Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчёт

по лабораторной работе №8

**Стеганографические методы**

Выполнил:

Александров А.А.

Проверил:

Артемьев В.С.

Минск 2019

Содержание

[1. Постановка задачиError: Reference source not found](#_Toc506485975)2

2. Краткие теоретические сведения

3[. Результаты выполненияError: Reference source not found](#_Toc506485977)

4[. ВыводыError: Reference source not found](#_Toc506485978)

[ПРИЛОЖЕНИЕ 1. Исходный код программы.Error: Reference source not foundError: Reference source not foundError: Reference source not found](#_Toc506485979)

# 1. Постановка задачи

1. Изучить теоретические сведения.

2. Реализовать программное средство сокрытия (извлечения) текстового сообщения в (из) JPEG изображение(я) на основе метода сокрытия в частотной области изображения.

**2. Краткие теоретические сведения**

**Стеганогра́фия** — способ передачи или хранения [информации](https://ru.wikipedia.org/wiki/Информация) с учётом сохранения в тайне самого факта такой передачи (хранения). Этот термин ввел в [1499 году](https://ru.wikipedia.org/wiki/1499_год) [аббат](https://ru.wikipedia.org/wiki/Аббат) бенедиктинского монастыря Св. Мартина в Шпонгейме[[](de:Sponheim)de] [Иоганн Тритемий](https://ru.wikipedia.org/wiki/Тритемий,_Иоганн) в своем трактате «Стеганография» ([лат.](https://ru.wikipedia.org/wiki/Латинский_язык) *Steganographia*), зашифрованном под магическую книгу.

В отличие от [криптографии](https://ru.wikipedia.org/wiki/Криптография), 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/Судоку). 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. Таким образом, криптография защищает содержание сообщения, а стеганография защищает сам факт наличия каких-либо скрытых посланий.

## Классификация стеганографии

В конце 1990-х годов выделилось несколько направлений стеганографии:

* **классическая**,
* **компьютерная**,
* **цифровая**.

## Классическая стеганография

### Симпатические чернила

Одним из наиболее распространённых методов **классической стеганографии** является использование [симпатических (невидимых) чернил](https://ru.wikipedia.org/wiki/Симпатические_чернила). Текст, записанный такими чернилами, проявляется только при определённых условиях (нагрев, освещение, химический проявитель и т. д.)[[6]](https://ru.wikipedia.org/wiki/Стеганография" \l "cite_note-gromov-6) Изобретённые ещё в I веке н. э. [Филоном Александрийским](https://ru.wikipedia.org/wiki/Филон_Александрийский)[[7]](https://ru.wikipedia.org/wiki/Стеганография" \l "cite_note-7), они продолжали использоваться как в [средневековье](https://ru.wikipedia.org/wiki/Средневековье), так и в [новейшее время](https://ru.wikipedia.org/wiki/Новейшее_время), например, в письмах русских революционеров из тюрем. В советское время школьники на уроках литературы изучали рассказ, как Владимир Ленин писал молоком на бумаге между строк (см. «[Рассказы о Ленине](https://ru.wikipedia.org/wiki/Рассказы_о_Ленине_(Михаил_Зощенко))»). Строки, написанные молоком, становились видимыми при нагреве над пламенем свечи.

Существуют также чернила с химически нестабильным [пигментом](https://ru.wikipedia.org/wiki/Краситель). Написанное этими чернилами выглядит как написанное обычной ручкой, но через определённое время нестабильный пигмент разлагается, и от текста не остаётся и следа. Хотя при использовании обычной шариковой ручки текст можно восстановить по деформации [бумаги](https://ru.wikipedia.org/wiki/Бумага), этот недостаток можно устранить с помощью мягкого пишущего узла, наподобие [фломастера](https://ru.wikipedia.org/wiki/Фломастер).

### Другие стеганографические методы

Во время [Второй мировой войны](https://ru.wikipedia.org/wiki/Вторая_мировая_война) активно использовались [**микроточки**](https://ru.wikipedia.org/wiki/Микроточка) — микроскопические фотоснимки, вклеиваемые в текст писем.

Также существует ряд альтернативных методов сокрытия информации:

* запись на боковой стороне колоды карт, расположенных в условленном порядке;
* запись внутри варёного яйца;
* «жаргонные шифры», где слова имеют другое обусловленное значение;
* [трафареты](Кардано), которые, будучи положенными на текст, оставляют видимыми только значащие буквы;
* геометрическая форма — метод, в котором отправитель старается скрыть ценную информацию, поместив ее в сообщение так, чтобы важные слова расположились в нужных местах или в узлах пересечения геометрического рисунка;
* семаграммы — секретные сообщения, в которых в качестве шифра используются различные знаки, за исключением букв и цифр;
* узелки на нитках и т. д.

