УДК 621.391.7

Сдвиговый метод обнаружения скрытой информации

В. А. Монарев

В работе описан новый метод обнаружения скрытой информации. Данный метод может быть применён для поиска скрытой методом LSB-замещения информации. Приведены экспериментальные данные, позволяющие сравнить эффективность метода с лучшими известными методами стегоанализа.

Ключевые слова: стегонаграфия, стегоанализ.

1. Введение

Стеганография – это раздел защиты информации, который занимается скрытием факта передачи информации. Как правило, информация скрывается в объекты с «безобидным» содержанием: звуковые файлы, изображения и видео. Файл, в котором скрыта информация, называют контейнером. Количество информации, которое можно внедрить в контейнер, называют ёмкостью контейнера.

Методы стегоанализа – это методы, которые позволяют определить наличие скрытой информации в контейнере. Основная сложность заключена в том, что информации в контейнер может быть внедрено очень мало (например, 5% от ёмкости), и это сильно усложняет задачу обнаружения.

Данная работа посвящена стегоанализу, который основан на методах сжатия. Впервые методы стегоанализа, основанные на методах сжатия, были предложены в [1]. Они позволяют эффективно определять внедрение при 35%-ом заполнении от ёмкости контейнера.

В данной работе рассмотрен стегоанализ файлов с изображением в неискажающем формате (bmp, png, tiff и другие). Информация внедрялась в последние значащие биты (LSB). В литературе такой способ внедрения называют LSB-замещением [2]. Наиболее эффективные и известные методы обнаружения LSB-замещения описаны в [3 – 5]. Экспериментально показано, что при определённых условиях новый метод существенно лучше обнаруживает скрытую информацию. Тестирование было проведено на 18 тысячах цветных фотографий.

2. Сжатие и информация

2.1. Основная идея

Методы сжатия информации не так давно стали применяться для поиска скрытой информации [1]. Основная идея, на которой основаны эти методы, заключается в следующем. Если взять пустое (не содержащее скрытой информации) изображение I и взять это же изображение, но с внедрёнными h%-ми скрытой информации методом LSB (обозначим это изображение через I_h), то можно наблюдать следующую закономерность: чем больше информа-

ции внедрено в файл (чем больше h), тем больше размер файла $\varphi(I_h)$, где φ – это метод сжатия информации. Поясним, что здесь и далее под словами «внедрено h%» подразумевается, что в изображение внедрено h% информации от общей ёмкости контейнера. Отметим также, что в качестве метода сжатия можно использовать любой архиватор (WinRar, WinZip и другие).

Основная проблема этого метода в том, что изначально неизвестно, как хорошо сжимается файл. Поэтому такой метод эффективно определяет внедрение в случае, если внедрено больше, чем 35% (см. [1]). Например, методы описанные [2 – 5], способны обнаруживать 3%-ое внедрение.

2.2. Описание нового метода

Рассмотрим 24-битное изображение I в формате bmp. В данном формате каждому пикселю соответствует три байта (R, G, B) , где R – это компонента, соответствующая красному цвету, G – зеленому и B – голубому. При использовании LSB-внедрения в младшие биты компонент цвета внедряется некоторая информация. То есть, если дано изображение размера 800×600, то в него можно внедрить 3×800×600 бит информации. Если объём информации, которую надо внедрить, меньше чем 3×800×600, то эта информация случайным образом размещается внутри файла (в младших битах). Почему внедрение производится только в младшие биты? Ответ очень прост. Цвет точек в изображении изменяется на минимальное значение, поэтому на глаз практически невозможно определить внедрённую информацию.

Обозначим через shift(I) изображение, полученное из исходного изображения, сдвигом всех компонент на 1 бит вправо. То есть, вместо (R, G, B) в новом файле будут стоять байты ($R>>1$, $G>>1$, $B>>1$), где >> – стандартная операция нециклического сдвига вправо. Также обозначим через I_{+1} и I_{-1} изображения, полученные из исходного, прибавлением единицы (и вычитанием соответственно) из каждой компоненты. Ясно, что I_{+1} будет немногим отличаться от изображения I. Это будет немного более яркое изображение. Легко проверить, что значения $|\varphi(I_{-1})| \approx |\varphi(I_{+1})| \approx |\varphi(I)|$, где через $|\varphi(I)|$ обозначим длину файла $\varphi(I)$, который соответствует изображению I. Естественным будет предположить следующее: если исходная компонента R была равна максимальному значению (максимальное значение байта 255), то после операции +1 полагаем, что она не изменяется. Аналогично с операцией −1 и нулевыми байтами.

