ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

Факультет безопасности информационных технологий

Дисциплина:

«Основы стеганографии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

«Встраивание информации в картинки»

Выполнил:

Студент гр. N3352

Шарипов Ф.Р.

Проверил:

Давыдов В. В.

Санкт-Петербург

2020г.

Цель работы: Реализация встраивание информации в картинку методом замены наименее значимых битов.

Теоретическая часть

Стеганография – это наука о способах передачи или хранения информации при условии сохранения в тайне самого факта наличия скрытой информации.

Основными стеганографическими понятиями являются сообщение и контейнер.

Сообщение – это секретная информация, наличие которой необходимо скрыть.

Контейнер - некоторая информация, в частности файл, в которую можно внедрить другую информацию, не предназначенную для посторонних глаз [1].

Наиболее распространённым случаем является передача скрытых сообщений в графических контейнерах. Наиболее известным методом является, замены наименее значащего бита. Можно считать, что он лежит в основе методов псевдослучайного интервала, псевдослучайной перестановки и блочного скрытия, которые созданы для исправления главного недостатка – слабой стойкости к внешним воздействиям [1,2].

Пустым контейнером в этих случаях является графический файл формата bmp.

Практическая часть

Часть 1

В данной работе в качестве пустого контейнера используется графический файл формата bmp, размером 320x480 пикселей и глубиной цвета 24 бита, представленный на Рисунке 1. Скрытой же информацией будет отрывок из английской статьи «Internet Security» [3].

Рисунок 1- Пустой контейнер формата bmp.

Рассмотрим алгоритм, встраивания сообщения в контейнер, представленный на Рисунке 2.

Входные файлы:

- «1.bmp» - пустой контейнер;
- «inf.txt»- встраиваемое сообщение;

Выходные файлы:

- «steg.bmp»