В настоящее время под **стеганографией** чаще всего понимают скрытие информации в текстовых, графических либо аудиофайлах путём использования специального [программного обеспечения](https://ru.wikipedia.org/wiki/Программное_обеспечение).

## Стеганографические модели

**Стеганографические модели** — используются для общего описания стеганографических систем.

### Основные понятия

В 1983 году [Симмонс](https://ru.wikipedia.org/wiki/Симмонс,_Густавус_Джеймс) 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/Электронное_сообщение) 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/Информация). При построении такой системы условились о том, что: 1) враг представляет работу стеганографической системы. Неизвестным для противника является [ключ](https://ru.wikipedia.org/wiki/Ключ_(криптография)), 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/Ключ_(криптография)). 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/Информация), 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/Ключ_(криптография)), 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/Открытый_ключ), должна быть устроена таким образом, чтобы было невозможно получить из него [закрытый ключ](https://ru.wikipedia.org/wiki/Закрытый_ключ). В этом случае открытый ключ можно передавать по незащищённому каналу.

## Компьютерная стеганография

**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/Файл), подмена символов в названиях [файлов](https://ru.wikipedia.org/wiki/Файл), [текстовая стеганография](https://ru.wikipedia.org/wiki/Текстовая_стеганография) и т. д. Приведём некоторые примеры:

* Использование зарезервированных полей компьютерных [форматов файлов](https://ru.wikipedia.org/wiki/Формат_файла) — суть метода состоит в том, что часть [поля расширений](https://ru.wikipedia.org/wiki/Расширение_файла), 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/Гибкий_диск) — при использовании этого метода [информация](https://ru.wikipedia.org/wiki/Информация) записывается в неиспользуемые части [диска](https://ru.wikipedia.org/wiki/Жёсткий_диск), 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/Файловая_система) — 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/Кластер_(единица_хранения_данных)) (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/Килобайт). 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.

**Цифровая стеганография** — направление классической стеганографии, основанное на сокрытии или внедрении дополнительной информации в цифровые объекты, вызывая при этом некоторые искажения этих объектов. Но, как правило, данные объекты являются мультимедиа-объектами (изображения, видео, аудио, текстуры 3D-объектов) и внесение искажений, которые находятся ниже порога чувствительности среднестатистического человека, не приводит к заметным изменениям этих объектов. Кроме того, в оцифрованных объектах, изначально имеющих аналоговую природу, всегда присутствует шум квантования; далее, при воспроизведении этих объектов появляется дополнительный аналоговый шум и нелинейные искажения аппаратуры, все это способствует большей незаметности сокрытой информации.

### Алгоритмы

Все алгоритмы встраивания скрытой информации можно разделить на несколько подгрупп:

* Работающие с самим цифровым сигналом. Например, метод LSB.
* «Впаивание» скрытой информации. В данном случае происходит наложение скрываемого изображения (звука, иногда текста) поверх оригинала. Часто используется для встраивания цифровых водяных знаков (ЦВЗ).
* Использование особенностей форматов [файлов](https://ru.wikipedia.org/wiki/Файл). Сюда можно отнести запись информации в [метаданные](https://ru.wikipedia.org/wiki/Метаданные) или в различные другие не используемые зарезервированные поля файла.

По способу встраивания информации стегоалгоритмы можно разделить на линейные (аддитивные), нелинейные и другие. Алгоритмы аддитивного внедрения информации заключаются в линейной модификации исходного изображения, а её извлечение в декодере производится корреляционными методами. При этом ЦВЗ обычно складывается с изображением-контейнером либо «вплавляется» (fusion) в него. В нелинейных методах встраивания информации используется скалярное либо векторное квантование.

#### Стегоалгоритмы встраивания информации в изображения

Именно изображения чаще всего используются в качестве стегоконтейнеров. Вот несколько значимых причин:

* практическая значимость задачи защиты фотографий, картин, видео и прочей графической информации от незаконного копирования и распространения.
* большой информационный объем цифрового изображения, что позволяет скрывать [ЦВЗ](https://ru.wikipedia.org/wiki/Цифровой_водяной_знак) (скрываемую информацию) большого объема, либо делать больше [устойчивость](https://ru.wikipedia.org/wiki/Робастность) внедрения.
* на момент встраивания ЦВЗ известен конечный объём контейнера.
* нет ограничений на встраивание ЦВЗ в режиме реального времени, как, например, в потоковом видео.
* многие изображения имеют области, имеющие шумовую структуру и хорошо подходящих для встраивания информации.
* криптоанализ таких систем начинается, обычно, с визуальной оценки, однако глаз не может различить незначительное изменение оттенка, вызванное записью информации в битовое представление цветов, что оставляет такой канал передачи информации вне подозрения.

