Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №8

По дисциплине «Методы защиты информации»

По теме «Стенографические методы»

Выполнила:

Студентка гр. 653501

Д. А. Гуринович

Проверил:

В. С. Артемьев

Минск 2019

Содержание

[1. Постановка задачи](#_2et92p0)

2. Краткие теоретические сведения

3. Схема алгоритма

4[. Результаты выполнения](#_tyjcwt)

5[. Вывод](#_3dy6vkm)

[ПРИЛОЖЕНИЕ 1. Исходный код программы.](#_1t3h5sf)

# 1. Постановка задачи

В современном мире остро стоит вопрос о безопасности. Обеспечение безопасности является важным аспектом деятельности любой компании. Для обеспечения безопасности используется множество различных средств, как аппаратных, так и программных.

Стеганография — способ передачи или хранения [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) с учётом сохранения в тайне самого факта такой передачи (хранения). Этот термин ввел в [1499 году](https://ru.wikipedia.org/wiki/1499_%D0%B3%D0%BE%D0%B4) [аббат](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D0%B1%D0%B0%D1%82) бенедиктинского монастыря Св. Мартина в Шпонгейме [Иоганн Тритемий](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D1%82%D0%B5%D0%BC%D0%B8%D0%B9,_%D0%98%D0%BE%D0%B3%D0%B0%D0%BD%D0%BD) в своем трактате «Стеганография», зашифрованном под магическую книгу.

В ходе данной лабораторной работы необходимо изучить теоретическую часть о стенографических методах, на основании которых создать приложение, использующие данные технологии.

**2. Краткие теоретические сведения**

Стеганогра́фия — способ передачи или хранения [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) с учётом сохранения в тайне самого факта такой передачи (хранения). Этот термин ввел в [1499 году](https://ru.wikipedia.org/wiki/1499_%D0%B3%D0%BE%D0%B4) [аббат](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D0%B1%D0%B0%D1%82) бенедиктинского монастыря Св. Мартина в Шпонгейме[Иоганн Тритемий](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D1%82%D0%B5%D0%BC%D0%B8%D0%B9,_%D0%98%D0%BE%D0%B3%D0%B0%D0%BD%D0%BD) в своем трактате «Стеганография», зашифрованном под магическую книгу.

В отличие от [криптографии](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F), которая скрывает содержимое тайного сообщения, стеганография скрывает сам факт его существования. Как правило, сообщение будет выглядеть как что-либо иное, например, как изображение, статья, список покупок, письмо или [судоку](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B4%D0%BE%D0%BA%D1%83). Стеганографию обычно используют совместно с методами криптографии, таким образом, дополняя её.

Преимущество стеганографии над чистой криптографией состоит в том, что сообщения не привлекают к себе внимания. Сообщения, факт шифрования которых не скрыт, вызывают подозрение и могут быть сами по себе уличающими в тех странах, в которых запрещена криптография[[1]](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%B3%D0%B0%D0%BD%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F#cite_note-1). Таким образом, криптография защищает содержание сообщения, а стеганография защищает сам факт наличия каких-либо скрытых посланий.

В 1983 году [Симмонс](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%BE%D0%BD%D1%81,_%D0%93%D1%83%D1%81%D1%82%D0%B0%D0%B2%D1%83%D1%81_%D0%94%D0%B6%D0%B5%D0%B9%D0%BC%D1%81) предложил т. н. «проблему заключённых». Её суть состоит в том, что есть человек на свободе (Алиса), в заключении (Боб) и охранник Вилли. Алиса хочет передавать [сообщения](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B5) Бобу без вмешательства охранника. В этой модели сделаны некоторые допущения: предполагается, что перед заключением Алиса и Боб договариваются о кодовом символе, который отделит одну часть текста письма от другой, в которой скрыто сообщение. Вилли же имеет право читать и изменять сообщения. В 1996 году на конференции Information Hiding: First Information Workshop была принята единая терминология:

