2024

**Цель работы:** описать стеганографические методы и выполнить практическую работу.

**Ход работы:**

Работа программы:

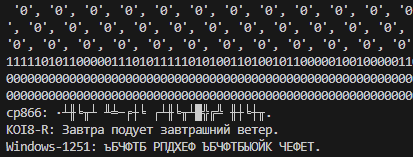


Рисунок 1 – Вывод работы программы

Сокрытие было выполнено с использованием кодировки koi8\_r в цвете шрифта.

Скрытая фраза: Завтра подует завтрашний ветер.

**Вывод:** В данной лабораторной работе были получены навыки обнаружения замаскированных сообщений, были изучены стеганографические методы и выполнена практическая работа.

**Листинг программного кода:**

from docx import Document

def rgb\_to\_binary(rgb):

binary\_values = [format(component, '08b')[2:] for component in rgb]

binary\_string = ''.join(binary\_values)

return binary\_string

def get\_binary\_color(docx\_file):

listofcolors=[]

doc = Document(docx\_file)

for paragraph in doc.paragraphs:

for run in paragraph.runs:

for char in run.text:

print(f"Character: {char}")

print(f"Text Color: {run.font.color.rgb}")

print(f"Background Color: {run.font.highlight\_color}")

print(f"Font Size: {run.font.size.pt} pt")

print("-" \* 20)

if run.font.color is not None:

rgb = run.font.color.rgb

binary\_color = rgb\_to\_binary(rgb)

listofcolors.append(binary\_color)

print(listofcolors)

listofcolors = list(map(lambda x: x.replace('000000000000000000', '0'), listofcolors))

listofcolors = list(map(lambda x: x.replace('000001000001000001', '1'), listofcolors))

print(listofcolors)

stringofcolors = ''.join(listofcolors)

print(stringofcolors)

binary\_chunks = [stringofcolors[i:i+8] for i in range(0, len(stringofcolors), 8)]

encodings = {

'cp866': 'cp866',

'KOI8-R': 'koi8-r',

'Windows-1251': 'windows-1251',

}

decoded\_strings = {encoding: "" for encoding in encodings}

for chunk in binary\_chunks:

chunk\_bytes = int(chunk, 2).to\_bytes(len(chunk) // 8, byteorder='big')

for encoding\_name, encoding in encodings.items():

decoded\_strings[encoding\_name] += chunk\_bytes.decode(encoding)

for encoding\_name, decoded\_string in decoded\_strings.items():

print(f"{encoding\_name}: {decoded\_string}")

get\_binary\_color("variant01.docx")