Правительство Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

# ОТЧЕТ
# О ПРАКТИЧЕСКОЙ РАБОТЕ № 1

по дисциплине «Методы защиты мультимедиа-данных»

Стенографическое встраивание информации в пространственную область цифровых изображений

Студент гр. БПИ196
Е.Н. Мосолков
«16» марта 2022 г.

Руководитель
МНС кафедры информационной безопасности киберфизических систем
_____А.С. Мельман
«___» _____ 2022 г.

Москва 2022

**СОДЕРЖАНИЕ**

## 1 Задание на практическую работу

Целью работы является приобретение навыков программной реализации стенографического встраивания информации в цифровые изображения

В рамках практической работы необходимо выполнить следующее:

1. Написать программную реализацию одного из следующих рассмотренных стеганографического методов для цифровых изображений по выбору студента: QIM, PVD, NMI;
2. Провести вычислительные эксперименты с полученной программной реализацией и сделать выводы об эффективности рассмотренного метода встраивания;
3. подготовить отчёт о выполнении работы.

Программа должна обладать следующей функциональностью:

1) при встраивании:

- принимать на вход цветное (RGB) изображение-контейнер;
- принимать на вход изображение для встраивания;
- рассчитывать показатели качества встраивания;

2) при извлечении:

- принимать на вход цветное стегоизображение;

3) осуществлять встраивание или извлечение информации по выбору пользователя.

Вычислительные эксперименты с полученной программной реализацией должны включать следующее:

1) встраивание максимально возможного для данного контейнера объёма информации с оценкой незаметности встраивания;

2) встраивание разного количества информации и сравнение незаметности встраивания;

3) извлечение встроенной информации из стегоизображений в условиях отсутствия постобработки и при наличии различных операций обработки изображений, оценка робастности;

4) сравнение гистограмм изображений до и после встраивания.

Отчёт должен содержать следующие составные части:

1) раздел с заданием;

2) раздел с краткой теоретической частью;

3) раздел с результатами работы программы;

4) раздел с результатами вычислительных экспериментов;

5) раздел с выводами о проделанной работе.

## 2 Краткая теоретическая часть

Стеганография – это наука о скрытой передаче и хранении информации таким образом, чтобы сам факт наличия этой информации был тайной для третьих лиц. Цифровая стеганография работает с цифровой информацией. В качестве контейнеров для секретных сообщений могут выступать самые разные цифровые объекты: мультимедиа-данные, исполняемые файлы, интернет-трафик, данные с датчиков «интернета вещей» и т.д.

Одним из наиболее распространённых типов контейнеров для стеганографического встраивания являются цифровые изображения. В первую очередь это связано с их высокой пространственной избыточностью: соседние пиксели изображения похожи друг на друга, и небольшое изменение оттенка отдельных пикселей останется незаметным для глаза человека. Существует большое количество различных методов встраивания дополнительной информации в цифровые изображения, отличающихся разными особенностями. Все существующие методы делятся на два больших класса: методы пространственного встраивания и методы частотного встраивания. Методы пространственного встраивания напрямую изменяют значения пикселей изображения, в то время как методы частотного встраивания вносят изменения в значения частотных коэффициентов, полученных после предварительного применения некоторого частотного преобразования к матрице пикселей.

### МЕТОД QIM

Метод модуляции индекса квантования (quantization index modulation, QIM) [3] заключается в модуляции значений элементов данных изображения в зависимости от встраиваемого бита сообщения. Встраивание бита $b$ осуществляется в пиксель контейнера $c$ по формуле

$$c' = q * \left[ \frac{c}{q} \right] + \frac{q}{2} * b,$$

где $c'$ – коэффициент после встраивания, $q$ – шаг квантования (чётное число), [x] - взятие целой части.