* Стеганографическая система (стегосистема) — объединение методов и средств, используемых для создания скрытого канала для передачи [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F). При построении такой системы условились о том, что: 1) враг представляет работу стеганографической системы. Неизвестным для противника является [ключ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)), с помощью которого можно узнать о факте существования и содержание тайного сообщения. 2) При обнаружении противником наличия скрытого сообщения он не должен смочь извлечь сообщение до тех пор, пока он не будет владеть [ключом](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)). 3) Противник не имеет технических и прочих преимуществ.
* Сообщение — общее название передаваемой скрытой информации, будь то лист с надписями молоком, голова раба или цифровой файл.
* Контейнер — любая [информация](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), используемая для сокрытия тайного сообщения.
  + Пустой контейнер — контейнер, не содержащий секретного послания.
  + Заполненный контейнер (стегоконтейнер) — контейнер, содержащий секретное послание.
* Стеганографический канал (стегоканал) — канал передачи стегоконтейнера.
* Ключ (стегоключ) — секретный [ключ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)), нужный для сокрытия стегоконтейнера. Ключи в стегосистемах бывают двух типов: закрытые (секретные) и открытые. Если стегосистема использует закрытый ключ, то он должен быть создан или до начала обмена сообщениями, или передан по защищённому каналу. Стегосистема, использующая [открытый ключ](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D0%BA%D0%BB%D1%8E%D1%87), должна быть устроена таким образом, чтобы было невозможно получить из него [закрытый ключ](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D0%BA%D0%BB%D1%8E%D1%87). В этом случае открытый ключ можно передавать по незащищённому каналу.

Компьютерная стеганография — направление классической стеганографии, основанное на особенностях компьютерной платформы. Примеры — стеганографическая файловая система StegFS для [Linux](https://ru.wikipedia.org/wiki/Linux), скрытие данных в неиспользуемых областях форматов [файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), подмена символов в названиях [файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), [текстовая стеганография](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D1%82%D0%B5%D0%B3%D0%B0%D0%BD%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) и т. д. Приведём некоторые примеры:

* Использование зарезервированных полей компьютерных [форматов файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0) — суть метода состоит в том, что часть [поля расширений](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%B8%D0%B5_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0), не заполненная информацией о расширении, по умолчанию заполняется нулями. Соответственно мы можем использовать эту «нулевую» часть для записи своих данных. Недостатком этого метода является низкая степень скрытности и малый объём передаваемой информации.
* Метод скрытия информации в неиспользуемых местах [гибких дисков](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%B1%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA) — при использовании этого метода [информация](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) записывается в неиспользуемые части [диска](https://ru.wikipedia.org/wiki/%D0%96%D1%91%D1%81%D1%82%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA), к примеру, на нулевую дорожку. Недостатки: маленькая производительность, передача небольших по объёму сообщений.
* Метод использования особых свойств полей форматов, которые не отображаются на экране — этот метод основан на специальных «невидимых» полях для получения сносок, указателей. К примеру, написание чёрным шрифтом на чёрном фоне. Недостатки: маленькая производительность, небольшой объём передаваемой информации.
* Использование особенностей [файловых систем](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) — при хранении на жёстком диске файл всегда (не считая некоторых ФС, например, [ReiserFS](https://ru.wikipedia.org/wiki/ReiserFS)) занимает целое число [кластеров](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%B5%D0%B4%D0%B8%D0%BD%D0%B8%D1%86%D0%B0_%D1%85%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)) (минимальных адресуемых объёмов информации). К примеру, в ранее широко используемой файловой системе [FAT32](https://ru.wikipedia.org/wiki/FAT32)(использовалась в [Windows98](https://ru.wikipedia.org/wiki/Windows_98)/[Me](https://ru.wikipedia.org/wiki/Windows_Me)/[2000](https://ru.wikipedia.org/wiki/Windows_2000)) стандартный размер кластера — 4 [КБ](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BB%D0%BE%D0%B1%D0%B0%D0%B9%D1%82). Соответственно для хранения 1 КБ информации на диске выделяется 4 КБ памяти, из которых 1 КБ нужен для хранения сохраняемого файла, а остальные 3 ни на что не используются — соответственно их можно использовать для хранения информации. Недостаток данного метода: лёгкость обнаружения.

**3. Схема алгоритма**

# 

# 4. Результаты выполнения

# 

Рисунок 4.1. Исходное изображение

# 

Рисунок 4.2. Изображение со встроенным текстом “Hello world”

**5. Вывод**

В рамках данной лабораторной работы были рассмотрены стенографические методы защиты информации.

