Задачу встраивания и выделения сообщений из некоторой информационной последовательности – стегоконтейнера, выполняет стегосистема. Под стегосистемой понимают совокупность средств и методов, которые используются для формирования скрытого канала передачи информации – стегоканала.

Скрытая информация внедряется как в мультимедийный контент, так и в защищаемые программные продукты. Внесенные изменения могут рассматриваться как веские доказательства в ходе судебных разбирательств об авторских правах собственности или для доказательств факта нелегального копирования и распространения и зачастую имеют решающее значение.

Также они используются террористами для скрытого обмена информацией.

К другим задачам, которые могут быть решены с помощью стегоалгоритмов, можно отнести аутентификацию и идентификацию, т.е. определение подлинности передаваемой по открытым каналам связи информации, такой как, например, результаты дистанционного тестирования в системах электронного обучения.

Стегосистема состоит из следующих основных элементов, представленных на рисунке 1.

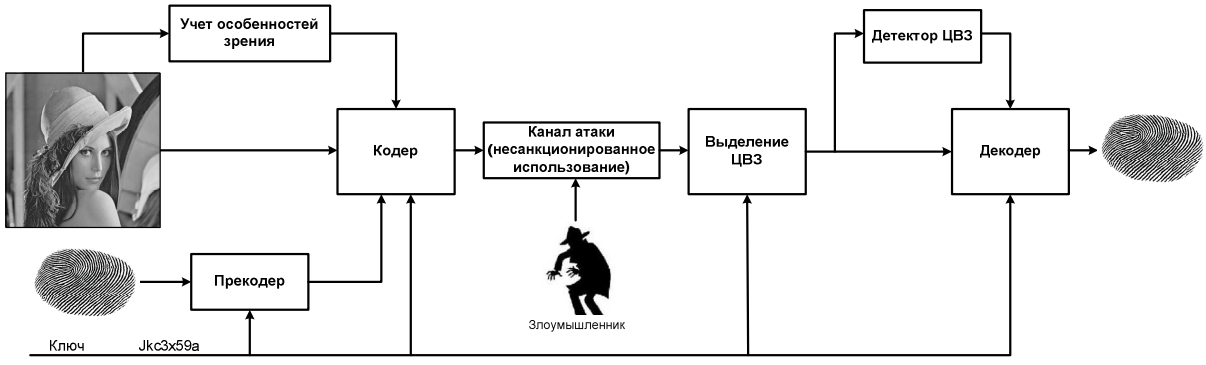


Рисунок 1 – Структурная схема стегосистемы цифровых водяных знаков.

Прекодер – устройство, предназначенное для преобразования скрываемого сообщения к виду, удобному для встраивания в стегоконтейнер.

Например, стегосообщение может представляться в виде последовательности бит или как двумерный массив бит, в случае, если стегосообщение представляет собой изображение, например, отпечаток пальца или логотип.

Прекодер также может выполнять дополнительное логическое кодирование с целью повышения помехоустойчивости или криптографической стойкости (стегосообщение дополнительно зашифровывается).

Стегокодер – устройство или программное обеспечение, предназначенное для осуществления встраивания скрытого сообщения в другие данные с учетом их модели.

В стегоалгоритмах обычно используются те же преобразования, что и в современных алгоритмах сжатие. Например, для изображений в роли таких преобразований выступают дискретное косинусное преобразование или вейвлет-преобразование. Встраивание информации может производиться в исходный контейнер с последующим сжатием, либо одновременно с осуществлением сжатия контейнера, либо в уже сжатый каким-либо алгоритмом контейнер.

Стегодетектор – устройство или программное обеспечение, предназначенное для определения наличия стегосообщения, которое может сильно измениться в силу активных или пассивных атак. Это изменение может быть обусловлено влиянием ошибок в канале связи, операций геометрической обработки изображения, сжатия и т.п.

Декодер – устройство, восстанавливающее скрытое сообщение.

Цифровые водяные знаки делятся на видимые и невидимые. Видимые ЦВЗ довольно просто удалить или заменить. Для этого могут быть использованы графические или текстовые редакторы. Невидимые ЦВЗ представляют собой встраиваемые в компьютерные файлы вставки, не воспринимаемые человеческим глазом.

