Московский Физико-Технический Институт
(Государственный Университет)
Кафедра Радиотехники
http://re.mipt.ru/infsec

Эссе по курсу «Защита Информации»

Водяные знаки в неподвижных изображениях

Выполнил
Клинчаев О. А.
Студент 111 гр.

г. Долгопрудный 2005

Введение

В наши дни Интернет прочно вошел в нашу жизнь. Новые технологии с каждым днем повышают скорость обмена информацией, растет количество пользователей Глобальной Сети. Этот процесс ставит ряд проблем, связанных с её защитой от несанкционированного доступа, получения защищенного соединения, авторских прав. Данная работа ставит целью провести небольшой обзор приемов защиты авторских прав неподвижных изображений. Наряду с обычным добавлением элементов, обозначающих авторские права, сейчас для этих целей широко применяются технологии, изучаемые **стеганографией**.

Стеганография изучает технологии, позволяющие «скрывать» одни данные в других. Методы стеганографии очень сильно зависят от области применения. Можно выделить следующие основные требования:

1. Необходимо скрыть наличие информации внутри «носителя».
2. Важно затруднить удаление сокрытой информации из носителя.
3. Важно передать большое количество информации.

В приложении к защите авторских прав важную роль играют первые два критерия. Но так как они являются частично взаимоисключающими, то обычно методы копирайта удовлетворяют лишь одному из них.

Общие положения

Рассмотрим общие ограничения, которые влияют на процесс стеганографии.

1. Исходный сигнал («носитель») не должен сильно изменяться после внесения в него данных, связанных с копирайтом. То есть на картинке, к которой была применена та или иная методика стеганографии, не должно быть «артефактов», которые легко затрудняют её восприятие или портят её внешний вид.
2. Данные должны быть вставлены непосредственно в растр изображения, так как данные, вставленные, скажем, в заголовок файла, теряются при конвертировании из одного формата в другой.
3. Вставленная информация не должна быть подвержена влиянию попыток её удалить. Фактически это означает, что она должна быть «труднодоступной» в случае, если криптоаналитику не известен алгоритм или некоторые параметры алгоритма.
4. Данные не должны теряться при воздействии на изображение шума, при аффинных преобразованиях картинки или при сжатии её с ухудшением качества.
5. Должна быть возможность восстановить сокрытые данные по фрагменту изображения.

Таким образом, можно легко сформулировать техническое описание задачи, обозначив важность каждого из этих пунктов.

Видимые водяные знаки.

Наиболее простым методом обозначения авторских прав является непосредственное нанесение на изображение информации о владельце или какой-либо другой, по которой можно будет легко определить автора. В алгоритме, описанном в [3], предлагается добавлять водяные знаки путем наложения серого изображения-идентификатора на исходную

$$I_W(m,n) = \begin{cases} I(m,n) + W(m,n)\left(\frac{I_{white}}{38.667}\right)\left(\frac{I(m,n)}{I_{white}}\right)^{\frac{2}{3}}\alpha_I \\ \qquad \text{for } \frac{I(m,n)}{I_{white}} > 0.008856 \\ I(m,n) + W(m,n)\left(\frac{I(m,n)}{903.3}\right)\alpha_I \\ \qquad \text{for } \frac{I(m,n)}{I_{white}} \leq 0.008856 \end{cases} \quad (1)$$

картинку. При этом упрощенная формула для вычисления интенсивности точек имеет следующий вид. (Здесь I_W – интенсивность серого на «выходе», I – интенсивность серого на исходном изображении, W – интенсивность серого на водяном знаке, I_{white} – интенсивность белого).

При такой обработке исходное изображение примет вид, показанный на рисунке.

Иногда используется обычное нанесение надписи (с затиранием старого содержимого). Такие методы чаще всего применяются на сайтах, для которых копирайт на изображение не является первостепенным фактором.

Сокрытые водяные знаки

Более сложными являются алгоритмы, описывающие процесс нанесения водяных знаков на изображение на основе стеганографии. Несмотря на то, что в этом случае нет возможности визуально определить копирайт, они тоже имеют широкое применение. При таком способе защиты применяется не просмотр внешнего вида картинки, а специальные программы-роботы, которые сканируют сайты и ищут картинки со вставленной информацией.

Методы сокрытия информации можно разделить на 2 подгруппы: параметризуемые и непараметризуемые. У непараметризуемых алгоритмов для извлечения информации достаточно знать сам алгоритм. Параметризуемыми же являются методы, у которых области, куда вставляются данные, зависят от некоторых параметров. Таким образом, зная алгоритм, но, не зная этих параметров, нельзя ни извлечь вставленную информацию, ни обнаружить её наличие.