Для демонстрации работы описанных выше алгоритмов было разработано и реализовано приложение с использованием указанного выше алгоритма.

# ПРИЛОЖЕНИЕ 1. ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ

import PIL

import numpy as np

from skimage import io

from skimage.util import view\_as\_blocks

from scipy.fftpack import dct, idct

u1, v1 = 4, 5

u2, v2 = 5, 4

n = 8

P = 25

def nsplit(s, n):

return [s[k:k + n] for k in range(0, len(s), n)]

def binvalue(val, bitsize):

binval = bin(val)[2:] if isinstance(val, int) else bin(ord(val))[2:]

if len(binval) > bitsize:

raise Exception("binary value larger than the expected size")

while len(binval) < bitsize:

binval = "0" + binval

return binval

def string\_to\_bit\_array(raw\_text): # Convert a string into a list of bits

array = list()

for char in raw\_text:

binval = binvalue(char, 8) # Get the char value on one byte

array.extend([int(x) for x in list(binval)]) # Add the bits to the final list

return array

def bit\_array\_to\_string(array): # Recreate the string from the bit array

res = ''.join([chr(int(y, 2)) for y in [''.join([str(x) for x in \_bytes]) for \_bytes in nsplit(array, 8)]])

return res

def double\_to\_byte(arr):

return np.uint8(np.round(np.clip(arr, 0, 255), 0))

def increment\_abs(x):

return x + 1 if x >= 0 else x - 1

def decrement\_abs(x):

if np.abs(x) <= 1:

return 0

else:

return x - 1 if x >= 0 else x + 1

def abs\_diff\_coefs(transform):

return abs(transform[u1, v1]) - abs(transform[u2, v2])

def valid\_coefficients(transform, bit, threshold):

difference = abs\_diff\_coefs(transform)

if (bit == 0) and (difference > threshold):

return True

elif (bit == 1) and (difference < -threshold):

return True

else:

return False

def change\_coefficients(transform, bit):

coefs = transform.copy()

if bit == 0:

coefs[u1, v1] = increment\_abs(coefs[u1, v1])

coefs[u2, v2] = decrement\_abs(coefs[u2, v2])

elif bit == 1:

coefs[u1, v1] = decrement\_abs(coefs[u1, v1])

coefs[u2, v2] = increment\_abs(coefs[u2, v2])

return coefs

def embed\_bit(block, bit):

patch = block.copy()

coefficients = dct(dct(patch, axis=0), axis=1)

while not valid\_coefficients(coefficients, bit, P) or (bit != retrieve\_bit(patch)):

coefficients = change\_coefficients(coefficients, bit)

patch = double\_to\_byte(idct(idct(coefficients, axis=0), axis=1) / (2 \* n) \*\* 2)

return patch

def embed\_message(original, msg):

changed = original.copy()

blue = changed[:, :, 2]

width, height = np.shape(blue)

width -= width % n

height -= height % n

blue = blue[:width, :height]

blocks = view\_as\_blocks(blue, block\_shape=(n, n))

h = blocks.shape[1]

for index, bit in enumerate(msg):

i = index // h

j = index % h

block = blocks[i, j]

blue[i \* n: (i + 1) \* n, j \* n: (j + 1) \* n] = embed\_bit(block, bit)

changed[:width, :height, 2] = blue

return changed

def retrieve\_bit(block):

transform = dct(dct(block, axis=0), axis=1)

return 0 if abs\_diff\_coefs(transform) > 0 else 1

def retrieve\_message(img, length):

img = img[:, :, 2]

width, height = np.shape(img)

width -= width % n

height -= height % n

img = img[:width, :height]

blocks = view\_as\_blocks(img, block\_shape=(n, n))

h = blocks.shape[1]

return [retrieve\_bit(blocks[index // h, index % h]) for index in range(length)]

def encrypt(data\_in, image\_in, image\_out):

image\_in = io.imread(image\_in)

data\_in = string\_to\_bit\_array(data\_in)

data\_in\_len = len(data\_in)

image\_out = embed\_message(image\_in, data\_in)

return image\_out, data\_in\_len

def decrypt(text\_length, image\_out):

image\_out = io.imread(image\_out)

out\_message = retrieve\_message(image\_out, text\_length)

out\_message = bit\_array\_to\_string(out\_message)

return out\_message