Для извлечения встроенной информации необходимо для каждого пикселя смоделировать ситуацию встраивания информации для нулевого и единичного битов и оценить, насколько результаты близки к реальному значению. Формула извлечения одного бита выглядит следующим образом:

$$b_i' = arg\,min ¿ c'' - c_p'' \vee ¿, p \in [0,1] ¿$$

Где $c''$ – это пиксель, содержащий бит сообщения, $c_0'' = q * [\frac{c''}{q}]$, $c_1'' = q * \left[ \frac{c''}{q} \right] + \frac{q}{2}$

### 3 Программная реализация

В рамках данной лабораторной работы был написан алгоритм QIM для шифрования и дешифрования текстовых сообщений в картинке.

Главную работу выполняют следующие функции:

```
def encode_QIM(arr, msg, q):
    byte_array = bt.bitarray()
    byte_array.frombytes(msg.encode('utf-8'))
    cnt = 0
    for i in arr:
        for j in i:
            for k, l in enumerate(j):
                if len(byte_array) > cnt:
                    j[k] = q * (l // q) + q / 2 + byte_array[cnt]
                else:
                    break
                cnt += 1
    return arr
```

*Рисунок 1. Функция для шифрования текста*

```
def decode_QIM(arr, bit_len, q):
    cnt = 0
    res = ''
    for i in arr:
        for j in i:
            for k in j:
                if cnt < bit_len:
                    c_0 = q * (k // q)
                    c_1 = q * (k // q) + q / 2
                    if abs(k - c_1) > abs(k - c_0):
                        res += '1'
                    else:
                        res += '0'
                cnt += 1
    return ''.join(chr(int(res[i*8:i*8+8],2)) for i in range(len(res)//8))
```

*Рисунок 2. Функция для дешифрования текста*

Первая функция принимает на вход матрицу изображения, сообщение для кодирования и шаг квантования, а выдает матрицу измененного изображения.

Вторая функция принимает размер зашифрованного сообщения, матрицу измененного изображения и шаг квантования, а выдает расшифрованное сообщение.

```python
image_path = 'image.png'
message = ''

while True:
    print('Input 1 to encode, or 2 to decode')
    choice = input('Your input: ')
    if choice == '1':
        message = input('Write a message to encode: ')
        matrix = np.array(Image.open('image.png'))
        matrix_encoded = encode_QIM(matrix, message, 2)
        im = Image.fromarray(matrix_encoded, "RGBA")
        im.save('encoded.png')
    elif choice == '2' and message != '':
        matrix_message = np.array(Image.open('encoded.png'))
        print(f'Decoded message: {decode_QIM(matrix_message, len(message) * 8, 2)}')
    else:
        print('Invalid argument')
```

*Рисунок 3. Код основной программы*

В качестве шага квантования было выбрано число 2.

При выборе опции 1, программа выполняет шифрование текста в картинку, загруженную в файл “image.png” и сохраняет новую картинку с сообщением в файл “encoded.png”

При выборе опции 2, программа считывает картинку из файла “encoded.png” и выводит на экран закодированное сообщение.

В качестве открытого ключа выступает длина сообщения

Весь код программы написан на языке python в среде Jupyter notebook.

## 4 Результаты экспериментов

При тестировании использовалось следующее изображение

*Рисунок 4. Изображение до кодирования*

Запускаем программу и выбираем опцию «1»

```
Input 1 to encode, or 2 to decode
Your input:  1
Write a message to encode:  hello world
Input 1 to encode, or 2 to decode
Your input:  2
Decoded message: hello world
```

*Рисунок 5. Пример выполнения шифрования и дешифрования текста с картинки*

Теперь смотрим на получившуюся картинку (рис. 6). Заметим, что видимых отличий нет. Алгоритм работает.

*Рисунок 6. Изображение после кодирования*

При тестировании программы на данных строки в 1000,  10000 ничего визуально не менялось кроме времени выполнения программы, сама картинка получается визуально не отличимой от оригинала.

## 5 Выводы о проделанной работе

В данной работе я изучил и применил на практике стенографические методы защиты информации, написал программную реализацию для метода QIM