DOI: 10.15593/2499-9873/2019.4.04
УДК 004.056.5

Е.Л. Кротова, А.В. Чекменев, А.О. Болгов

Пермский национальный исследовательский политехнический университет,
Пермь, Россия

МЕТОД ВЫДЕЛЕНИЯ СТЕГАНОГРАФИЧЕСКИХ ВОДЯНЫХ ЗНАКОВ ПО КЛЮЧУ С ИСПОЛЬЗОВАНИЕМ ВЕЙВЛЕТОВ ХААРА

Рассмотрен способ нанесения закодированного цифрового стеганографического водяного знака на цифровое изображение и последующее его выделение при помощи вейвлетов Хаара. Освещен метод нанесения цифрового стеганографического водяного знака по ключу и выделение этого знака с помощью ключа. Рассмотрена актуальность данного способа нанесения и выделения цифрового стеганографического водяного знака. В нескольких словах описан метод разбиения сигнала на подсигналы при помощи алгоритма Хаара и то, как он применим в контексте цифровых изображений. Представлены результаты проверки нанесения цифрового водяного знака на устойчивость к различным преобразованиям, таким как размытие с ядром 3×3, 5×5, jpeg-сжатие с коэффициентом сжатия 50 и 70 %, удаление одного младшего, двух и четырех младших бит. Представлены соответствующие изображения, которые иллюстрируют результаты проверок на устойчивость к преобразованиям цифрового стеганографического водяного знака. Приведен иллюстративный и простой в реализации пример нанесения цифрового стеганографического водяного знака, а также его извлечение по заранее созданному ключу с использованием простого кодирования, которое заключается в том, что столбцы пикселей исходного изображения сдвигаются на определенное число позиций. Также в статье представлено краткое описание LSB-алгоритма и рассмотрены основные преимущества и недостатки алгоритма, разработанного и представленного в этой статье, со стандартным LSB-алгоритмом. В заключение сделаны соответствующие выводы о применимости разработанного алгоритма, о его недостатках и достоинствах.

Ключевые слова: преобразование Хаара, стеганографические водяные знаки, цифровые водяные знаки, стеганографические алгоритмы, кодирование, интернет.

E.L. Krotova, A.V. Chekmenev, A.O. Bolgov

Perm National Research Polytechnic University, Perm, Russian Federation

THE METHOD OF EXTRACTING THE STEGANOGRAPHY WATERMARKS BY KEY USING HAAR WAVELETS

This article describes a method for applying an encoded digital steganographic watermark to a digital image and its subsequent extraction using Haar wavelets. The method of applying a digital steganographic watermark by key, and highlighting this sign with a key, is considered. The relevance of this method of applying and highlighting a digital steganographic watermark is considered. A few words describe the method of splitting a signal into sub-signals using the Haar algorithm and how it is applicable in the context of digital images. The results of checking the application of a digital watermark for resistance to various transformations are presented, such as: blurring with a 3×3, 5×5 core, jpeg com-

pression with a compression ratio of 50 and 70 %, deleting the 1 LSB, 2 and 4 LSBs. Corresponding images are presented that illustrate the results of tests for resistance to conversion of a digital steganographic watermark. A small, illustrative and easy to implement example of applying a digital steganographic watermark, as well as its extraction using a previously created key using simple coding, which consists in the fact that the columns of pixels of the original image are shifted by a certain number of positions, is presented. Also, the article provides a brief description of the LSB algorithm and considers the main advantages and disadvantages of the algorithm developed and presented in this article with the standard LSB algorithm. In conclusion, the corresponding conclusions were drawn about the applicability of the developed algorithm, about its shortcomings and advantages, described in this article.

Keywords: Haar transform, steganography watermarks, digital watermarks, steganography algorithms, coding, Internet.

Введение

В современном мире с его огромными, тяжело поддающимися контролю информационными потоками проблема отслеживания и тем более доказательства своего авторского права на цифровые информационные ресурсы встает особо остро. Зачастую для этого используют СВЗ (стеганографические водяные знаки) [1–5], в данной статье сделана попытка осуществить один из таких алгоритмов СВЗ, а именно рассмотрен способ нанесения и последующего выделения стеганографического знака при помощи преобразований Хаара [6, 7], а также кодирование и декодирование знака по специально сгенерированному ключу [8] и различные проверки на попытки стереть СВЗ с исходного изображения.