Главная идея заключается в следующем. Если рассмотреть изображение I со скрытой информацией, то изображение shift(I) не зависит от внедрённой информации, поскольку она при сдвиге удалится. С другой стороны, shift(I_{+1}) зависит от внедрённой информации, поскольку после операции +1 от внедрённой информации стал зависеть не только первый бит.

Поясним идею на примере. Предположим, что все компоненты двух последовательных пикселей равны 100. То есть в файле записаны шесть подряд идущих байт: 100, 100, 100, 100, 100, 100. После внедрения информации в младшие биты у нас, допустим, получилась последовательность: 101, 100, 100, 100, 101, 101. В изображениях shift(I_{+1}) и shift(I) этим пикселям будут соответствовать байты: 51, 50, 50, 50, 51, 51 (для shift(I_{+1})) и 50, 50, 50, 50, 50, 50 (для shift(I)). Видим, что внедрённая информация повлияла на изображение shift(I_{+1}). Аналогичный вывод можно сделать относительно изображения shift(I_{-1}). Таким образом, если информация была внедрена, то

$$|\varphi(\text{shift}(I_{+1}))| > |\varphi(\text{shift}(I))| \quad . \tag{1}$$

Если нет, то

$$\left|\varphi(\text{shift}(I_{+1}))\right| \approx \left|\varphi(\text{shift}(I))\right| \quad . \tag{2}$$

Ясно, что даже в случае, когда I не содержит скрытой информации, условие $\left|\varphi(\text{shift}(I_{+1}))\right| > \left|\varphi(\text{shift}(I))\right|$ будет часто выполняться, поскольку выражение (2) подразумевает как знак больше, так и меньше. То есть важно, насколько левая часть в неравенстве (1) больше, чем правая.

Заметим, что у нас есть «эталонное» изображение, которое не содержит скрытой информации – это изображение shift(I). Внедряя в него информацию и исследуя размер архива, можно оценить, сколько скрытой информации содержится в $\text{shift}(I_{+1})$ и $\text{shift}(I_{-1})$, а следовательно, и в I. Поясним алгоритм на примере одного из изображений. Находим длины архивов:

$\left|\varphi(\text{shift}(I))\right|$ = 795427 байт, $\frac{1}{2}\left(\left|\varphi(\text{shift}(I_{+1}))\right| + \left|\varphi(\text{shift}(I_{-1}))\right|\right)$ = 796044 байт. Далее, внедряем в «эталонное» изображение shift(I) (обозначим его через I^*) 0.5%, 1%, 2%, 5% и вычисляем значения: $\left|\varphi(I^*_{0.5})\right|$ = 797617 байт, $\left|\varphi(I^*_1)\right|$ = 799843 байт, $\left|\varphi(I^*_2)\right|$ = 804035 байт, $\left|\varphi(I^*_5)\right|$ = 815430 байт. Откуда легко сделать вывод, что изображение содержит меньше, чем 0.5% скрытой информации. Если же мы возьмём это же изображение, внедрим в него 1.5% и попытаемся обнаружить эту скрытую информацию, то получим:

$\left|\varphi(\text{shift}(I))\right|$ = 795427 байт; $\frac{1}{2}\left(\left|\varphi(\text{shift}(I_{+1}))\right| + \left|\varphi(\text{shift}(I_{-1}))\right|\right)$ = 801722 байт,

$$\left|\varphi(I^*_1)\right| \le \frac{1}{2}\left(\left|\varphi(\text{shift}(I_{+1}))\right| + \left|\varphi(\text{shift}(I_{-1}))\right|\right) \le \left|\varphi(I^*_2)\right|.$$

Откуда можно сделать вывод, что содержится больше 1% скрытой информации, но меньше, чем 2%. В программной реализации метода использовалась линейная интерполяция для более точной оценки количества внедрённой информации. В качестве метода сжатия был использован формат png, позволяющий хранить изображение в сжатом и неискаженном виде.