Patchwork: A statistical approach

Данный подробно метод описан в [1]. Данный метод основывается на предположении, что в среднем яркости двух любых точек равны и имеют гауссовское распределение. А именно: пусть **S = a – b**, где **a** и **b** – яркости двух точек, выбранных случайным образом. Согласно нашему предположению **S = 0**. Но так как у нас точки выбираются случайным образом, то **S** – случайная величина с дисперсией $\sigma_S^2 = \sigma_a^2 + \sigma_b^2$. Таким образом, при одной итерации (по 1-й паре точек) вероятность того, что **|S| > 43** будет больше **0.5**. Но если использовать большое количество пар, то можно получить с достаточно высокой степенью вероятности, что в данное изображение вставлена информация. Общий алгоритм, описанный в [1], состоит из следующих шагов.

1. Создаем псевдослучайную последовательность пар точек ($\mathbf{a_i}$, $\mathbf{b_i}$), зависящую от параметра (seed number).
2. Для всех **i**: у точки $\mathbf{a_i}$ увеличиваем яркость на некоторую величину, а у $\mathbf{b_i}$ – уменьшаем.

Для улучшения стойкости данного алгоритма, например, к аффинным преобразованиям, его можно слегка модифицировать, изменяя яркость не в точках, а в небольших областях.

При использовании данного описания “as is” мы можем проверить только наличие «подписи». В [2] данный алгоритм расширяется таким образом, что можно вставить небольшое количество информации (порядка 64-256 бит для картинки 640x480). Для этого в начале нашей «цепочки» для большого количества элементов мы увеличиваем яркость точки **a** и уменьшаем яркость точки **b**. Таким образом, по ней мы сможем определить, что в изображение действительно вставлена информация об авторских правах. После чего записываются более короткие цепочки. В зависимости от того, нужно нам записать **1** или **0**, мы будем увеличивать яркость для точек **a** соответствующей цепочки или уменьшать. То есть будем считать, что положительное значение S для данного участка цепочки означает, что была записана **1**, а отрицательная – что был записан **0**. Описанный выше алгоритм подходит для любых типов изображений.

Phase watermarking

Данный алгоритм, разобранный в [4], основан на изменении фазы и амплитуды сигнала, получаемого из изображения путем дискретного преобразования Фурье (DFT – Digital Fourier Transform). Формулы DFT выглядят следующим образом:

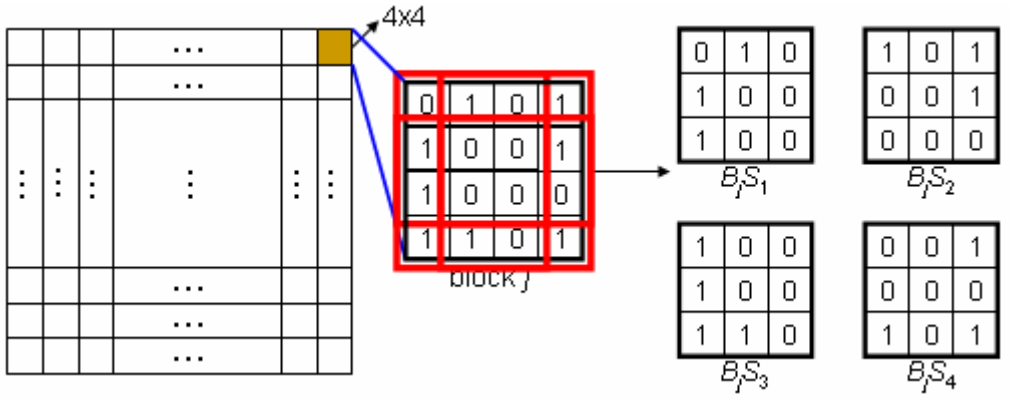
- Прямое
$$F(k_1,k_2)=\beta\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}f(n_1,n_2)e^{-j2\pi n_1k_1/N_1-j2\pi n_2k_2/N_2}$$
- Обратное
$$f(n_1,n_2)=\beta\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}F(k_1,k_2)e^{j2\pi k_1n_1/N_1+j2\pi k_2n_2/N_2}$$

Где коэффициент $\beta=(N_1N_2)^{-1/2}$. определяется из условия, что после последовательного применения прямого и обратного преобразований мы получим исходный сигнал. Такое преобразование генерирует комплексные значения и, следовательно, может быть представлено в виде амплитуды и фазы. Так как мы работаем с картинкой, у которой не может быть комплексных значений яркости точек, то при изменении фазы и амплитуды в точке (k_1, k_2) на **+α** и **+A** мы должны изменить фазу и амплитуду в точке (N_1-k_1, N_2-k_2) на **–α** и **+A** соответственно. Исследования показали, что фаза сильнее влияет на изображение, чем амплитуда, поэтому для увеличения стойкости следует использовать именно её. Для извлечения информации данные дискретного преобразования Фурье полученные из проверяемой картинки сравниваются с оригиналом.