1.1. Метод Хаара

Метод Хаара для одномерного сигнала заключается в разбиении значений сигнала на полусумму и полуразность [1].

$$x_1 = \frac{\text{value}_i + \text{value}_{i+1}}{2},$$

$$x_2 = \frac{\text{value}_i - \text{value}_{i+1}}{2}.$$

В контексте изображений эти значения представляют собой яркости пикселей в grayscale-формате. Полусумма представляет собой среднее значение яркости двух пикселей, а полуразность – их отличие.

Чтобы посчитать новый вектор полусумм и полуразностей, следует просто умножить исходный вектор пикселей на матрицу преобразования:

– a – первый пиксель,

– b – второй пиксель.

$$\begin{pmatrix} 1/2 & 1/2 \\ -1/2 & 1/2 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \dfrac{a+b}{2} \\ \dfrac{a-b}{2} \end{pmatrix}.$$

Для восстановления необходимо умножить полусумму и полуразность на обратную матрицу преобразования:

$$\begin{pmatrix} \dfrac{a+b}{2} \\ \dfrac{a-b}{2} \end{pmatrix}\begin{pmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

1.2. Нанесение и выделение стеганографического водяного знака

Для начала необходимо проверить, насколько возможен сам факт использования такого СВЗ (без кодирования), для этого ниже приведен алгоритм нанесения и последующего выделения такого знака. Для примера попробуем нанести изображение буквы W. В данной работе мы называем изображение, которое наносится внутрь другого изображения, стегосообщением, а изображение, внутрь которого встраивается стегосообщение, – контейнером.

Нанесение стеганографического водяного знака происходит следующим образом:

1. Считывается исходное изображение или кадр из видео, на которое необходимо нанести водяной знак.

2. Создается отдельная grayscale-копия исходного изображения.

3. На grayscale-изображении ищется область с похожей яркостью пикселей (в данном случае разность между максимальной и минимальной яркостью не больше 5).

4. Далее данная область на цветном изображении подвергается небольшому размытию.

5. К найденной области прикладывается маска водяного знака. Пример исходного изображения с нанесенным водяным знаком (рис. 1). Как можно заметить, водяного знака на картинке практически не видно.

6. Далее, при необходимости выделения знака, картинка с нанесенным водяным знаком подвергается прямому преобразования Хаара.

В результате изображение будет разбито на две части, первая часть – приближенная версия исходного изображения, вторая – информация, которая содержит в себе детали для восстановления исходного изображения (далее – шумы).

Рис. 1. Изображение с нанесенным СВЗ

Выделим часть фотографии с шумами для лучшей видимости водяного знака (рис. 2):

Рис. 2. Изображение с СВЗ, подвергнутое преобразованию Хаара

1.3. Кодирование по ключу

Алгоритм выделения марки базируется на простом преобразовании Хаара, поэтому скрыть информацию в контейнере или доказать, что данное видео является вашей собственностью, при помощи простого нанесения знака довольно затруднительно. А значит, необходимо добавить метод, который позволил бы однозначно доказать принадлежность объекта авторского права автору. Самым очевидным и, скорее всего, простым методом является кодирование/декодирование по ключу.

Действительно, при попытке выделить нанесенный СВЗ, закодированный при этом специальным ключом, мы получим лишь непонятный шум вместо водяного знака (рис. 3).

Рис 3. Изображение с закодированной маркой,
подвергнутое преобразованию Хаара

На изображении выше представлена попытка выделения водяного знака без предварительного декодирования по специальному ключу.

Для эксперимента в качестве ключа была использована простая строка символов, длина которой равна размеру СВЗ на изображении. Каждый *i*-й символ в данной строке декодируется по таблице ACII в число, которое представляет собой количество пикселей, на которое нужно сдвинуть *i*-й столбец стегосообщения (если количество пикселей, на которое нужно сдвинуть столбец, превышает количество пикселей столбца, то следует просто смещать пиксели по кругу, т.е. последний пиксель столбца становится первым и т.д.). После символов в строке выделено 3 байта под кодирование размера водяного знака.

