МЕТОД ВЫЯВЛЕНИЯ СКРЫТОЙ ИНФОРМАЦИИ, БАЗИРУЮЩИЙСЯ НА СЖАТИИ ДАННЫХ*

М. Ю. Жилкин, Н. А. Меленцова
Сибирский государственный университет телекоммуникаций и информатики, Новосибирск, Россия
e-mail: myz@csu.ru, mele_na@rambler.ru

Б. Я. Рябко
Институт вычислительных технологий СО РАН, Новосибирск, Россия
e-mail: boris@ryabko.net

A method for detecting the presence of a "latent" information in the BMP graphic data is proposed. It is experimentally shown that the method is more efficient than the ones known earlier.

Введение

Бурное развитие вычислительной техники и появление новых систем передачи информации приводят к широкому использованию стеганографических алгоритмов, т. е. таких методов передачи данных, когда сам факт передачи скрыт от постороннего наблюдателя. Обычно скрыто передаваемые сообщения встраивают в "невинные" данные — фотографии, спам, видео, аудиозаписи и т. п. Возникновение проблемы защиты прав собственности на информацию, представленную в цифровом виде, приводит к развитию еще одного вида стеганографии — так называемых водяных знаков, которые являются специальной меткой, незаметно внедряемой в изображение или другой носитель информации с целью тем или иным образом контролировать его использование. Методы современной компьютерной стеганографии находят применение в области военной и правительственной связи, защиты авторских прав и других областях [1]. Очевидно, что методы стеганографии могут быть использованы и самыми разнообразными "злоумышленниками", поэтому становится актуальной задача так называемого стегоанализа, предназначенного для обнаружения "скрытой" информации.

В настоящее время секретные данные встраиваются в сообщения самой разной природы, обычно называемые "контейнерами". Контейнер называется пустым, если он не содержит скрытой информации, в противном случае он называется стегоконтейнером или заполненным контейнером.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 06-07-89025).

Целью данной работы является построение эффективного метода "автоматического" (т. е. без участия человека) определения факта наличия скрытых данных в графических файлах формата BMP. Стоит отметить, что этот формат довольно распространен. В настоящее время существуют многочисленные алгоритмы и программы, предназначенные для встраивания "скрытой" информации в файлы формата BMP (см., например, обзор в [1, 2]). Задача стегоанализа данных в формате BMP также привлекает многих исследователей [3 – 6].

В настоящей работе предлагается новый метод стегоанализа, основанный на применении сжатия данных. Среди достоинств разработанного метода следует подчеркнуть высокую эффективность, скорость, применимость к широкому классу методов модификации LSB (least significant bits) пикселей изображения, а также то, что стегоанализ проводится без участия человека.

1. Методы включения скрытой информации

Известные на сегодняшний день стеганографические программные пакеты используют различные методы включения данных в BMP-файлы путем модификации младших значащих бит. Одним из распространенных является *метод последовательного заполнения*, в котором пиксели изображения изменяются один за другим от начала изображения. Если объем секретных данных меньше емкости контейнера, то оставшиеся пиксели не модифицируются. Данный метод прост в реализации, однако и наличие скрытых данных в таком контейнере достаточно легко определяется известными статистическими тестами [7], а в ряде случаев даже при визуальном исследовании.

Метод последовательного заполнения совместно с заполнением до конца файла случайными данными отличается от предыдущего тем, что в неиспользованный остаток изображения включается случайная последовательность данных. Данный подход упрощает задачу определения факта наличия скрытых данных за счет того, что даже при небольшом объеме скрытой информации контейнер заполняется полностью. Однако одновременно усложняется процесс локализации "вложенной" информации.

В отличие от вышеприведенных, *метод "случайного" ("разбросанного") заполнения* делает практически невозможным визуальное определение наличия скрытых данных. Здесь посредством псевдослучайного генератора выбираются пиксели, в которые будет включена секретная информация. Такие контейнеры труднее всего подвергаются стеганографическому анализу. Известные в литературе [2] атаки позволяют обнаружить факт наличия скрытых данных при заполнении изображения-контейнера не менее чем на 50 % от его емкости. Под емкостью контейнера понимается максимальный объем информации, который можно в него "спрятать". Емкость зависит от характеристик контейнера, таких как формат, размер файла и т. д., и от выбора алгоритма включения данных в этот контейнер; обычно она не превосходит 12.5 % от емкости контейнера.

