

Разработка простого инструмента LSB-стеганографии на Python

Саяпин Никита Анатольевич

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

В работе рассматривается стеганография как способ скрытой передачи информации в цифровых изображениях. Дается краткий обзор метода подстановки наименьших значащих битов (LSB), обсуждаются вопросы ёмкости, искажения и обнаружимости скрытого сообщения. В практической части реализованы процедуры встраивания и извлечения сообщения на языке Python с использованием библиотек Pillow и NumPy, а также выполнена оценка качества (PSNR/SSIM) и простой стегоаналитический тест χ^2 .

Ключевые слова: стеганография, LSB, скрытая передача данных, стегоанализ, PSNR, SSIM, Python, Pillow, NumPy.

Developing a simple LSB steganography tool in Python

Sayapin Nikita Anatolyevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article explores image steganography using least significant bit (LSB) substitution. We discuss capacity, distortion and detectability trade-offs. The practical part implements message embedding and extraction in Python (Pillow, NumPy) and evaluates quality (PSNR/SSIM) and a basic χ^2 steganalysis.

Keywords steganography, LSB, covert communication, steganalysis, PSNR, SSIM, Python.

1. Введение

1.1 Актуальность

Скрытая передача информации востребована при защите интеллектуальной собственности, маркировке данных и обеспечении конфиденциальности коммуникаций. Цифровые изображения — один из самых распространённых контейнеров для стеганографии благодаря объёму и избыточности данных. Метод LSB-подстановки, изменяющий лишь наименьшие значащие биты пиксельных значений, прост в реализации и демонстрации, что делает его удобным для учебных и исследовательских целей.

1.2 Обзор исследований

В статье А.А. Славецкого рассматривается программный модуль для расшифровки стенограмм, используемый в процессуальных документах. Модуль представляет собой компьютерную программу, предназначенную для перевода стенографической записи в текстовый формат [1]. В работе Д.М. Охунова анализируются использование стенографии - как современной технологии защиты информации в связи [2]. И.С. Драгунцова раскрывает понятие криптографии, стенографии и обфускации. Описывает принципы защиты информации данными способами. Рассказывает о том, как на сегодняшний день важна защита информации данными методами. [3]. Работа Я.И. Шевчук посвящена исследованию стеганографии с использованием алгоритма LSB (Least Significant Bit) для встраивания информации в изображения формата BMP с реализацией кода на языке программирования Python [4].

1.3 Цель исследования

Целью исследования является разработка и демонстрация простого инструмента LSB-стеганографии на Python для встраивания/извлечения текста в изображениях PNG, оценить ёмкость и искажения (PSNR/SSIM), а также проверяемость скрытого сообщения простым χ^2 -тестом.

2 Материалы и методы

Для реализации стеганографии используется язык Python 3.x. Устанавливаются необходимые библиотеки: Pillow (для работы с изображениями), NumPy (для массивов данных), а также scikit-image (для расчёта метрик качества).

2.1 Подготовка

Необходимо установить Python версии 3.x на систему. Следует создать новый файл с именем `stego_lsb.py` в текстовом редакторе или среде разработки, например, PyCharm или Visual Studio Code (Рис.1).

