УДК 004.932

Какабьян К.С.,
студент
4 курс, факультет №1 Кибербезопасности и управления
Поволжский государственный университет
телекоммуникаций и информатики
Россия, г. Самара

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ СТЕГАНОГРАФИИ PM1 И LSB

Аннотация: *В данной статье рассматриваются методы стеганографии в пространственной области: «Наименьший значащий бит (НЗБ)» и «плюс/минус один (±1)». Проведено сравнение ±1 и НЗБ по показателям встраивания, производительности, и качества выходного изображения. Оба алгоритма ±1 и НЗБ реализованы на языке Python с использованием библиотеки Pillow. Алгоритмы принимают исходное изображение и секретное сообщение в качестве входных данных. Результатом работы программ является изображение с встроенным в него секретным сообщением.*

Ключевые слова: *стеганография, информационная безопасность, анализ методов, изображения, сокрытие данных, статистика, встраивание, извлечение, оценка алгоритмов.*

COMPARATIVE ANALYSIS
OF PM1 AND LSB STEGANOGRAPHY METHODS

Abstract: *This article examines steganography methods in the spatial domain: Least Significant Bit (LSB) and Plus-Minus 1 (PM1). A comparison of PM1 and LSB was conducted based on embedding capacity, performance, and output image quality. Both PM1 and LSB algorithms were implemented in Python using the Pillow library. The algorithms take the original image and secret message as input data. The output of the programs is an image with a secret message embedded in it.*

Keywords: *steganography, information security, methods analysis, images, data hiding, statistics, embedding, extraction, algorithm evaluation.*

Стеганография — это наука, изучающая способы скрытой передачи информации. В качестве носителя информации (стегоконтейнера) может использоваться изображение, аудио или видеофайл. Цель стеганографии - сделать встраиваемое сообщение незаметным и обеспечить его извлечение только предполагаемым получателем. Стеганография может применяться в различных областях, таких как создание цифровых водяных знаков, защищенная коммуникация, защита данных и управление цифровыми правами (DRM).

Сокрытие информации в изображении алгоритмами PM1 и LSB осуществляется путем изменения битовых значений пикселей. PM1 работает путем добавления или вычитания единицы из каждого значения пикселя в изображении, в зависимости от того, является ли соответствующий бит секретного сообщения единицей или нулем. LSB работает путем изменения наименее значимого бита каждого пикселя в изображении для кодирования секретного сообщения.

Алгоритмы PM1 и LSB могут быть реализованы следующим образом:

```python
from PIL import Image
def encode_lsb(img, message):
    width, height = img.size
    message_bin = ''.join(format(ord(i), '08b') for i in message)
    if len(message_bin) > width * height * 3:
        raise ValueError('Слишком длинное сообщение')
    message_bin += '00000000'
    message_bits = list(message_bin)
    index = 0
    for y in range(height):
        for x in range(width):
            r, g, b = img.getpixel((x, y))
            if index < len(message_bits):
                r &= ~1;g &= ~1;b &= ~1
                r |= int(message_bits[index]);index += 1
            if index < len(message_bits):
                g |= int(message_bits[index]);index += 1
            if index < len(message_bits):
                b |= int(message_bits[index]);index += 1
            img.putpixel((x, y), (r, g, b))
    return img
```

Рисунок 1. Функция кодирования сообщения методом LSB

```python
def decode_lsb(img):
    width, height = img.size
    message_bits = []
    for y in range(height):
        for x in range(width):
            r, g, b = img.getpixel((x, y))
            message_bits.append(str(r & 1))
            message_bits.append(str(g & 1))
            message_bits.append(str(b & 1))
    message_bin = ''.join(message_bits)
    try:
        end = message_bin.index('00000000')
    except ValueError:
        raise ValueError('Маркер конца сообщения не найден')
    message = ''.join(chr(int(message_bin[i:i+8], 2)) for i in range(0, end, 8))
    return message
```

Рисунок 2. Функция декодирования сообщения методом LSB

```
def encode_pm1(img, message):
    pixels = img.load()
    width, height = img.size
    binary_message = ''.join(format(ord(c), '08b') for c in message)
    binary_message += '00000000'
    if len(binary_message) > width * height:
        raise ValueError('Слишком длинное сообщение')
    message_index = 0
    for y in range(height):
        for x in range(width):
            if message_index >= len(binary_message):
                break
            r, g, b = pixels[x, y]
            if binary_message[message_index] == '0':
                if b % 2 != 0:b -= 1
            else:
                if b % 2 == 0:b += 1
            pixels[x, y] = (r, g, b)
            message_index += 1
    return img
```

Рисунок 3. Функция кодирования сообщения методом PM1

```
def decode_pm1(img):
    pixels = img.load()
    width, height = img.size
    message = ""
    for y in range(height):
        for x in range(width):
            r, g, b = pixels[x, y]
            if message.endswith('00000000'):break
            if b % 2 == 0:message += '0'
            else:message += '1'
    text = ""
    for i in range(0, len(message), 8):
        byte = message[i:i+8]
        text += chr(int(byte, 2))
    return text[:-1]
```