3. Результаты экспериментов

3.1. Описание базы изображений

Для экспериментов использовалась общедоступная база изображений, которая использовалась в ходе конкурса BOSS («Break Our Steganographic System»). База содержит 9000 изображений в RAW-формате. Поскольку давно известна зависимость эффективности обнаружения скрытой информации от разрешения изображения, то для более детального исследования эффективности стегоанализа исходная база данных была преобразована тремя способами. В первом случае из исходного изображения в RAW-формате сначала было вырезано квадратное изображение максимально возможного размера (если разрешение исходного изображения было, например, 5000×4000, то вырезаем 4000×4000). Далее, это изображение было уменьшено до размера 1800×1800 и преобразовано в формат bmp. После всего из полученного изображения было вырезано изображение 512×512. Таким образом, было получено первое множество изображений. Далее, аналогично были получены изображения двух других множеств; единственное отличие в том, что уменьшение на втором шаге преобразований

происходило не до 1800×1800, а до 1200×1200 и 800×800 для второго и третьего множества соответственно.

В качестве методов сжатия было использовано 4 архиватора (один из них исследовался в двух режимах). В таблице приведены названия архиваторов и параметры их запуска. Последние два архиватора были взяты с сайта `http://compressionratings.com`, судя по данным этого сайта, это два лучших архиватора для сжатия изображений.

Номер архиватора	1	2	3	4	5
Название	PNG	RAR	RAR	BSC	NANOZIP
Параметры	-	-m2	-m5	-m3 -b26	-m.5g -cDP -y