Цифровые изображения характеризуются большой психовизуальной избыточностью и представляют из себя матрицу пикселей – единичных

элементов изображения. Пиксели изображения кодируются 8-разрядными значениями. Младший значащий бит такого 8-разрядного значения несет в себе меньше всего информации.

Человеческая зрительная система невосприимчива к мелким деталям, которые определяются изменениями в младшем значащем бите. Эти особенности человеческого зрения используются, например, при разработке алгоритмов сжатия аудио, изображений и видео. Другими словами, младший значащий бит можно использовать для встраивания информации. Таким образом, для полноцветного изображения объем встраиваемого стегосообщения может составлять более $\frac{1}{8}$ объема контейнера.

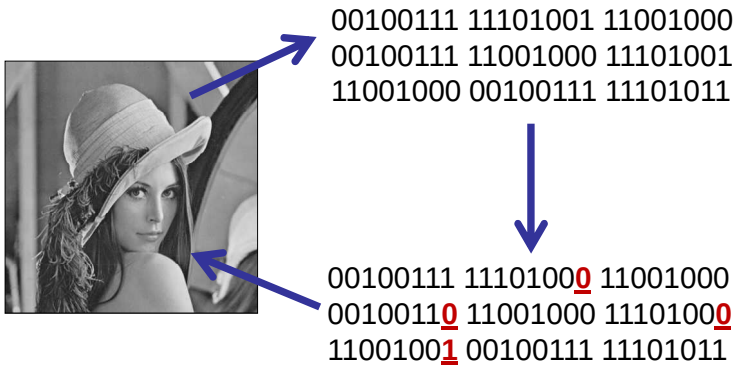


Рисунок 2 – Встраивание информации методом наименьших значащих бит.

Метод встраивания данных в наименьшие значащие биты, или метод LSB (Least Significant Bit), является одним из самых простых в реализации, и одним из самых быстрых стегоалгоритмов. Основная идея, в соответствии с которой осуществляется защита информации, показана на рисунке 2.

Метод наименьших значащих битов является одним из самых ранних методов в стеганографии и используется для встраивания в различные виды носителей. Он заключается в использовании погрешности дискретизации, которая всегда присутствует в изображениях.

В методе наименьших значащих битов не используются модели психовизуального восприятия, или какие-либо оценки вносимой методом в контейнер ошибки, возникающей при встраивании стегосообщения. Для

повышения стойкости, как правило, используется секретный ключ, определяющий множество доступных для встраивания пикселей. На рисунке 2 показан пример защиты изображения методом наименьших значащих битов.

Дополнительно можно почитать следующий материал:

http://habrahabr.ru/post/112976/

http://habrahabr.ru/post/140373/

http://habrahabr.ru/post/114875/

Также на хабре есть несколько статей, "более-менее" следующие:

http://habrahabr.ru/post/230747/

http://habrahabr.ru/post/115287/

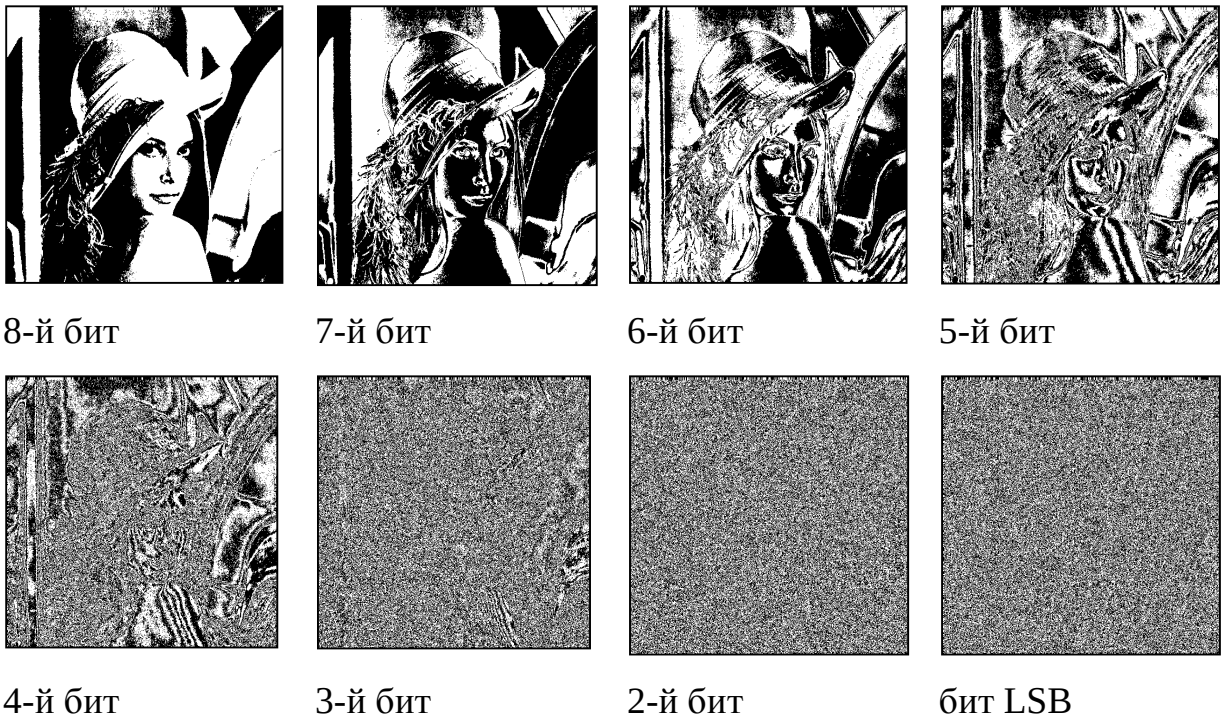


Рисунок 3 – битовые плоскости изображения.

В результате внесения стегоалгоритмом искажений при встраивании, воздействия случайных и преднамеренных атак в канале передачи, а также погрешностей при извлечении, **восстановленное декодером стегосообщение может отличаться от исходного**.

В тоже время, **заполненный стегоконтейнер будет отличаться от исходного**, т.к. обязательно будет искажаться при встраивании скрываемого сообщения.

С одной стороны, используемый стегоалгоритм должен позволять осуществлять достоверную передачу стегосообщений в соответствии с допустимым уровнем вносимых ошибок. Проще говоря, желательно, чтобы стегосообщение можно было выделить из стегоконтейнера, и оно не сильно отличалось от исходного.

С другой стороны, используемый стегоалгоритм должен минимально искажать сам стегоконтейнер, что само по себе невозможно без учета структуры изображения, иначе злоумышленник легко выявит факт встраивания водяного знака.

Следствием строгого соответствия пространственного местоположения исходных битов битам скрываемого сообщения является низкая стойкость к любым видам атак, а также относительно низкая секретность встраивания, т.к. злоумышленнику точно известно местоположение битов скрываемого сообщения в случае, когда секретный ключ не используется.

Одной из серьезных проблем стеганографии заключается в том, что до сих пор не найден адекватный критерий оценки потерь качества изображений при встраивании.

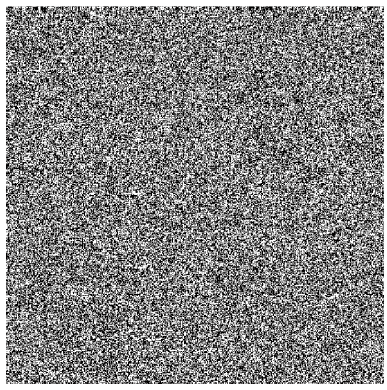
Цифровое изображение – дискретное поле f_{ij} (матрица значений $m \times n$).

$$f_{ij} = f(x_i, y_j) \quad (1)$$

Таким образом, изображение f – исходное изображение, т.е. матрица фиксированных значений $f(x, y)$ взятых в фиксированных точках (x_i, y_j), т.е. $f(x_i, y_j)$ – значение пикселей пустого контейнера, а $g(x_i, y_j)$ – контейнера, содержащего цифровой водяной знак (изображения, полученного после встраивания ЦВЗ).

Пустой стегоконтейнер

Защищенное изображение
PSNR = 51.14 дБ



Внесенные алгоритмом искажения

Рисунок 4 – Искажения, вносимые в контейнер алгоритмом в результате встраивания стегосообщения.