Алгоритм сдвига столбцов для кодирования марки:

1. Копируем значения столбца изображения

$$\text{OneVectorVaue}[i] = \text{WatermarkMatrix}[j][i],$$

где $i = 0 \div \text{WatermarkCols}$, $j = 0 \div \text{WatermarkRows}$.

2. Присваиваем эти значения обратно в матрицу, учитывая сдвиг и перенос:

$$\text{WatermarkMatrix}[(j + \text{Key}[i]) \,\%\, \text{WatermarkRows}][i] = \text{OneVectorValue}[i],$$

где Key[*i*] – *i*-й элемент строки, содержащей ключ.

Пусть имеется водяной знак, заданный матрицей, показанной ниже, элемент матрицы – координата *i*-го пикселя:

1	2	3
4	5	6
7	8	9

Для данной матрицы будет использован ключ длиной 3 (так как высота и ширина матрицы равна трем). Ключ: Dos.

Обратимся к таблице ASCII для того, чтобы представить сообщение соответствующими числами из этой таблицы:

– D = 104,

– o = 157,

– s = 163.

Данные десятичные значения указывают, на сколько элементов нужно сдвинуть каждый столбец: 1 – на 104, 2 – на 117, 3 – на 123.

Для того чтобы не выполнять сдвиг вручную, можно воспользоваться формулой, приведенной выше, и узнать координаты нового места после сдвига (2-й пункт формулы сдвига столбцов).

Матрица после сдвига:

4	8	9
7	2	3
1	5	6

Данный алгоритм кодирования неэффективен, но вполне подойдет для описания процесса нанесения по ключу и заключения некоторых выводов, которые мы приведем позже.

Алгоритм нанесения закодированного вотермарка почти ничем не отличается от алгоритма нанесения незакодированного, добавляется лишь шаг (между 3-м и 4-м шагом исходного алгоритма) с кодированием изображения марки и последующим наложением закодированной марки на исходное изображение.

1.4. Декодирование марки и ее выделение

Допустим, мы получили изображение с нанесенной закодированной маркой. Для получения изображения водяного знака необходимо применить обратный сдвиг пикселей, но так как мы заранее не знаем, где расположен водяной знак, а знаем его размер, мы просто выделяем шумы с изображения, а затем делим изображение с шумами на облас-

ти, размер которых равен размеру изображения водяного знака, и применяем к каждой такой области декодирование по ключу (выполняем обратный сдвиг пикселей). В результате должен получиться устойчивый образ водяного знака (рис. 4).

Рис. 4. Изображение с декодированной маркой(*png)

Алгоритм сдвига столбцов для декодирования марки:

1. Копируем значения столбца изображения:

$$\text{OneVectorVaue}[i] = \text{WatermarkMatrix}[j][i],$$

где $i = 0 \div \text{WatermarkCols}$, $j = 0 \div \text{WatermarkRows}$.

2. Считаем обратный сдвиг:

$$\text{Shift} = \text{WatermarkRows} - (\text{Key}[i] \ \% \ \text{WatermarkRows}),$$

где Key[i] – i-й элемент строки, содержащей ключ.

3. Присваиваем эти значения обратно в матрицу, учитывая сдвиг и перенос (см. рис. 4):