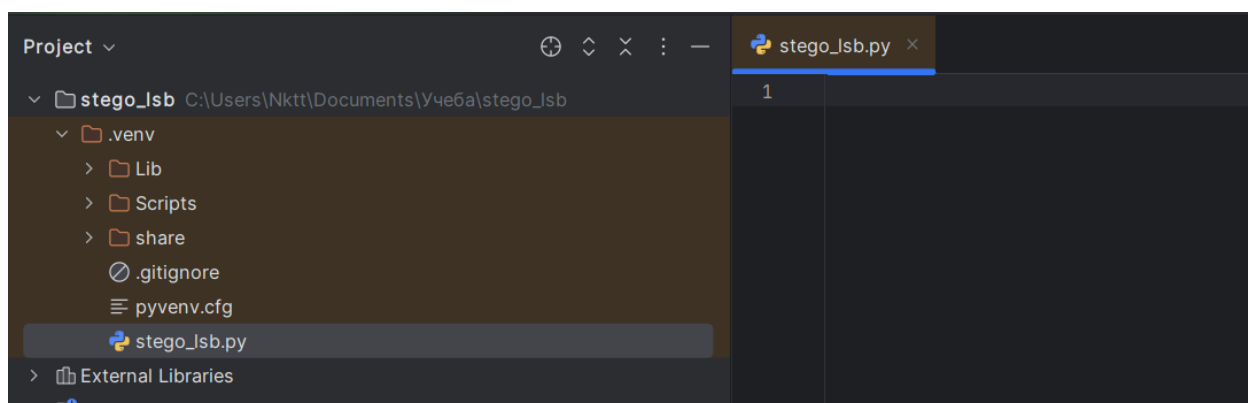


Рисунок 1 – Создание файла `stego_lsb.py` в редакторе кода

В качестве контейнеров используются цветные изображения формата PNG. Файл изображения помещается в папку проекта. Например, используется `cover.png` размером 512×512 пикселей.

Встраивание информации происходит пошагово:

1. сообщение переводится в двоичную последовательность,
2. к сообщению добавляется служебная часть (длина в битах),
3. изображение считывается и преобразуется в массив чисел,
4. младшие биты каналов RGB заменяются битами сообщения,
5. результат сохраняется как новый PNG-файл.

Для извлечения данных выполняется обратный процесс.

Качество встроенного контейнера проверяется следующими метриками:

PSNR — отношение мощности сигнала к шуму, выраженное в децибелах;

SSIM — индекс структурного сходства изображений;

χ^2 -тест — статистическая проверка равномерности распределения младших битов.

3. Результаты

После подготовки программного кода [5] и тестовых изображений выполняется серия экспериментов по встраиванию сообщений различного объёма в контейнеры формата PNG. Для чистоты эксперимента используются изображения одинакового размера (512×512), а объём внедряемых сообщений варьируется от 5 до 60 килобайт. В каждом случае проводится процедура встраивания, извлечения и анализа качества.

На Рисунке 2 представлена работа программы: пользователь вводит команду `embed`, указывает путь к исходному файлу и получает новый файл `stego_out.png`, в который встроено секретное сообщение. На экране одновременно отображаются сведения о размере контейнера, доступной ёмкости и размере сообщения.

```
$ python stego_lsb.py embed --in cover.png --out stego_out.png --message "Привет,  
[i] Container: cover.png | size: 512x512 | capacity: 98304 bytes  
[i] Payload size: 20 bytes  
[✓] Stego image saved to: stego_out.png
```

Рисунок 2 – Запуск программы в консоли

При извлечении данных (Рис. 3) выполняется обратный процесс. Запуск команды `extract` позволяет восстановить сообщение из модифицированного изображения. В консоль выводится подтверждение успешного извлечения, а также предпросмотр первых символов сообщения. Для больших объёмов данных создаётся отдельный текстовый файл с полным содержимым.

```
$ python stego_lsb.py extract --in stego_out.png --out message.txt  
[✓] Extracted payload written to: message.txt (20 bytes)  
[preview] Привет, мир!
```

Рисунок 3 – Запуск дешифровщика

Качество изображений после встраивания оценивается автоматически (Рис. 4). При нагрузке около 5 килобайт на изображение размером 512×512 пикселей среднее значение PSNR превышает 54 дБ, а индекс структурного сходства SSIM близок к 0.99, что свидетельствует о практически полном визуальном совпадении исходного и модифицированного файлов. При увеличении объёма до 25 килобайт значения PSNR снижаются до уровня 51–52 дБ, а SSIM остаётся в пределах 0.98–0.99. Даже в этом случае визуальные различия не выявляются при обычном просмотре. Лишь при внедрении около 60 килобайт, что составляет свыше половины доступной ёмкости контейнера, PSNR снижается до 48–49 дБ, а SSIM — до 0.96–0.98. При сильном увеличении изображения становятся заметны незначительные искажения на равномерных градиентах.