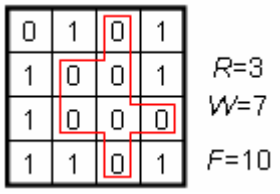
Data hiding in binary image

Приведенные выше алгоритмы не могут применяться к двоичным изображениям. Jeanne Chen , Tung-Shon Chen , Meng-Wen Cheng разработали алгоритм [5], позволяющий вставлять данные в такой тип изображений. Суть метода состоит в том, чтобы, меняя в блоке размером 4x4 один пиксель, добиваться «четности» или «нечетности» блока. Рассмотрим его более подробно. Исходное изображение разбивается на блоки 4x4 пикселя. После чего специальным образом вычисляется характеристическое значение для данного блока. Далее его содержимое анализируется и изменяется специальным образом для записи единицы или нуля. В авторском виде описание алгоритма выглядит следующим образом:

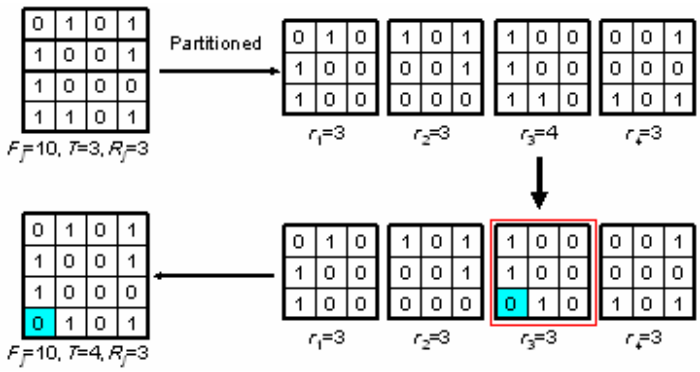
1. Исходное изображение разбивается на блоки 4 на 4, которые разделяются на блоки



размером 3 на 3 как показано на рисунке.
2. Для каждого из блоков 3x3 вычисляется значение **r**, равное количеству единиц в этом блоке, после чего вычисляются параметры **R** и **T**, равные минимальному $\mathbf{r_{min}}$ и количеству блоков 3x3 с таким $\mathbf{r_{min}}$ соответственно.
3. Блоки сортируются по возрастанию **R.**
4. Для каждого блока вычисляются **W**, которое равно количеству соседних точек с одинаковым значением, и **F** = **W** + **R**.



5. Блоки сортируются по возрастанию **F**.
6. Шифрование и дешифрование основывается на формуле $En(D) = \begin{cases} 1, & if\ T = 1,\ 3 \\ 0, & if\ T = 2,\ 4 \end{cases} \quad else$
7. Для шифрования один угловых пикселей соответственного блока 3x3 изменяется таким



образом, чтобы значение **T** блока 4x4 стало четным (нечетным в случае записи единицы).
8. Получение записанной информации проводится аналогично записи: картинка разбивается на блоки, для которых вычисляется значение **T**, после чего на основании этого значения делается вывод, была записана единица или ноль.

Orthogonal patterns

В работе [6] была поставлена задача создания алгоритма, который мог бы использоваться не зависимо от того, знает ли криптоаналитик сам алгоритм. То есть построение алгоритма, работа которого зависит от некоторого параметра – *secret key*. В основе одного из алгоритмов лежат ортогональные функции. Определим произведение двух изображений размера MxN следующим

образом: $<A,B> = \sum_{i=1}^{M}\sum_{j=1}^{N} A_{ij}B_{ij}$. Пусть для данных изобюражений у нас есть полный набор **G** ортогональных функций. В нем будет MxN элементов. Выделим из этого множенства подмножество **F**, состоящее из J элементов. Тогда наше изображение может быть предтавлено как $I = \sum_{i=1}^{J} c_i f_i + g$, где $\mathbf{f_i}$ – элемент из **F**, коэффициенты $\mathbf{c_i}$ могут быть вычислены по формуле $c_i =< f_i, I >$, а **g** – некоторая линейная комбинация элементов **G**, которые не входят в **F**. То есть мы раскладываем наше изображение по выбранной ортогональной системе функций **G** и выбираем часть коэффициентов $\mathbf{c_i}$ для дальнейшей обработки. Выбранные коэффициенты мы заменяем на новые, которые вычисляются по формуле $c_i' = (1+\alpha w_i)c_i$, где **α** – некоторый коэффициент, определяющий стойкость водяных знаков и их «заметность», $\mathbf{w_i}$ – встраиваемые данные. На основе новых коэффициентов создаем изображение с водяными знаками $Iw = \sum_{i=1}^{J} c'_i f_i + g$. Для проверки изображения $Im = \sum_{i=1}^{J} c_i'' f_i + g'$ на наличие в нем водяных знаков следует вычислить для него коэффициент корреляции $corr = \frac{(c''-c)(c'-c)}{\|c''-c\|\,\|c'-c\|}$ и сравнить его с некоторым пороговым значением.

