Исследование стеганографического метода на основе преобразования наименее значащих битов

Выполнил: Макаров Алексей Игоревич, 3 курс 4 группа 2 подгруппа

2024

**Теоритические сведения**

Стеганографическая система (stegosystem, стегосистема или стеганосистема) – совокупность средств и методов, которые используются для формирования скрытого канала передачи (или хранения) информации.

Основные компоненты стеганосистемы:

• контейнер С (файл-контейнер или электронный документ произвольного формата), в котором размещается (осаждается, скрывается) тайное сообщение М; именно контейнер является упомянутым скрытым каналом;

• тайное сообщение М, осаждаемое в контейнер для передачи или хранения (например, с целью доказательства или защиты авторских прав на документ-контейнер; здесь речь может идти о невидимых цифровых водяных знаках (ЦВЗ));

• ключи, или ключевая информация, K системы, выполняющие ту же функцию, что и криптографические ключи; ключей может быть несколько, в соответствии с этим современные стеганосистемы характеризуют как многоключевые: один ключ отождествляется с методом встраивания/извлечения тайной информации, другой – с выбором элементов (например, битов) контейнера для его модификации при осаждении тайной информации, третий – для предварительного (перед встраиванием) преобразования тайной информации (например, на основе помехоустойчивого кодирования, сжатия или зашифрования) и т. д.;

• контейнер со встроенным сообщением, или стеганоконтейнер, S, который передается по открытому каналу, также являющемуся важным компонентом анализируемой системы; стеганоконтейнер будем именовать также стеганосообщением;

• для полноты упомянем также субъектов системы: отправителя и получателя.

В зависимости от формата документа-контейнера цифровую (или компьютерную) стеганографию подразделяют на классы:

• аудиостеганография;

• видеостеганография;

• графическая стеганография;

• текстовая стеганография;

• и др.

При построении стеганосистемы должны учитываться следующие основные положения:

• свойства контейнера должны быть модифицированы так, чтобы изменение невозможно было выявить при визуальном контроле; это требование определяет качество сокрытия внедряемого сообщения: для обеспечения беспрепятственного прохождения стеганосообщения по каналу связи оно никоим образом не должно привлечь внимание атакующего;

• противник (интруз) имеет полное представление о стеганографической системе и деталях ее реализации; единственной информацией, которая остается ему неизвестной, является ключ, с помощью которого только его держатель может установить факт присутствия и содержание скрытого сообщения;

• если противник каким-то образом узнает о факте существования скрытого сообщения, это не должно позволить ему извлечь подобные сообщения до тех пор, пока ключ хранится в тайне;

• потенциальный противник должен быть лишен каких-либо технических и иных преимуществ в распознавании или раскрытии содержания тайных сообщений.

Метод НЗБ основывается на ограниченных способностях зрения или слуха человека, вследствие чего людям тяжело различать незначительные вариации цвета или звука.

Рассмотрим это на примере 24-битного растрового RGB-изображения. Как известно, каждая точка кодируется тремя байтами. Каждый байт определяет интенсивность красного (Red), зеленого (Green) и синего (Blue) цветов. Совокупность интенсивностей цвета в каждом из трех каналов определяет оттенок пикселя.

Младшие биты дают незначительный «вклад» в изображение по сравнению со старшими. Замена одного или даже нескольких младших битов для человеческого глаза будет почти незаметна, поскольку реально человек может различать около полторы сотни цветовых оттенков.

**Ход работы**

Вариант 8

В данной работе было необходимо реализовывать собственное приложение, в котором должен быть реализован метод НЗБ..

Код программы представлен в листинге 1.