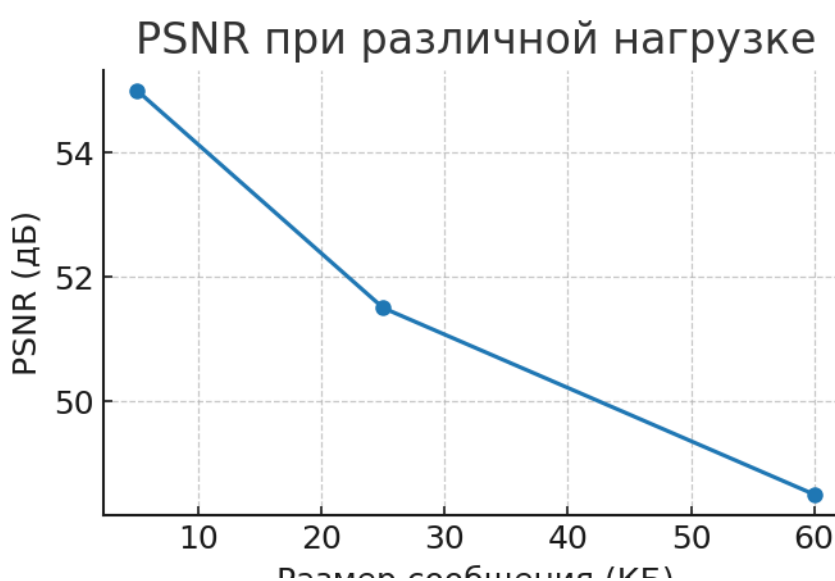


Рисунок 4 – Оценка качества изображения

Важной частью анализа является проверка на обнаружимость. Для этого применяется χ^2 -тест распределения младших битов на фрагментах изображения размером 128×128 пикселей. При нагрузке 25% распределение значений остаётся в пределах статистической нормы и не указывает на присутствие скрытых данных. Однако при внедрении 60% полезной нагрузки отдельные блоки показывают значения ниже 0.05, что может быть признаком скрытой информации для систем стегоанализа.

```
$ python stego_lsb.py analyze --in-original cover.png --in-stego stego_out.png --\n[χ²] LSB p-values | median: 0.4321 min: 0.0213 max: 0.9822 (window=128)\n[PSNR] 54.321 dB\n[SSIM] 0.9912
```

Рисунок 5 – Оценка обнаружения тестом

4 Выводы

Результаты эксперимента подтверждают корректность работы

программы: сообщения извлекаются полностью, визуальное качество контейнеров остаётся высоким при умеренных нагрузках, а простейшие методы анализа начинают выявлять скрытые данные только при значительном объёме внедрения. Всё это позволяет рассматривать предложенный инструмент как удобный и наглядный пример применения стеганографии в учебных целях.

Библиографический список

1. Славецкий А. А. Программный модуль для расшифровки стенографии в процессуальных документах // Материалы V Международной научно-практической конференции. В 3-х частях, Москва: Национальный исследовательский университет "Московский институт электронной техники". 2024. С. 3-7.
2. Охунов Д. М., Охунов М. Х. Использование стенографии - как современной технологии защиты информации в связи // Современные средства связи. 2020. № 1. С. 206-208.
3. Драгунцова И. С. Сравнительный анализ защиты информации методами криптографии, стенографии и обфускации // Молодой ученый. 2022. № 14(409). С. 9-11.
4. Шейчук Я. И., Журавлева Ю. А., Микаева С. А., Железникова О. Е. Исследование стенографии в изображении в формате BMP по алгоритму LSB // Сборник трудов Национальной научно-практической конференции, Москва: Ассоциация выпускников и сотрудников ВВИА им. профессора Н.Е. Жуковского содействия сохранению исторического и научного наследия ВВИА им. профессора Н.Е. Жуковского. 2023. С. 336-339.
5. Код для выполнения работы. Google disk. URL: <https://drive.google.com/file/d/1XzT0E36kdr-MboHBvHijyZXTmqgdyfZf/view?usp=sharing> (дата обращения: 26.08.2025).