Остановимся на описании формата BMP. Он имеет много разновидностей, составляющих два класса форматов — палитровые (индексированные) и беспалитровые. При этом сложность стегообработки палитровых форматов делает их применение редким. Наиболее часто в качестве контейнеров используются беспалитровые форматы. Следует отметить, что один и тот же стеганографический алгоритм нельзя использовать для включения данных методом замены младших значащих бит для обоих классов.

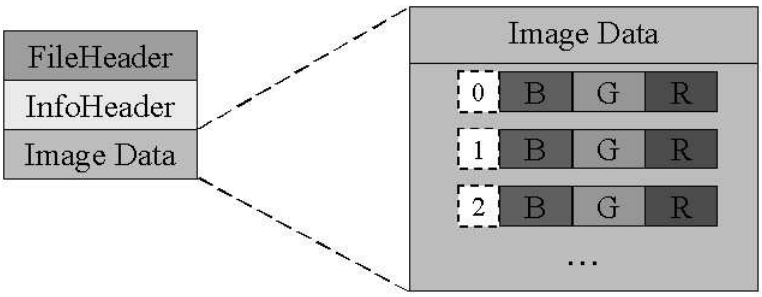


Рис. 1. Структура беспалитрового BMP-файла

BMP-файл состоит из заголовка — структуры размером в несколько десятков байт, содержащей основные параметры изображения (размеры, глубину цвета и т. д.). Следующим элементом структуры является палитра — массив, описывающий цвета, используемые в изображении. Каждый элемент палитры состоит из четырех байт (R-, G-, B-компоненты и альфа-канал). Размер палитры 1 кб. Палитра не является обязательной в некоторых типах BMP-файлов. Последняя часть — область данных — содержит последовательность кодов пикселей изображения.

В беспалитровых изображениях область данных содержит описание всех цветовых компонентов каждого пикселя изображения (рис. 1). В этом случае скрытые данные включаются без всяких дополнительных преобразований. Самым простым и поэтому наиболее распространенным форматом данного класса является 24-битный BMP. По этой причине в работе исследуются изображения только такого формата.

Рассматривается задача разработки метода автоматического определения факта наличия скрытых данных без привлечения на отдельных этапах аналитических способностей человека. Это, в частности, позволит значительно ускорить процесс тестирования серий из большого числа изображений при реализации метода в качестве “фильтра” в системах централизованного управления Интернет-трафиком (маршрутизаторы, прокси-серверы и т. д.) для организации автоматизированного поиска “скрытых” данных в пересылаемой информации.

Предлагаемый нами метод работает на контейнерах с разными способами наполнения, однако результаты приводятся для “случайного” (“разбросанного”) способа, так как этот вариант наиболее труден для стегоанализа; для других методов наполнения эффективность метода выше.

2. Описание метода

Новый подход в стегоанализе базируется на сжатии данных. При этом в предлагаемом методе могут применяться широко распространенные программы-архиваторы.

Идея метода опирается на то, что включаемые данные статистически независимы от “контейнера”, поэтому при сжатии объем полученного архивного файла возрастает по сравнению со сжатием исходного (“пустого”) контейнера.

Метод состоит в сравнении коэффициентов сжатия исследуемого контейнера и его копии, заполненной случайной последовательностью данных. Если коэффициенты близки по значению, т. е. их разность достаточно мала, то весьма вероятно, что исходный файл содержал скрытую информацию. При практической реализации этой идеи контейнер и его заполненная копия рассматриваются не целиком, а делятся на несколько

равных частей, после чего оцениваются коэффициенты сжатия соответствующих частей. Это позволяет увеличить точность метода.

Разработанный алгоритм выглядит следующим образом. Пусть $X = \{x_1, \ldots, x_N\}$ — последовательность байтов в поле данных изображения BMP, где $|X| = N$ — длина последовательности. Разобьем последовательность X на d равных отрезков и обозначим каждый отрезок X_i, где $i = 1, 2, \ldots, d$. Пусть $\psi(X)$ — алгоритм сжатия, примененный к последовательности X. Введем величину

$$f(X, n) = \frac{|\psi(X_n)|}{|X_n|},$$

которую назовем коэффициентом сжатия отрезка n последовательности X алгоритмом ψ.

Обозначим через $\varphi(X)$ псевдослучайное изменение младших бит всех байтов последовательности X. Пусть X — последовательность, которая подается на вход программе, а $Y = \varphi(X)$ — полученная из нее новая последовательность ("заполненный" контейнер). Исходная последовательность X должна сжиматься "сильнее" по сравнению с измененной последовательностью Y, если исходный контейнер не заполнен или частично заполнен.