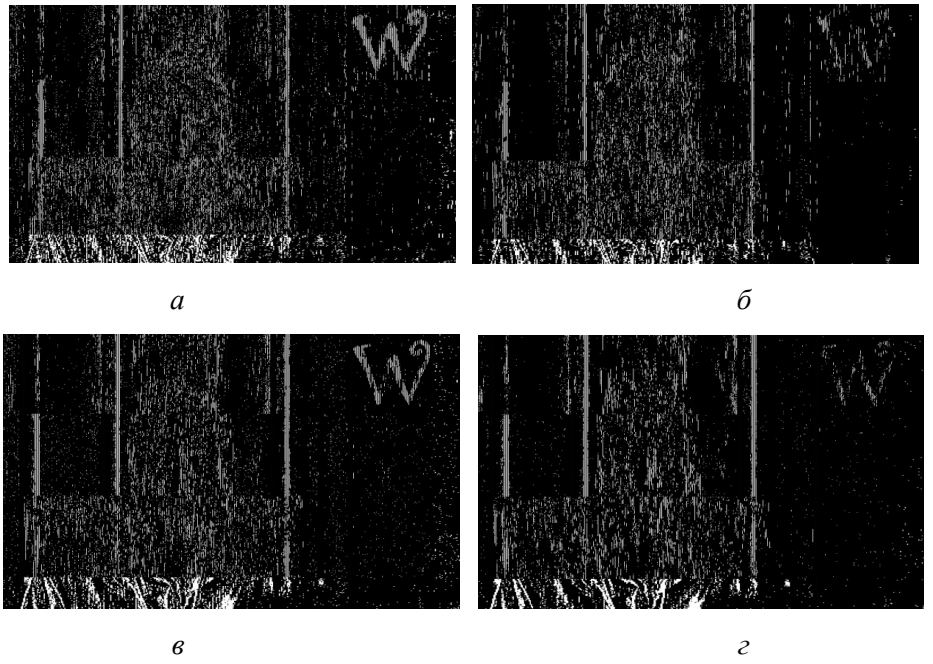
$$\text{WatermarkMatrix}[(j + \text{Shift}) \ \% \ \text{WatermarkRows}][i] = \text{OneVectorValue}[i],$$

1.5. Проверка на устойчивость

Все попытки нанесения СВЗ на контейнер для защиты авторского права не имеют смысла, если этот СВЗ можно будет без особых усилий стереть, а сам контейнер при этом никак не изменится (по изображению не будет понятно, подвергалось ли оно преобразованиям, которые характерны для попыток удалить стегосообщение).

Проверим данный алгоритм на устойчивость к некоторым видам преобразования:

1. Jpeg-сжатие 50 % (рис. 5).
2. Jpeg-сжатие 70 % (см. рис. 5, *б*).
3. Размытие с ядром (3×3) (см. рис. 5, *в*).
4. Размытие с ядром (5×5) (см. рис. 5, *г*).
5. Удаление 1 младшего бита в каждом байте изображения (рис. 6).
6. Удаление 2 младших бит в каждом байте изображения (см. рис. 6).
7. Удаление 4 младших бит в каждом байте изображения (см. рис. 6).



а *б*

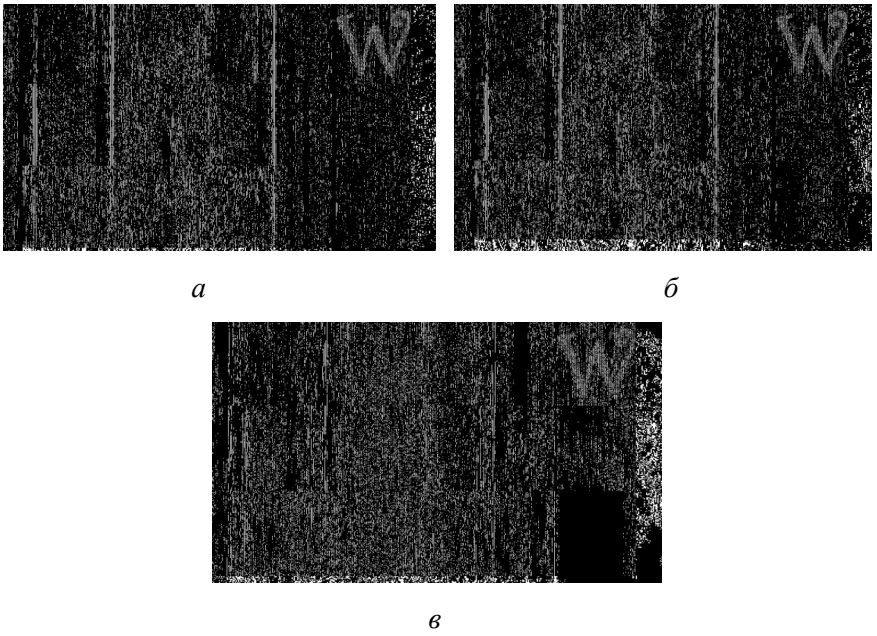
в *г*

Рис. 5. Устойчивость стегосообщения к преобразованиям:
а – jpeg-сжатие 50 %; *б* – jpeg-сжатие 70 %; *в* – размытие
с ядром (3×3); *г* – размытие с ядром (5×5)

Как можно заметить (рис. 6, 7), СВЗ устойчив к большинству видов преобразований, попытки удалять его дальше могу привести к сильной потере качества изображения.

