Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Лабораторная работа № 8

“Стеганография”

Выполнил студент гр. 753504

Горбачёнок К. Н.

Проверил

Протько М.И.

Минск, 2020

# Постановка задачи

Реализовать программное средство формирования электронной цифровой подписи на основе эллиптических кривых ECDSA и программное средство и программное средство, реализующее простой подход к шифрованию/дешифрованию с использованием *эллиптических кривых*.

# Описание

Стеганография (от греч. στεγανός — скрытый и греч. γράφω — пишу, буквально «тайнопись») — это наука о скрытой передаче информации путём сохранения в тайне самого факта передачи.

* отличие от криптографии, которая скрывает содержимое секретного сообщения, стеганография скрывает само его существование. Стеганографию обычно используют совместно с методами криптографии, таким образом, дополняя её.

Классификация стеганографических методов защиты информации

* Классическая стеганография
* Компьютерная стеганография
* Цифровая стеганография

Цифровая стеганография — направление классической стеганографии, основанное на сокрытии или внедрении дополнительной информации в цифровые объекты. Как правило, данные объекты являются мультимедиа-объектами (изображения, видео- или аудио-файлы, текстуры 3D-объектов), внесение изменений в которые вызывает лишь незначительные искажения, находящиеся ниже порога чувствительности среднестатистического человека, что не приводит к заметным изменениям этих объектов.

В рамках лабораторной работы необходимо реализовать программные средства сокрытия текстового сообщения в изображение на основе метода сокрытия частной области изображения.

Анализ тенденций развития технологий, использующихся для обеспечения безопасности информации вообще и, в частности, для защиты авторских прав в области программного обеспечения, показывает, что *применение компьютерной стеганографии наряду с методами, традиционно применяемыми для защиты программных продуктов, увеличивает мощность механизмов защиты* .

Анализ стеганографических методов защиты информации, технологий и стеганографических средств защиты интеллектуальной собственности, представленных на рынке программного обеспечения, а также проблем, связанных с применением данных методов, позволяет сделать следующие выводы.

1. В настоящее время рынок программных средств защиты интеллектуальной собственности только складывается.
2. Несмотря на низкую стойкость цифровых меток, цифровые стеганографические технологии и системы успешно применяются на практике для защиты авторских прав создателей мультимедийной продукции при распространении их продуктов в компьютерных сетях и на цифровых носителях: компакт-дисках, цифровых музыкальных дорожках и видео.
3. В настоящее время среди производителей программного обеспечения только разработчики и издатели мультимедиа обладают средствами, обеспечивающими на том или ином уровне защиту прав авторов.
4. Не все существующие методы компьютерной стеганографии могут быть использованы для защиты авторских прав на компьютерные программы.
5. Наиболее развитые методы и алгоритмы компьютерной стеганографии, относящиеся к цифровой стеганографии, не могут применяться для внедрения скрытой информации, идентифицирующей автора, в исполняемые файлы программ.
6. Стеганографические методы защиты авторских прав на компьютерные программы сегодня недостаточно развиты.

# Блок-схема алгоритма

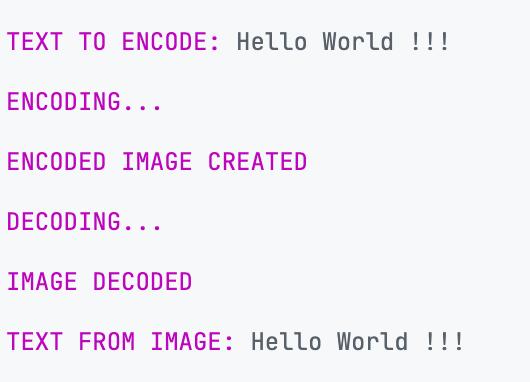


# Результаты работы программы



***Исходное изображение***

******

******

***Результат работы программы***

# Прграммный код

from PIL import Image

def genData(data):

newd = []

for i in data:

newd.append(format(ord(i), '08b'))

return newd

def modPix(pix, data):

datalist = genData(data)

lendata = len(datalist)

imdata = iter(pix)

for i in range(lendata):

pix = [value for value in imdata.\_\_next\_\_()[:3] +

imdata.\_\_next\_\_()[:3] +

imdata.\_\_next\_\_()[:3]]

for j in range(0, 8):

if (datalist[i][j] == '0' and pix[j] % 2 != 0):

pix[j] -= 1

elif (datalist[i][j] == '1' and pix[j] % 2 == 0):

if(pix[j] != 0):

pix[j] -= 1

else:

pix[j] += 1

if (i == lendata - 1):

if (pix[-1] % 2 == 0):

if(pix[-1] != 0):

pix[-1] -= 1

else:

pix[-1] += 1

else:

if (pix[-1] % 2 != 0):

pix[-1] -= 1

pix = tuple(pix)

yield pix[0:3]

yield pix[3:6]

yield pix[6:9]

def encode\_enc(newimg, data):

w = newimg.size[0]

(x, y) = (0, 0)

for pixel in modPix(newimg.getdata(), data):

newimg.putpixel((x, y), pixel)

if (x == w - 1):

x = 0

y += 1

else:

x += 1

def encode(message, img\_path, encode\_img\_path):

image = Image.open(img\_path, 'r')

if (len(message) == 0):

raise ValueError('Message is empty')

newimg = image.copy()

encode\_enc(newimg, message)

newimg.save(encode\_img\_path, str(encode\_img\_path.split(".")[1].upper()))

def decode(img\_path):

image = Image.open(img\_path, 'r')

data = ''

imgdata = iter(image.getdata())

while (True):

pixels = [value for value in imgdata.\_\_next\_\_()[:3] +

imgdata.\_\_next\_\_()[:3] +

imgdata.\_\_next\_\_()[:3]]

binstr = ''

for i in pixels[:8]:

if (i % 2 == 0):

binstr += '0'

else:

binstr += '1'

data += chr(int(binstr, 2))

if (pixels[-1] % 2 != 0):

return data

from helpers.colored\_text import colored\_text

from helpers.file\_info import read\_text

from algorithms import sten

BASE\_PATH = "resources/lab8/"

IMG\_PATH = BASE\_PATH + "img.png"

ENCODE\_PATH = BASE\_PATH + "img\_enc.png"

def encode():

text\_to\_encode = "Hello World !!!"

print(colored\_text("TEXT TO ENCODE:"), text\_to\_encode)

print(colored\_text("ENCODING..."))

sten.encode(message=text\_to\_encode, img\_path=IMG\_PATH, encode\_img\_path=ENCODE\_PATH)

print(colored\_text("ENCODED IMAGE CREATED"))

def decode():

print(colored\_text("DECODING..."))

text = sten.decode(img\_path=ENCODE\_PATH)

print(colored\_text("IMAGE DECODED"))

print(colored\_text("TEXT FROM IMAGE:"), text)

def main():

encode()

decode()

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Вывод

В ходе написания лабораторной работы было изучено понятие стенографии, а также написана программная реализация кодирования информации в изображение.