Рисунок 4. Функция декодирования сообщения методом PM1

```
def embed_message(img_file, msg, steg_method):
    # Загрузка стегоконтейнера
    img = Image.open(img_file).convert('RGB') # цветное изображение
    #img = Image.open(img_file).convert('L') # градации серого без α - канала
    if steg_method == 'LSB':img_steg = encode_lsb(img, msg)
    elif steg_method == 'PM1':img_steg = encode_pm1(img, msg)
    else:pass
    img_steg.save('stego_'+steg_method+'.bmp')
# Извлечение сообщения
def extract_message(steg_file, steg_method):
    img = Image.open(steg_file).convert('RGB') # цветное изображение
    #img = Image.open(img_file).convert('L') # градации серого без α - канала
    if steg_method == 'LSB':return decode_lsb(img)
    elif steg_method == 'PM1':return decode_pm1(img)
    else:pass
```

Рисунок 5. Функция встраивания сообщения в изображение

```
orig_message = input('Введите сообщение: ')
# Встраивание сообщения с использованием метода LSB
embed_message('input.bmp', orig_message, 'LSB')
# Извлечение сообщения с использованием метода LSB
message_lsb = extract_message('stego_LSB.bmp', 'LSB')
print('\nИзвлеченное сообщение (LSB):', message_lsb)
# Встраивание сообщения с использованием метода PM1
embed_message('input.bmp', orig_message, 'PM1')
# Извлечение сообщения с использованием метода PM1
message_pm1 = extract_message('stego_PM1.bmp', 'PM1')
print('\nИзвлеченное сообщение (PM1):', message_pm1)
```

Рисунок 6. Вызов функций с заданными параметрами

В качестве стегоконтейнеров используются цветные и полутоновые растровые изображения формата BMP с глубиной цвета 24 бита. В данной реализации стеганографии методом PM1 я использовал только синий канал изображения для скрытия сообщения, потому что он менее заметен для глаза человека, по сравнению с другими цветовыми каналами. *[1, с.25–29]*.

При использовании только синего канала, можно скрыть только один бит информации в каждом пикселе, что ограничивает вместимость стегоконтейнера для сообщения. В реализации метода LSB при встраивании сообщения модификации подвергается младший бит каждого канала RGB, следовательно в одном пикселе изображения возможно скрыть 3 бита информации.

Также следует отметить, что выбор конкретного цветового канала зависит от характеристик конкретного изображения, а также от особенностей атак, которым оно может подвергаться.

Методика исследования

В экспериментах в качестве стегоконтейнеров были использованы цветные и полутоновые растровые изображения формата BMP с глубиной цвета 24 бита и разрешением 512x512, 1024x1024 и 2048x2048 пикселей. В каждое изображения были встроены сообщения различного размера с использованием PM1 и LSB стеганографии. Качество полученных

изображений было вычислено с использованием метода PSNR. Для измерения времени шифрования и извлечения сообщений написан скрипт на языке Python, автоматически встраивающий сообщение в изображение с последующим извлечением.

Результаты исследования

Для сравнения методов PM1 и LSB, была проведена серия экспериментов на изображениях различного размера и разрешения со встраиваемыми сообщениями различной длины. В качестве метрик были использованы время встраивания, время извлечения сообщения и качество полученного изображения.

В ходе эксперимента определялись следующие параметры:

• Время встраивания сообщения: время, необходимое для встраивания сообщения в изображение-носитель с использованием выбранного метода.

• Искажение изображения: степень визуального искажения, внесенного процессом встраивания сообщения, измеренная пиковым отношением сигнал/шум (PSNR) между оригинальным и закодированным изображениями.

• Время извлечения сообщения: время, необходимое для извлечения сообщения из закодированного изображения с использованием выбранного метода.

Для каждого изображения и каждого метода стеганографии было проведено 10 запусков, время выполнения каждой итерации было замерено.

Результаты, полученные в ходе эксперимента представлены в таблице 1.

Таблица 1.

Сравнение производительности методов PM1 и LSB и степени искажения изображений

Метод	Разрешение изображения	Размер сообщения	Время встраивания, мс	PSNR, дБ	Время извлечения, мс
PM1	512x512	100 байт	18	57,26	15
PM1	1024x1024	100 байт	33	57,10	27
PM1	2048x2048	100 байт	121	56,77	115
PM1	512x512	1 КБ	82	43,94	69
PM1	1024x1024	1 КБ	193	43,59	161
PM1	2048x2048	1 КБ	749	42,67	647
LSB	512x512	100 байт	9	57,26	11
LSB	1024x1024	100 байт	18	57.10	21
LSB	2048x2048	100 байт	75	56.77	91
LSB	512x512	1 КБ	45	43.94	54
LSB	1024x1024	1 КБ	102	43.59	121
LSB	2048x2048	1 КБ	396	42.67	473

Произведем визуальную оценку качества изображений, обработанных алгоритмами PM1 и LSB различных разрешений и сравним результаты.