Для оценки изменений можно использовать среднеквадратичное отклонение Mean Square Error:

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \left| f(x_i, y_j) - g(x_i, y_j) \right|^2 \quad (2)$$

Или метрику Root Mean Square (разновидность среднеквадратичного отклонения), которая запишется в следующем виде:

$$RMS = \sqrt{\frac{1}{m \cdot n} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \left| f(x_i, y_j) - g(x_i, y_j) \right|^2} \quad (3)$$

Согласно этой метрике изображение будет сильно испорчено при понижении яркости всего на 5%, но глаз этого не заметит, т.к. у разных мониторов настройка яркости варьируется гораздо сильнее. В то же время изображения с шумом – резким изменением цвета отдельных точек, слабыми полосами или муаром будут признаны почти не изменившимися.

Максимальное отклонение:

$$MD = max \left| f(x_i, y_j) - g(x_i, y_j) \right| \quad (4)$$

Средняя пиксельная ошибка APE или Average Pixel Error:

$$APE = \frac{1}{m \cdot n} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \left| f(x_i, y_j) - g(x_i, y_j) \right| \quad (5)$$

Также часто применяется мера пикового отношения сигнала к шуму (PSNR, Peak Signal-to-Noise Ratio), определяемая как:

$$PSNR = 10 \cdot log_{10} \left[\frac{m \cdot n \cdot N}{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \left| f(x_i, y_j) - g(x_i, y_j) \right|^2} \right] \quad (6)$$

где $N = 255^2$ для полутоновых изображений и $N = 3 \cdot 255^2$для полноцветных.

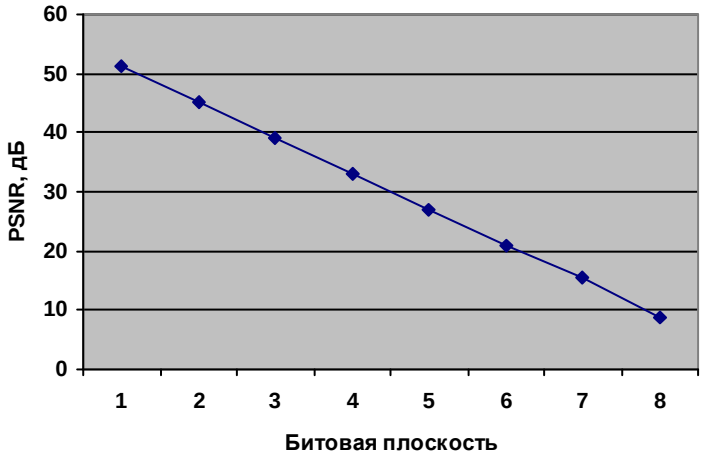


Рисунок 5 – Зависимость отношения уровня сигнала к уровню шума от местоположения встраивания.

Мера PSNR не всегда хорошо согласуется с визуально наблюдаемой ошибкой. Хотя визуально разность между изображениями на рисунке 4 незаметна, PSNR показывает, что изображения отличаются друг от друга.

Метрика BER или Bit Error Rate (коэффициент битовых ошибок, проще говоря – процент искаженных бит) чаще используется для оценки искажений, вносимых в стегосообщение:

$$BER = \frac{N_{error}}{N_{total}} \times 100\%$$

где N_{error} – количество бит ЦВЗ, которые изменились, а N_{total} – общее количество бит ЦВЗ.

На рисунке 5 показана зависимость отношения уровня сигнала к уровню шума от номера битовой плоскости, в которую производится встраивание стегосообщения. Из рисунка видно, что скрытое встраивание возможно лишь в первые три битовые плоскости. В противном случае, если используются старшие биты для встраивания, становятся четко видны изменения, являющиеся искажениями, которые вносит алгоритм в стегоконтейнер.