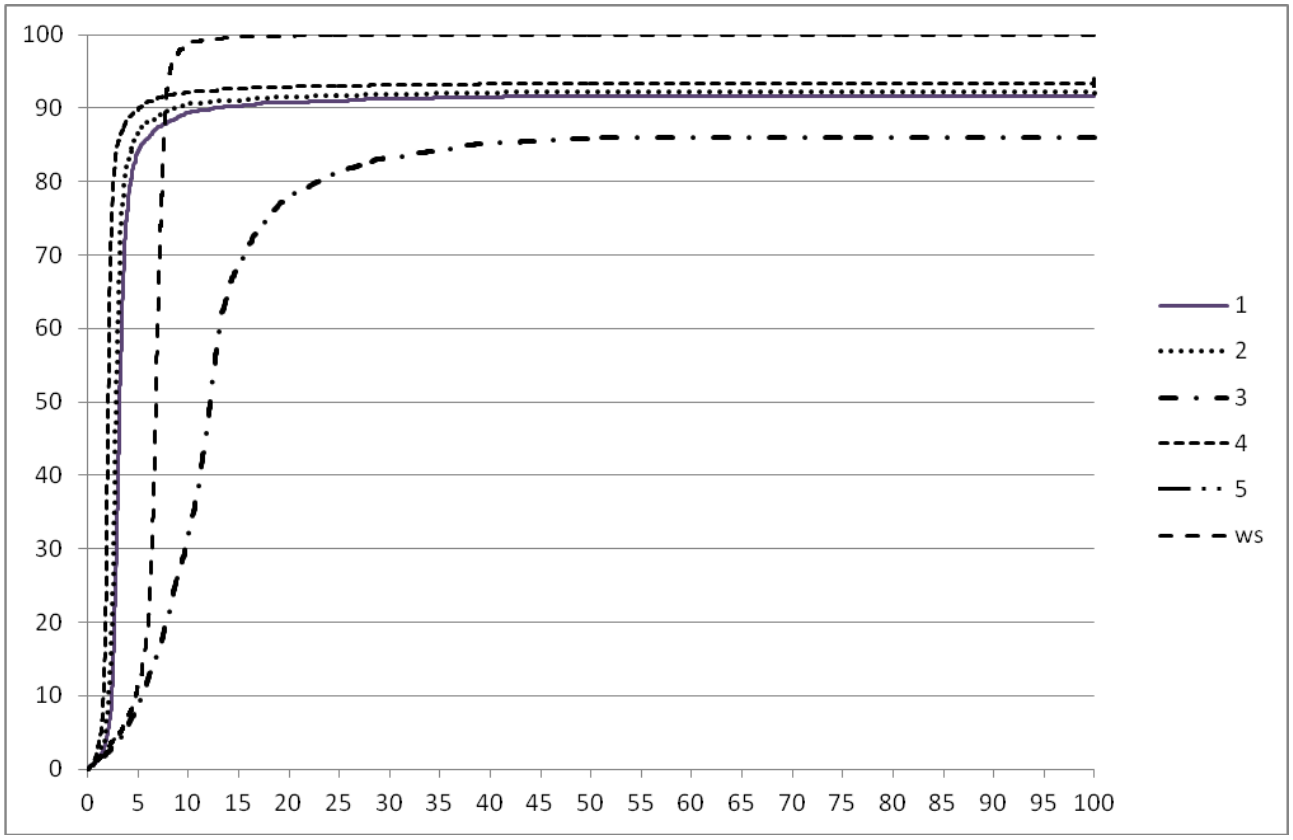

Рис. 1. ROC-диаграмма для первого множества изображений

Для сравнения результатов обнаружения срытой информации был использован лучший стегоанализ LSB-замещения, который называется улучшенный WS-анализ, см.[6]. Поскольку сдвиговый метод даёт количественную оценку скрытой информации, то результаты были приведены на ROC-диаграммах (по оси абсцисс – вероятность ложного обнаружения, по оси ординат – вероятность обнаружить в заполненном). Вначале были выбраны два лучших метода сжатия на первом множестве изображений (при 5%-ом внедрении) и после этого метод был протестирован на двух остальных множествах.