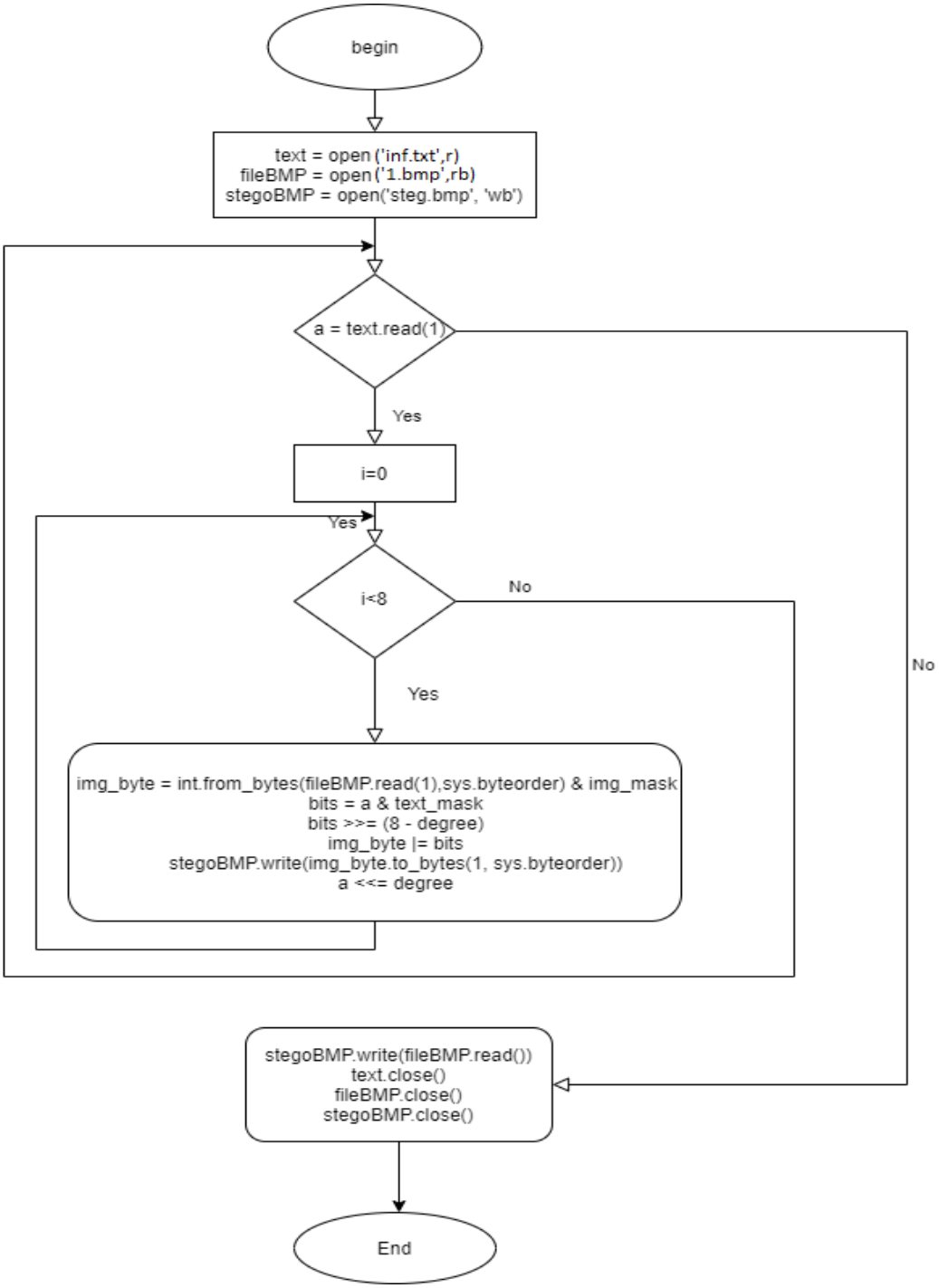

Рисунок 2 – *блок-схема алгоритма встраивания информации.*

Рассмотрим процесс извлечения информации из стегоконтейнера, представленный на Рисунке 3.

Входной данные:

- Стегоконтейнер «steg.bmp»;
- Количество скрытой информации в стегоконтейнере;

Выходные данные:

- Секретное сообщение

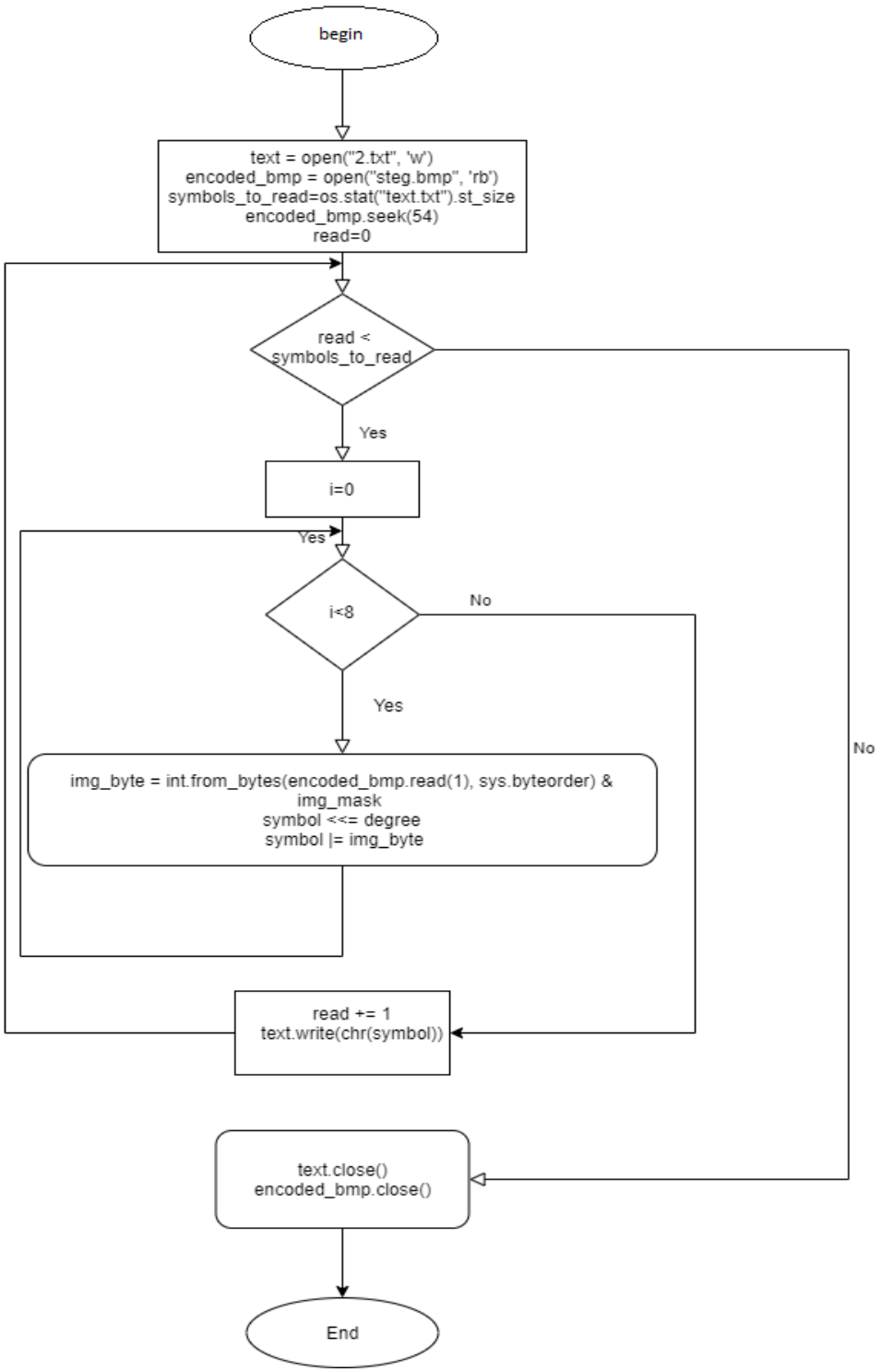

Рисунок 3 - *блок-схема алгоритма извлечения информации.*

Часть 2

В качестве относительной количественной оценки искажений в изображении рассматривается метрика PSNR.

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

Где MSE среднеквадратичное отклонение

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|I(i,j) - K(i,j)|^2$$

PSNR от исходного изображения и изображения с встроенной информации равно: 69.24340583950305