Введем новую величину

$$\delta(X, n) = |f(X, n) - f(Y, n)|.$$

Если отрезок X_i последовательности X содержит "скрытую" информацию, то коэффициент $f(X, i)$ и отвечающий ему $f(Y, i)$ различаются незначительно и, напротив, пустой участок сжимается лучше заполненного. Для определения факта включения информации выбирается пороговое значение для величины δ и производится оценка количества отрезков, на которых значение величины не превышает порог. Если таких отрезков больше $d/2$, то считается, что исходная последовательность X содержала скрытые данные, в противном случае последовательность X считается пустой. Порог можно варьировать, регулируя тем самым частоту ошибок программы на пустых и непустых контейнерах.

3. Экспериментальный анализ

Для экспериментального исследования метода подготовлена серия из 1000 случайно отобранных изображений ("контейнеров") формата 24-бит BMP. На вход программы подавались исходные контейнеры, контейнеры, заполненные на 10, 20 % и т. д. от своей емкости. Были проверены различные методы заполнения контейнеров, архиваторы и значения порогов δ.

Назовем для краткости ошибкой первого рода случай, когда метод принимает пустой контейнер за заполненный, и наоборот, назовем ошибкой второго рода случай, когда непустой контейнер принимается за пустой.

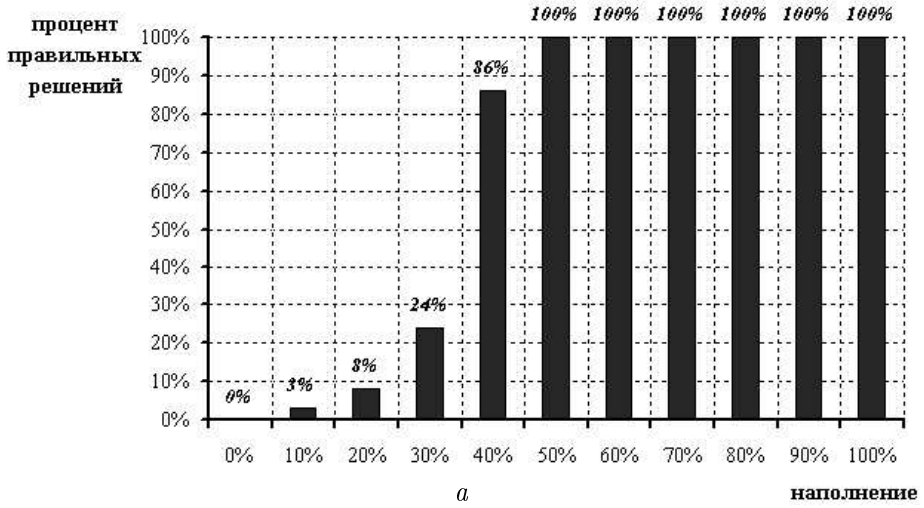
В качестве архиваторов использовались RAR, ZIP, GZIP, BZIP2. Архиватор RAR показал минимальную по сравнению с другими ошибку первого рода (табл. 1). Архиваторы ZIP и BZIP2 продемонстрировали минимальные ошибки второго рода (табл. 2). Анализ представленных данных приводит к выводу, что лучшим архиватором из рассмотренных по результатам тестирования является ZIP, поскольку он наиболее удачно сочетает малые значения обеих ошибок тестов.

Т а б л и ц а 1. Ошибка теста на пустых контейнерах в зависимости от величины порога δ, %