|  |
| --- |
| from PIL import Image from stegano import lsb import numpy as np import matplotlib.pyplot as plt  def generate\_message():  return "Lexa"  def embed\_message\_LSB(container\_file, message):  *# Метод размещения битового потока с использованием младших значащих битов  secret* = lsb.hide(container\_file, message)  *secret*.save("guitar.jpg")  print("Сообщение успешно осаждено методом LSB.")  def embed\_message\_with\_password(container\_file, message, password):  *# Метод размещения сообщения с использованием пароля  binary\_message* = ''.join([format(ord(*char*), '08b') for *char* in message])  *binary\_message* += '00000000' *# Null byte to indicate end of message   img* = Image.open(container\_file)  *img\_data* = np.array(*img*)  *stego\_image\_data* = *img\_data*.copy()   *# Преобразование пароля в бинарный формат  password\_binary* = ''.join(format(ord(*char*), '08b') for *char* in password)  *password\_length* = len(*password\_binary*)   *# Осаждение сообщения с использованием пароля  idx* = 0  for *i* in range(len(*img\_data*)):  for *j* in range(len(*img\_data*[0])):  for *k* in range(3): *# 3 цветовых канала* if *idx* < len(*binary\_message*):  *# Применяем XOR шифрование с битами пароля к битам сообщения  stego\_image\_data*[*i*][*j*][*k*] = (*img\_data*[*i*][*j*][*k*] & ~1) | (int(*binary\_message*[*idx*]) ^ int(*password\_binary*[*idx* % *password\_length*]))  *idx* += 1  else:  break   *stego\_image* = Image.fromarray(*stego\_image\_data*)  *stego\_image*.save("embedded\_with\_password.png")  print("Сообщение успешно осаждено с использованием пароля.")   *# Извлечение сообщения с использованием пароля  img* = Image.open(container\_file)  *img\_data* = np.array(*img*)   *# Преобразование пароля в бинарный формат  password\_binary* = ''.join(format(ord(*char*), '08b') for *char* in password)  *password\_length* = len(*password\_binary*)   *extracted\_message* = ""  *idx* = 0  for *i* in range(len(*img\_data*)):  for *j* in range(len(*img\_data*[0])):  for *k* in range(3): *# 3 цветовых канала* if *idx* < 8: *# Пропускаем первые 8 бит (маркер конца сообщения)  idx* += 1  continue  *# Применяем XOR шифрование с битами пароля к битам сообщения  extracted\_bit* = (*img\_data*[*i*][*j*][*k*] & 1) ^ int(*password\_binary*[*idx* % *password\_length*])  *extracted\_message* += str(*extracted\_bit*)  *idx* += 1   *# Находим маркер конца сообщения и извлекаем только биты сообщения  end\_idx* = *extracted\_message*.find("00000000")  *extracted\_message* = *extracted\_message*[8:*end\_idx*]   *# Преобразуем бинарное сообщение в строку символов ASCII  extracted\_text* = ""  for *i* in range(0, len(*extracted\_message*), 8):  *extracted\_text* += chr(int(*extracted\_message*[*i*:*i*+8], 2))   return *extracted\_text* def generate\_color\_matrix(container\_file):  *# Формирование цветовой матрицы  img* = Image.open(container\_file)  *img\_data* = np.array(*img*)    *# Создаем пустую матрицу для каждого канала цвета (R, G, B)  color\_matrices* = { "R": np.zeros\_like(*img\_data*[:,:,0]),   "G": np.zeros\_like(*img\_data*[:,:,1]),   "B": np.zeros\_like(*img\_data*[:,:,2]) }    *# Заполняем матрицы значениями младших битов каждого канала* for *channel*, *matrix* in *color\_matrices*.items():  *matrix*.flat = *img\_data*[:,:,["R", "G", "B"].index(*channel*)].flatten() & 1    *# Отображение цветовых матриц  fig*, *axs* = plt.subplots(1, 3, figsize=(15, 5))  for *i*, (*channel*, *matrix*) in enumerate(*color\_matrices*.items()):  *axs*[*i*].imshow(*matrix*, cmap='gray')  *axs*[*i*].set\_title(f"Channel: {*channel*}")  *axs*[*i*].axis('off')  plt.show()  *# Генерация сообщения и выбор пароля* message = generate\_message() password = "mysecretpassword"  *# Выбор и применение методов размещения сообщения* embed\_message\_LSB("koala.jpg", message) embed\_message\_with\_password("koala.jpg", message, password)  *# Формирование цветовых матриц* generate\_color\_matrix("koala.jpg")  from stegano import lsb  *# реализовать осаждаемое/извлекаемое сообщения* def embed\_message(container\_file, message):  *secret* = lsb.hide(container\_file, message)  *secret*.save("embedded\_" + container\_file)  return *secret* def extract\_message(embedded\_container\_file):  *message* = lsb.reveal(embedded\_container\_file)  return *message* container\_file = 'koala.jpg' message\_own\_info = 'Lexa' embedded\_container = embed\_message(container\_file, message\_own\_info) extracted\_message = extract\_message(embedded\_container) print("Извлеченное сообщение:", extracted\_message) |

Листинг 1 – Программная реализация приложения

Результат работы программы представлен на рисунке 1:

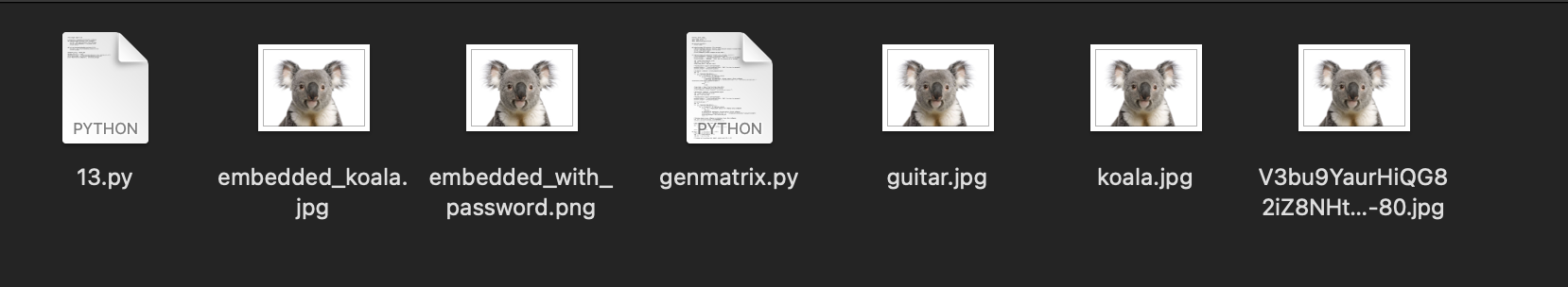


Рисунок 1 – Результат работы приложения

**Вывод**: таким образом, были закреплены теоретические знания из области стеганографического преобразования информации, моделирования стеганосистем, классификации и сущности методов цифровой стеганографии.