Раньше цифровой водяной знак старались вложить в незначащие биты цифрового представления, что уменьшало визуальную заметность изменений. Однако со временем алгоритмы сжатия стали столь совершенны, что сжатие стеганограммы с ЦВЗ в незначащих битах может привести к потере в том числе и скрываемой информации. Это заставляет современные стегоалгоритмы встраивать ЦВЗ в наиболее существенные с точки зрения алгоритмов сжатия области изображения — такие области, удаление которых привело бы к существенной деформации изображения. Алгоритмы сжатия изображений работают подобно системе человеческого зрения: выделяются наиболее значимые части изображения и отсекаются наименее значимые с точки зрения человека (например, длинные тонкие линии обращают на себя больше внимания, чем круглые однородные объекты). Именно поэтому в современных стегоалгоритмах анализу человеческого зрения уделяется такое же внимание, как и алгоритмам сжатия.

Техника FontCode — информация шифруется в незаметных глазу изменениях глифа (формы отдельных символов), не затрагивая суть самого текста[[9]](https://ru.wikipedia.org/wiki/Стеганография" \l "cite_note-9).

**Статистический метод** — метод сокрытия данных, при котором изменяются определенные статистические характеристики изображения, при этом получатель способен распознать видоизмененное изображение от исходного.

# Выводы

# Таким образом, у нас есть три доступных варианта стеганографии: auto, join и steg. Авто по умолчанию (мной так было решено) использует join для кодирования, склеивая файлы воедино (не обязательно с архивом — с чем угодно), единственное отличие, что только с join можно использовать пустой пароль для создания RarJPEG, а с auto и steg по соображениям безопасности нет. Существует ещё одна хитрая возможность: файл можно закодировать в контейнер steg методом, а потом к нему же прицепить что-нибудь join методом, это позволяет в случае «прижатия к стенке» выдать пароль от join-части, не скомпрометировав при этом steg-часть — получается такой контейнер с «двойным дном». К слову, если картинка будет хоть как-то изменена (обрезана, пережата etc.), никакая стеганография по определению, увы, не выживет, JPEG — формат сжатия с потерями качества.

# ПРИЛОЖЕНИЕ 1. ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ

from PIL import Image

import binascii

from binascii import unhexlify

import codecs

import optparse

def rgb2hex(r, g, b):

return '#{:02x}{:02x}{:02x}'.format(r, g, b)

def hex2rgb(hexcode):

return unhexlify(bytes(map(ord, hexcode[1:])))

def str2bin(message):

binary = bin(int(binascii.hexlify(message.encode("ascii")), 16))

return binary[2:]

def bin2str(binary):

message = binascii.unhexlify('%x' % (int('0b' + binary, 2)))

return message

def encode(hexcode, digit):

if hexcode[-1] in ('0', '1', '2', '3', '4', '5'):

hexcode = hexcode[:-1] + digit

return hexcode

else:

return None

def decode(hexcode):

if hexcode[-1] in ('0', '1'):

return hexcode[-1]

else:

return None

def hide(filename, message):

img = Image.open(filename)

binary = str2bin(message) + '1111111111111110'

if img.mode in ('RGBA'):

img = img.convert('RGBA')

datas = img.getdata()

newData = []

digit = 0

temp = ''

for item in datas:

if (digit < len(binary)):

newpix = encode(rgb2hex(item[0],item[1],item[2]),binary[digit])

if newpix == None:

newData.append(item)

else:

r, g, b = hex2rgb(newpix)

newData.append((r,g,b,255))

digit += 1

else:

newData.append(item)

img.putdata(newData)

img.save(filename, "PNG")

return "Completed!"

return "Incorrect Image Mode, Couldn't Hide"

def retr(filename):

img = Image.open(filename)

binary = ''

if img.mode in ('RGBA'):

img = img.convert('RGBA')

datas = img.getdata()

for item in datas:

digit = decode(rgb2hex(item[0], item[1], item[2]))

if digit == None:

pass

else:

binary = binary + digit

if (binary[-16:] == '1111111111111110'):

print("Success")

return bin2str(binary[:-16])

return bin2str(binary)

return "Incorrect Image Mode, Couldn't Retrieve"

parser = optparse.OptionParser('usage %prog ' + '-e/-d <target file>')

parser.add\_option('-e', dest='hide', type='string', help='target picture path to hide text')

parser.add\_option('-d', dest='retr', type='string', help='target picture path to retrieve text')

(options, args) = parser.parse\_args()

if (options.hide != None):

text = input("Enter a message to hide: ")

print(hide(options.hide, text))

elif (options.retr != None):

print(retr(options.retr))

else:

print(parser.usage)

exit(0)