В данном алгоритме основным фактором, определяющим его стойкость, простоту реализации и скорость работы, яляется процесс построения набора **G**. Для увеличения стойкости водяных знаков к таким преобразованиям, как сжатие с потерей качества, следует использовать функции, чья большая часть принадлежит к низким частотам (данный критерий применим ко всем алгоритмам). Здесь удобно использовать построение на основе уже известных ортогональных базисов, модифицируя его некоторым способом, зависящем от секретного ключа. Таким образом, полностью этот алгоритм записи водяных знаков выглядит следующим образом*).

1. На основе *secret key* создаем уникальный набор ортогональных функций, из которого выбираем подмножество функций, которые относятся к низким частотам.
2. Раскладываем изображение по выбранной ортогональной системе.
3. Коэффициенты разложения, относящиеся к функциям из выбранного подмножества изменяем в соответствии с вносимой сигнатурой.
4. «Собираем» картинку, используя измененные коэффициенты.

*) В [6] приведено описание алгоритма через псевдо-инструкции.

Алгоритм проверки наличия водяных знаков:

1. На основе *secret key* создаем уникальный набор ортогональных функций, из которого выбираем подмножество функций, которые относятся к низким частотам.
2. Раскладываем проверяемое, исходное и помеченное (водяными знаками) изображения по выбранной ортогональной системе.
3. Вычисляем коэффициент корреляции и сравнив его с пороговым значением делаем вывод, присутствует ли копирайт в проверяемом изображении.

Пример работы данного алгоритма

Исходное изображение
(Здесь был использован набор из 100 ортогональных функций)

После нанесения водяных знаков

Заключение.

Мы рассмотрели 5 различных алгоритмов нанесения водяных знаков на неподвижное изображение. Данные алгоритмы имеют существенные различия и поэтому выбор алгоритма зависит от поставленной задачи. Для работы с двоичными изображениями подходит только **Data hiding in binary image**; алгоритмы **Phase watermarking**, **Orthogonal patterns** легче всего реализуются для полутоновых (градации серого) изображений (так как для них легче определять скалярное произведение, используемое в преобразованиях). Алгоритм **Patchwork: A statistical approach** по сравнению с другими имеет более простую проверку наличия водяных знаков, не требующую оригинала картинки. Кроме того для него не нужны такие трудоемкие операции, как скалярное произведение, следовательно его удобно применять в ситуации, когда программа-робот выполняет сканирование сайтов для поиска помеченных изображений.

Список литературы

[1] W. Bender, D. Gruhl, N. Morimoto, A. Lu "**Techniques for data hiding**" *IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996*

[2] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. J. Paiz, S. Pogreb "**Applications for data hiding"** *IBM SYSTEMS JOURNAL, VOL 39, NOS 3&4, 2000*

[3] G. W. Braudaway, K. A. Magerlein, and F. Mintzer "**Protecting Publicly Available Images with a Visible Image Watermark**" *Proceedings of the SPIE Conference on Optical Security and Counterfiet Deterrence Technique (Vol. SPIE-2659)*, 1996, pp. 126–132.

[4] J.J.K. O Ruanaidh W.J. Dowling and F.M. Boland "**PHASE WATERMARKING OF DIGITAL IMAGES**" *Department of Electronic and Electrical Engineering University of Dublin Trinity College Dublin Ireland*

[5] Jeanne Chen , Tung-Shon Chen , Meng-Wen Cheng "**A New Data Hiding Method in Binary Image**" *Multimedia Software Engineering ,Proceedings. Fifth International Symposium on , 2003 , Pages : 88 - 93*

[6] Jiri Fridrich, Lt Arnold C. Baldoza and Richard J. Simard "**Robust digital watermarking based on keydependent basis functions**" *Center for Intelligent Systems SUNY Binghamton Binghamton, NY 13902-6000, Air Force Research Laboratory/IFEC 32 Hangar Road Rome, NY 13441-4114*