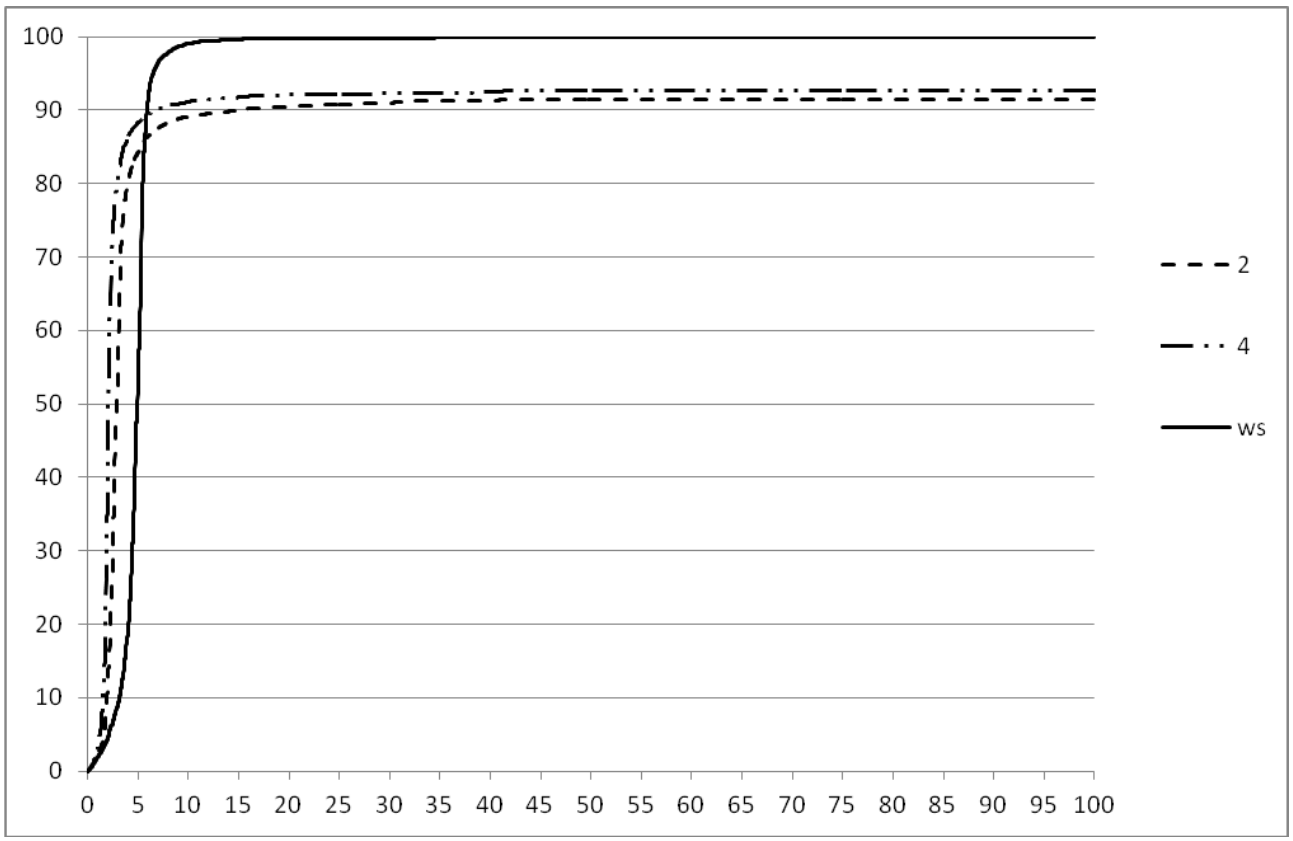

Рис. 2. ROC-диаграмма для второго множества изображений

Из первого графика видно, что WS-метод существенно хуже при малой вероятности ложного обнаружения. В [7] A. Кер предлагает называть метод «хорошим» («reliable»), если при 5%-ом ложном обнаружении вероятность правильно обнаружить больше 50%. Из первого графика следует, что на данном множестве метод WS не является «хорошим», в то время как сдвиговый метод удовлетворяет этому условию с запасом. Более того, 50%-ый порог обнаружения сдвиговый метод достигает при 2%-ах ложных определений! WS-метод начинает хорошо работать только на множествах два и три, но и на множестве два он существенно хуже метода сдвига. Объяснить эффективность метода, основанного на сжатии, можно тем, что он учитывает зависимость между цветовыми компонентами изображения, когда как WS-метод может работать только с одной компонентой, например компонентой красного цвета, и не учитывает зависимости цветов, которые нарушаются при внедрении скрытой информации. Объяснить то, что WS-метод начинает обгонять сдвиговый метод на третьем множестве, можно следующим: изображения третьего множества содержат значительно меньше «шумов» и поэтому у WS-метода значительно меньше ложных срабатываний, в то время как эффективность сдвигового метода почти не меняется от множества к множеству.

Из результатов работы следует, что сдвиговый метод можно рекомендовать для стегоанализа цветных изображений, если изображения содержат много естественного «шума».