Далее следует упомянуть в выборе алгоритма кодирования водяного знака то, что так как все попытки стереть стегосообщения сводятся к различным манипуляциям с контейнером, то следует брать во внимание возможные варианты этих манипуляций, будь то jpeg-

сжатие, размытие и т.д. Например, если предположить, что, скорее всего, контейнер будут пытаться «размыть», то алгоритм кодирования СВЗ следует подобрать такой, чтобы нанесенные впоследствии пиксели не находились рядом друг с другом.



а *б*

в

Рис. 6. Устойчивость стегосообщения к удалению битов:
а – удаление 1 младшего бита в каждом байте изображения;
б – удаление 2 младших бит в каждом байте изображения;
в – удаление 4 младших бит в каждом байте изображения

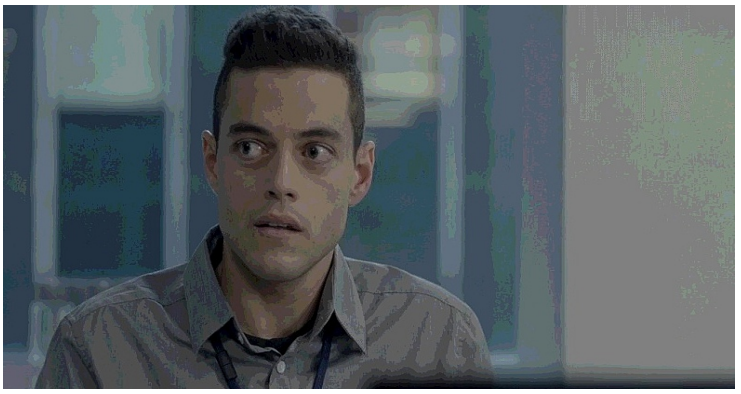


Рис. 7. Пример изображения с 4 удаленными младшими байтами

1.6. Сравнение с LSB-алгоритмом

Коротко скажем о LSB-алгоритме (Last Significant Bit) [9, 10]. Данный алгоритм позволяет записывать в каждый младший бит контейнера произвольную информацию, в том числе другое изображение.

Если сравнивать алгоритм, описанный выше, с LSB-алгоритмом, то можно сказать, что первый намного более устойчив к попыткам стереть стегосообщение, но в то же время LSB-алгоритм прячет стегосообщение в чистом виде, т.е. никак его не изменяя, в то время как описанный нами алгоритм наносит стегосообщение на контейнер, увеличивая разницу между соседними пикселями в контейнере, что подразумевает, что от стегосообщения остается черно-белый силуэт.

Но по очевидным причинам для защиты авторского права более значимым и важным является именно устойчивость: как можно убедиться, стегосообщение несмотря на преобразования хоть и теряет в качестве, но по-прежнему остается узнаваемым, в то время как при использовании LSB-алгоритма вся информация теряется даже при малейших преобразованиях.

Вывод

В данной статье был рассмотрен способ нанесения устойчивого СВЗ, а также выделение его при помощи преобразований Хаара, также был рассмотрен способ кодирования марки и выделения ее по ключу. Для примера был взят простой алгоритм кодирования, но в дальнейшем можно использовать другие, более подходящие алгоритмы. В статье были представлены результаты попыток различных преобразований с целью удаления СВЗ, также были сделаны выводы об устойчивости. Было приведено сравнение с LSB-алгоритмом. В результате можно заключить, что способ нанесения СВЗ, представленный в данной статье, обладает необходимым минимумом характеристик для использования его в целях защиты авторского права. Данный способ обладает приемлемой устойчивостью, практически незаметен на исходном изображении и весьма прост в реализации.

Список литературы

1. Стеганография в XXI веке. Цели. Практическое применение. Актуальность [Электронный ресурс]. – URL: https://habr.com/en/post/253045/ (дата обращения: 20.06.2019).