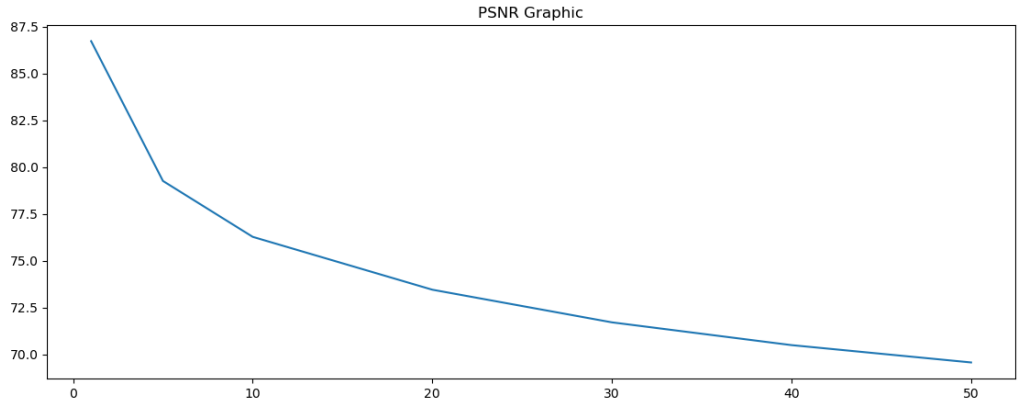

Рисунок 4 – *график зависимости, количество встраиваемой информации от PSNR.*

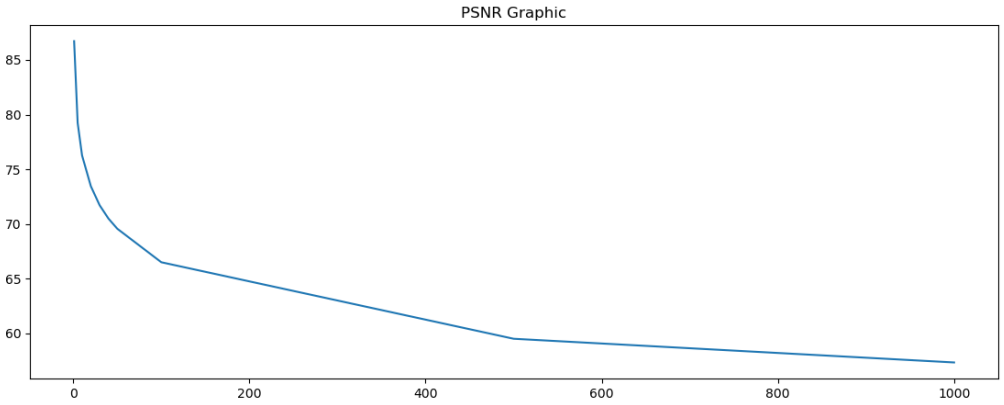

Рисунок 5 – *график зависимости, количество встраиваемой информации от PSNR.*

Проведя анализ картинок со встроенными сообщениями «Information» в длину 1, 5 ,10, 20, 30, 40, 50 слов, был построен график Рисунок 4.
Для более подробной оценки было встроено до 1000 слов, Рисунок 5.

В итоге проделанной работы можно сделать следующие выводы:

1. При встраивании большого объема данных значения искажения не доходит до значения в 50 децибел. Следовательно, можно встроить большие объемы данных без вызова подозрений со стороны.

2. Данный метод сильно зависит от канала передачи, так как малейшие искажения в изображении могут повлечь за собой искажения в отправляемом сообщении.

3. Визуально невозможно отличить изображение с встроенной информации, и оригиналом

Литература:

1. Зайцева А.В. Методика построения энтропийных стеганографических систем защиты сообщений в информационных сетях: автореф.дис. … к.т.н. МГТУ им. Н.Э.Баумана, Москва, 2014.
2. Изычаев А.В., Сидоренко В. Г. Стеногорафические методы защиты информации. Москва, 2017.
3. Английская статья [электронный ресурс] information-security-news.com //URL: https://www.information-security-news.com/2014/01/25/internet-security/
4. PSNR и SSIM [электронный ресурс] habr.com //URL: https://habr.com/ru/post/126848/

Листинг программы:

```
import os
import sys
degree=1
text_mask = 0b11111111
img_mask = 0b11111111

text_mask <<= (8 - degree)
text_mask %= 256
img_mask >>= degree
img_mask <<= degree
##Встраивание
text = open('inf.txt', 'r')
fileBMP = open('1.bmp', 'rb')
stegoBMP = open('steg.bmp', 'wb')

headBMP = fileBMP.read(54)
stegoBMP.write(headBMP)

while True:
    a = text.read(1)
    if not a:
        break
    a = ord(a)
    for i in range(0, 8, degree):
        img_byte = int.from_bytes(fileBMP.read(1), sys.byteorder) & img_mask
        bits = a & text_mask
        bits >>= (8 - degree)
        img_byte |= bits
        stegoBMP.write(img_byte.to_bytes(1, sys.byteorder))
```

```
        a <<= degree
stegoBMP.write(fileBMP.read())

text.close()
fileBMP.close()
stegoBMP.close()
 #Извлечение информации
text = open("2.txt", 'w',encoding='utf-8')
encoded_bmp = open("steg.bmp", 'rb')
symbols_to_read=os.stat("inf.txt").st_size
encoded_bmp.seek(54)
img_mask = ~img_mask

read = 0
while read < symbols_to_read:
    symbol = 0
    for bits_read in range(0, 8, degree):
        img_byte = int.from_bytes(encoded_bmp.read(1), sys.byteorder) &
img_mask
        symbol <<= degree
        symbol |= img_byte
    read += 1
    text.write(chr(symbol))
    print (chr(symbol))

text.close()
encoded_bmp.close()
```