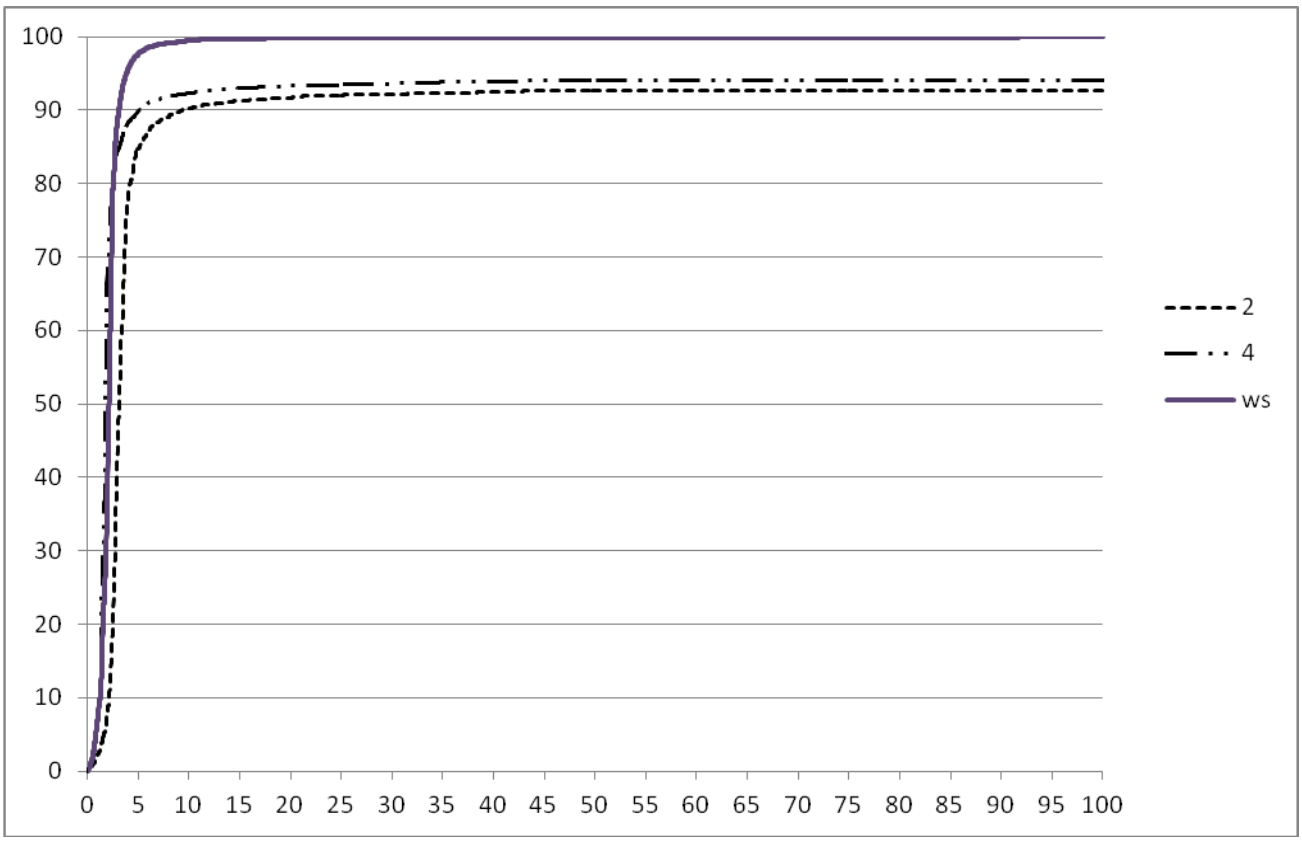

Рис. 3. ROC-диаграмма для третьего множества изображений

Литература

1. *Жилкин М.Ю., Меленцова Н.А., Рябко Б.Я.* Метод выявления скрытой информации, базирующийся на сжатии данных // Вычислительные технологии. 2007. Т. 12. Специальный выпуск 4. С. 26-31.
2. *Ker A.D*, Improved detection of LSB steganography in grayscale images // In Proc. 6th Information Hiding Workshop, LNCS 3200, pp, 97-115. Springer, 2004.
3. *Fridrich J., Goljan M., Holotyak T*, New Blind Steganalysis and its Implications // Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, vol. 6072, San Jose, CA, January 16-19, pp. 1-13, 2006.
4. *Dumitrescu, S., Wu, X., Wang, Z.*: Detection of LSB steganography via sample pair analysis // In: Proc. Information Hiding Workshop. Volume 2578 of Springer LNCS., pp. 355–372, 2002.
5. *Ryabko B., Ryabko D.* Provably secure universal steganographic systems // Cryptology ePrint Archive. 2006. Report 2006/063. URL: `http://eprint.iacr.org/2006/063` (дата обращения: 23.08.2007).
6. *Ker, A., Bohme, R.*: Revisiting Weighted Stego-Image Steganalysis // In Delp III, E.J., Wong, P.W., eds.: Security and Watermarking of Multimedia Contents X. Volume 6819 of Proc. SPIE. (2008) 501-517.
7. *Ker, A.*: Improved detection of LSB Steganography in grayscale images // In: Proc. Information Hiding Workshop. Volume 3200 of Springer LNCS. (2004) 97–115.

Статья поступила в редакцию 19.11.2012

Монарев Виктор Александрович
к.ф.-м.н, научный сотрудник ИВТ СО РАН, e-mail: vitkor.monarev@gmail.com.

Shift Method for Steganography

V. Monarev

In this paper, a new method of stenography is described. This method can be applied for revealing information hidden with LSB method. Experimental data are provided enabling to compare efficiency of the given method with the best known methods of stegoanalysis.

Keywords: steganography, stegoanalysis.