2. Стеганография [Электронный ресурс]. – URL: https://photodb.illusdolphin.net/media/4781/stego.pdf (дата обращения: 20.06.2019).

3. Steganography and digital watermarking. – URL: https://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.pdf (accessed 20.06.2019).

4. Watermark. – URL: https://en.wikipedia.org/ wiki/Watermark (accessed 20.06.2019).

5. Digital steganography: hidden data within data. – URL: https://ieeexplore.ieee.org/document/935180/keywords#keywords (accessed 20.06.2019).

6. Haar wavelet. – URL: https://en.wikipedia.org/wiki/Haar_wavelet (accessed 20.06.2019).

7. Вейвлет-сжатие «На пальцах» [Электронный ресурс]. – URL: https://habr.com/en/post/168517/ (дата обращения: 20.06.2019).

8. Kernel image processing. – URL: https://en.wikipedia.org/wiki/Kernel_(image_processing) (accessed 20.06.2019).

9. Least significant bits insertion. – URL: http://www.lia.deis.unibo.it/Courses/RetiDiCalcolatori/Progetti98/Fortini/lsb.html (accessed 20.06.2019).

10. LSB-стеганография. – URL: https://habr.com/ ru/post/112976/ (accessed 20.06.2019).

References

1. Steganographiya v XXI veke. Celi. Prakticheskoye priminenie. Aktyalnost, available at: https://habr.com/en/post/253045/ (accessed 20 June 2019).

2. Steganographiya, available at: https://photodb.illusdolphin.net/media/4781/stego.pdf (accessed 20 June 2019).

3. Steganography and digital watermarkin, available at: https://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.pdf (accessed 20 June 2019).

4. Watermark, available at: https://en.wikipedia.org/wiki/Watermark (accessed 20 June 2019).

5. Digital steganography: hidden data within data, available at: https://ieeexplore.ieee.org/document/935180/keywords#keywords (accessed 20 June 2019).

6. Haar wavelet, available at: https://en.wikipedia.org/wiki/ Haar_wavelet (accessed 20 June 2019).

7. Veivlet-szhatie “Na paltsah”, available at: https://habr.com/en/post/168517/ дата обращения (accessed 20 June 2019).

8. Kernel image processing, available at: https://en.wikipedia. org/wiki/Kernel_(image_processing) (accessed 20 June 2019).

9. Least significant bits insertion, available at: http://www.lia.deis.unibo.it/Courses/RetiDiCalcolatori/Progetti98/Fortini/lsb.html (accessed 20 June 2019).

10. LSB steganographiya, available at: https://habr.com/ru/post/112976/ (accessed 20 June 2019).

Получено 20.06.2019

Сведения об авторах

Кротова Елена Львовна (Пермь, Россия) – кандидат физико-математических наук, доцент кафедры «Высшая математика», Пермский национальный исследовательский политехнический университет (614990, Пермь, Комсомольский пр., 29, e-mail: lenkakrotova@yandex.ru).

Чекменев Андрей Владимирович (Пермь, Россия) – студент, электротехнический факультет, Пермский национальный исследовательский политехнический университет (614990, Пермь, Комсомольский пр., 29, e-mail: spiritofice@yandex.ru).

Болгов Александр Олегович (Пермь, Россия) – студент, электротехнический факультет, Пермский национальный исследовательский политехнический университет (614990, Пермь, Комсомольский пр., 29, e-mail: aleksynderbolgov@gmail.com).

About the authors

Elena L. Krotova (Perm, Russian Federation) – Ph.D. in Physics and Mathematics, Associate Professor, Department of the Higher Mathematics, Perm National Research Polytechnic University (614990, Perm, Komsomolsky av., 29, Russian Federation, e-mail: lenkakrotova@yandex.ru).

Andrey V. Chekmenev (Perm, Russian Federation) – Student, Electrical engineering faculty, Perm National Research Polytechnic University (614990, Perm, Komsomolsky av., 29, Russian Federation, e-mail: spiritofice@yandex.ru).

Aleksandr O. Bolgov (Perm, Russian Federation) – Student, Electrical engineering faculty, Perm National Research Polytechnic University (614990, Perm, Komsomolsky av., 29, Russian Federation, e-mail: aleksynderbolgov@gmail.com).