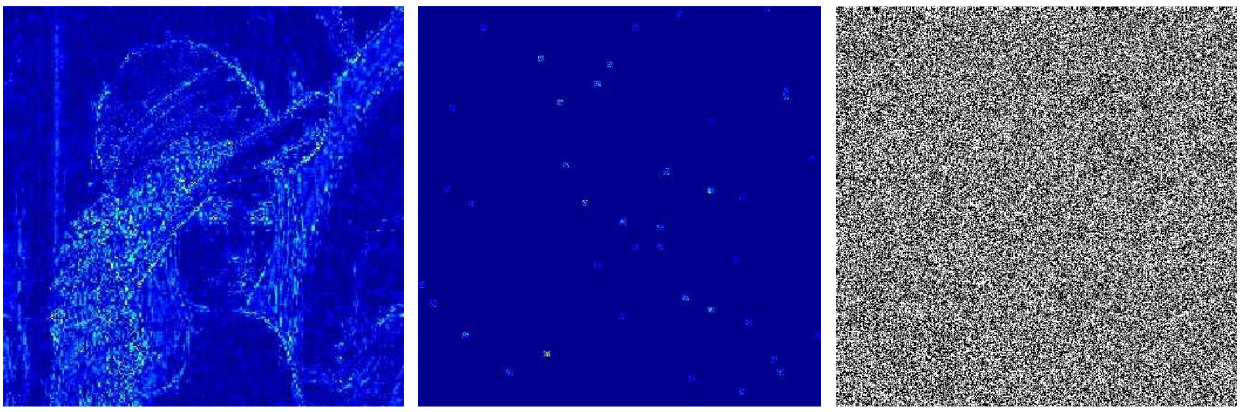


Рисунок 6 – Различные алгоритмы вносят различные изменения в изображения.

Реализация метода LSB, как и других методов, обычно, не составляет труда.

```
for (int i = 0; i < image.Height; i++)
{
for (int j = 0; j < image.Width; j++)
{
        Color pixel = image.GetPixel(j, i);
```

Далее Вы можете воспользоваться операцией остатка от деления на 2 (младший бит LSB либо равен нулю, либо единице):

Например, pixel.R%2 показывает значение младшего бита.

```
R = pixel.R - pixel.R % 2; // обнуляет младший бит в красной составляющей
```

Далее, можно добавлять 1 к pixel.R или не добавлять.

Вы также можете выделить младший (любой) бит с помощью маски. Для младшего бита маска 11111110 = FE.

```
Color pixel = image.GetPixel(j, i);
byte r = pixel.R;

// задаем значение бита
int bit = 0; // по умолчанию встраиваемый бит = 0

bit = ... // записываем в bit очередной бит встраиваемого сообщения

// меняем бит LSB в красной компоненте
r = (byte)((r & 0xFE) | bit); в LSB будет встроен bit
```

Можно скопировать цвет в битовый массив и обращаться к каждому биту.

```
BitArray r = new BitArray(pixel.R);
r[4] = bit; // встраиваем в 4-й бит
r[0] = bit; // встраиваем в LSB

Color newColor = Color.FromArgb(r.GetByte(), G, B);
image.SetPixel(j, i, newColor);
```

Задание

В первой части Вам предлагается выполнить предварительное проектирование и начальную разработку будущего приложения.

Приложение должно открывать, сохранять и выводить:

1. исходное изображение (пустой контейнер);
2. результирующее изображение (изображение со встроенным ЦВЗ);
3. разность исходного и результирующего, как на рисунке 6;
4. исходный ЦВЗ;
5. результирующий ЦВЗ (после извлечения);
6. разность исходного и результирующего ЦВЗ.

Постепенно Вы будете наращивать функционал приложения, добавляя новые методы встраивания и методики их оценки.

Приложение должно реализовывать раздел анализа, минимум 3 метрики: RMS, PSNR и BER.

Приложение должно реализовывать раздел атак, минимум, 1 атаку: атака изменением яркости.

Приложение должно реализовывать минимум 1 метод – LSB, а также дополнительно еще 1 любого пространственного метода из перечисленных ниже для сравнения друг с другом:

1. Метод псевдослучайного интервала
2. Метод псевдослучайной перестановки
3. Метод блочного скрытия
4. Метод квантования
5. Метод замены палитры
6. Метод Куттера-Джордана-Боссена (описан на хабре, см. ссылки)
7. Метод Дамстедтера и др. (ДДКМ)

Основные требования и рекомендации

1. Программирование должно осуществляться на языке высокого уровня, который выбирается студентом на основе собственных предпочтений.
2. Для упрощения задач рекомендуется использование сторонних библиотек, таких как Qt, OpenCV и т.п.