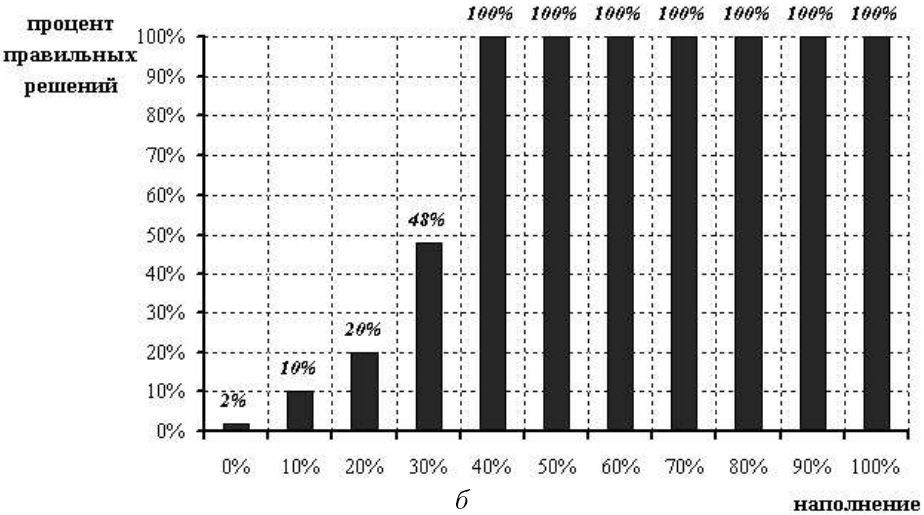
Архиватор	Пороговое значение δ												
	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
RAR	0	0	0	0	1	1	1	2	2	3	3	3	3
ZIP	0	1	1	1	1	1	2	2	2	4	4	4	5
GZIP	0	0	1	1	1	2	2	2	3	3	3	3	4
BZIP2	0	0	1	1	2	2	3	3	5	6	6	7	7

Т а б л и ц а 2. Правильные решения теста при величине порога $\delta = 1\,\%$

Архиватор	Степень наполнения контейнера, %										
	0	10	20	30	40	50	60	70	80	90	100
RAR	0	3	6	21	89	99	98	98	98	98	98
ZIP	1	3	12	30	94	100	100	100	100	100	100
GZIP	1	3	12	28	93	100	100	100	100	100	100
BZIP2	1	4	17	46	95	100	100	100	100	100	100



а



б

Рис. 2. Результаты тестов с архиватором ZIP на "разбросанном" заполнении: *а* — $\delta = \delta_{\min}$; *б* — $\delta = \delta_{\max}$

Проведено исследование влияния выбора порогового значения на результаты теста. Тест не работает при $\delta = 0\,\%$, при возрастании δ уменьшается ошибка второго рода, но вместе с этим увеличивается ошибка первого рода (рис. 2; табл. 1 и 2). Эмпирически были подобраны два значения: $\delta_{\min} = 0.8\,\%$ и $\delta_{\max} = 1.6\,\%$. Такое поведение ошибок в зависимости от δ позволяет выбрать два значения для пороговой величины: минимальное и максимальное. Значение $\delta_{\min}$ практически не вызывает ошибок первого рода (т. е. ложных срабатываний на пустых контейнерах), а значение $\delta_{\max}$ позволяет достичь "золотой середины" — небольшого значения ошибки второго рода при приемлемой ошибке первого рода.

Таким образом, два предлагаемых варианта достаточно эффективно позволяют решать задачу определения скрытой информации. Предлагаемый метод в отличие от ранее известных позволяет надежно и без участия человека выявлять факт включения данных в изображение формата 24-бит BMP. Ошибка при заполнении контейнера на $50\,\%$ и более не превышает $3\,\%$. Особенностью метода является наличие параметров, позволяющих регулировать чувствительность алгоритма.

Список литературы

[1] Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. М.: Солон-Пресс, 2002. 272 с.

[2] Westfeld A., Pfitzmann A. Attacks on steganographic systems. Breaking the steganographic utilities EzStego, Jsteg, Steganos and S-Tools— and some lessons learned // Lecture Notes in Comput. Sci. 2000. Vol. 1768. P. 61–75.

[3] Provos N., Honeyman P. Hide and seek: an introduction to steganography // IEEE Security & Privacy. 2003. Vol. 5. P. 32–44.

[4] Dabeer O., Sullivan K., Madhow U. et al. Detection of hiding in the least significant bit // IEEE Trans. on Signal Proc. 2004. Vol. 52. P. 3046–3058.

[5] Fridrich J., Goljan M., Du R. Reliable detection of LSB steganography in color and grayscale images // Proc. of the ACM Workshop on Multimedia and Security. 2001. P. 27–30.

[6] Mitra S., Roy T., Mazumdar D., Saha A.B. Steganalysis of LSB encoding in uncompressed images by close colour pair analysis // Indian Institute of Technology at Kanpur Hacker's Workshop. Feb. 2003.

[7] Ryabko B.Ya., Fionov A.N. Basics of Contemporary Cryptography for IT Practitioners. World Sci. Publ. Co., 2005.

Поступила в редакцию 18 июня 2007 г.