Качество изображения определяется степенью искажений, внесенных стеганографическим алгоритмом. Для расчета визуального оценочного балла используем пиковое отношение сигнала к шуму (PSNR) и индекс структурного сходства (SSIM).

Пиковое отношение сигнала к шуму (PSNR) измеряет соотношение между максимально возможной мощностью сигнала и мощностью помех, которые влияют на достоверность его представления.

Это популярный показатель качества изображения, определяемый как:

$$\mathrm{P}SNR = 10\, log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20\, log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right),$$

где

MAX_I - максимальное значение яркости пикселя,

MSE - среднеквадратичная ошибка между оригинальным и стеганографическим изображением.

Индекс структурного сходства (SSIM) измеряет сходство между двумя изображениями. Он сравнивает структурную информацию оригинального и стеганографического изображений, включая яркость, контрастность и структуру расположения пикселей.

Определяется как:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

где x и y - оригинальное и стеганографическое изображения соответственно, μ_x и μ_y - средние значения x и y, σ_x и σ_y- стандартные отклонения x и y, σ_{xy}- ковариация x и y, а c_1 и c_2 - переменные, стабилизирующие деление со слабым знаменателем.

Затем визуальный оценочный балл (VE) вычисляется как гармоническое среднее PSNR и SSIM:

$$VE = \frac{2 \cdot (PSNR \cdot SSIM)}{(PSNR + SSIM)}$$

Эта формула усредняет PSNR и SSIM при определении визуального оценочного балла. Более высокий визуальный оценочный балл указывает на

меньшее искажение и лучшее качество изображения. Результаты расчетов представлены в таблице 2.

Таблица 2.

Суммарные оценки визуального качества алгоритмов

Разрешение изображения	Оценка визуального качества PM1	Оценка визуального качества LSB
512 x 512	0.72	0.88
1024 x 1024	0.65	0.83
2048 x 2048	0.61	0.80

Статистический анализ с использованием метрик PSNR и SSIM показал, что PM1 вносит больше искажений в изображение-носитель, что делает его уязвимым к обнаружению алгоритмами стегоанализа. LSB, с другой стороны, производит меньше искажений и обеспечивает лучшее качество для всех разрешений изображений, что повышает безопасность использования алгоритма.

Также были проведены замеры производительности обоих методов на разных типах изображений, включая полутоновые и цветные изображения. Время встраивания информации алгоритмом LSB примерно в 2 раза меньше по сравнению с методом PM1. Извлечение сообщения с помощью PM1 занимает в среднем на 10-15% больше времени по сравнению с LSB. Качество полученных изображений с использованием PM1 метода стеганографии оказалось ниже, чем у LSB метода стеганографии, особенно при встраивании больших сообщений. Для изображений разрешением 512x512 пикселей качество изображений с использованием PM1 метода стеганографии было на 15-20% ниже, чем у LSB метода стеганографии.

Заключение

Исходя из результатов исследования, можно сделать вывод о том, что PM1 метод уступает LSB методу стеганографии в качестве полученного изображения и времени на встраивание и извлечение сообщения.

Метод LSB рекомендуется использовать для стеганографических приложений, которые требуют высокого уровня безопасности и качества изображения, а PM1 - для приложений, требующих большей ёмкости встраивания.

При выборе метода стеганографии необходимо учитывать конкретные задачи, ограничивающие область его применения и требования к качеству изображений и времени выполнения операций с ними.

Использованные источники:

1. Грибунин В.Г. Стеганографические системы. Критерии и методическое обеспечение: учебно–методическое пособие / В.Г. Грибунин, В.Е. Костюков, А.П. Мартынов, Д.Б. Николаев, В.Н. Фомченко. // Саров: ФГУП "РФЯЦ–ВНИИЭФ", 2016. – 324 с.
2. Сойфер В.А. Компьютерная обработка изображений // Вестник РАН. – 2001. – Т. 71, № 2. – С. 119–129.
3. Аграновский А.В. Стеганография, цифровые водяные знаки и стегоанализ / А.В. Аграновский, А.В. Балакин, В.Г. Грибунин. – М.: Вузовская книга, 2009. – 217 с.
4. Евсютин О.О. Улучшенный алгоритм встраивания информации в сжатые цифровые изображения на основе метода PM1 / О.О. Евсютин, А.С. Кокурина, А.А. Шелупанов, И.И. Шепелев // Компьютерная оптика. – 2015. – Т. 39, № 4. – С. 572-581.

5. Федосеев В.А. Цифровые водяные знаки и стеганография. – Самара: СГАУ, 2015. – 128 с.
6. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика – К.: «МК-Пресс», 2006. – 288 с.
7. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. – М.: Солон-Пресс, 2002. – 272 с.