УДК 003.26

Построение стегосистемы на базе растровых изображений с учётом статистики младших бит

Е. Ю. Елтышева, А. Н. Фионов

Предлагается метод сокрытия информации в младших битах растрового изображения, записанного файлах формата BMP, PNG и др., использующих неискажающие методы сжатия. В отличие от известных аналогов, предлагаемый метод кодирует внедряемое сообщение в соответствии с оценками вероятностей появления значений 0 и 1 в младших битах с целью минимального искажения статистики, свойственной "пустому" файлу-контейнеру. Проводится универсальный и специфический стегоанализ разработанного алгоритма. Показано преимущество предложенного метода по стойкости по сравнению с известными аналогами.

Ключевые слова: стеганография, стегоанализ, LSB-методы, статистическая модель, арифметическое кодирование, идеальные стегосистемы.

1. Введение

Классическая задача стеганографии состоит в организации передачи секретного сообщения таким образом, чтобы не только содержание сообщения, но и сам факт его передачи были скрыты ото всех, кроме предполагаемого ("законного") получателя. Эта задача обычно решается путём встраивания секретного сообщения в некоторое другое сообщение, называемое *контейнером*, содержание и факт передачи которого не могут вызывать никаких подозрений. Например, в качестве контейнеров могут выступать файлы, содержащие цифровые фотографии, музыку или видео. Такие типы файлов используются наиболее часто, т.к., с одной стороны, предоставляют достаточно много возможностей и места для размещения скрытых сообщений, а с другой стороны, их пересылка по компьютерным сетям является совершенно обычным делом. Например, вряд ли у кого вызовет подозрение серия фотографий, пересылаемая по электронной почте или выложенная на сайте в Интернете, с помощью которой один человек делится с другим, скажем, впечатлениями о своей туристической поездке. Заметим, что нет никаких видимых причин, запрещающих выбирать в качестве контейнеров "удобные" файлы, например, фотографии только природных объектов (как будет показано далее, такие фотографии в наибольшей степени подходят для предлагаемого в настоящей работе метода). При этом важно соблюдение одного условия: никто не должен иметь доступ одновременно к файлу, содержащему скрытое сообщение, и к исходному файлу, выбранному в качестве контейнера (в противном случае задача выявления скрытого сообщения тривиальна и сводится к простому сравнению файлов).

В настоящей работе в качестве контейнера используется 24-битовое растровое изображение в системе цветности RGB. Изображение представляется в виде матрицы, каждый элемент которой (пиксел) соответствует видимой точке и задаётся значениями яркости трёх составляющих — красного (R), зелёного (G) и синего (B) цвета. Яркость каждой составляю-

щей записывается 8-битным числом и, следовательно, может изменяться в диапазоне от 0 до 255 (значения 0, 0, 0 соответствуют чёрному цвету, а 255, 255, 255 — максимально яркому белому). Скрываемая информация записывается в младшие (наименее значимые) биты цветовых составляющих. Это позволяет отнести предлагаемый метод к классу так называемых LSB-методов[1], широко применяемых в стеганографии. Такой тип используемого контейнера может иметь внешнее представление в виде файлов формата BMP, TIFF, PNG, PCX, TGA и др. Эти форматы файлов строятся с использованием неискажающих методов сжатия, полностью сохраняющих все биты изображения (в BMP сжатие обычно не применяется), и поэтому между ними существует взаимно однозначное соответствие (для рассматриваемого типа изображения). В своих экспериментах, описываемых в настоящей работе, мы используем BMP-файлы из-за простоты их программной обработки, но все полученные результаты автоматически переносятся на случай сохранения изображения в файлах других указанных форматов.

Перечисленные выше форматы графических файлов противополагаются форматам, использующим сжатие с потерями, таким как JPEG. При сжатии с потерями удаётся достичь значительно меньших размеров файлов, чем при сжатии без потерь. Формат JPEG является сегодня преобладающим форматом для хранения и пересылки цифровых фотографий, но для него LSB-методы внедрения скрытой информации не применимы, т.к. JPEG не гарантирует целостность младших бит. Вместе с тем, в последнее время в связи с прогрессом в области систем памяти и каналов связи усиливается интерес к графическим форматам с неискажающим сжатием, т.к. они обеспечивают наиболее высокое качество воспроизведения. Так, формат TIFF активно используется в полиграфии. Многие современные цифровые фотоаппараты позволяют сохранять изображения в формате TIFF. Наконец, новые стандарты представления графической информации JPEG-2000 и HD-Photo содержат опции для неискажающего сжатия растровых изображений. Всё это делает задачу построения и исследования стеганографических методов LSB-типа для растровых изображений в качестве контейнеров весьма актуальной и интересной.

Традиционные реализации LSB-методов в стеганографии выполняются по следующей схеме: вначале сообщение шифруется с использованием секретного ключа, затем биты зашифрованного сообщения записываются на место младших бит цветовых составляющих изображения последовательно или в порядке, задаваемом псевдослучайными индексами, формируемыми на основе того же секретного ключа (так называемое "рассеянное заполнение"). Примером таких реализаций могут служить программы HIDE4PGP и STEGOTOOLS, находящиеся в свободном доступе в Интернете. Максимальное количество информации, которое можно записать в контейнер при использовании некоторого метода внедрения, можно назвать *эмпирической ёмкостью* контейнера. Эмпирическая ёмкость определяется не столько контейнером (его подлинная ёмкость нам не известна), сколько методом. Например, при внедрении информации путём замены младших бит цветовых составляющих эмпирическая ёмкость изображения равна 1/8 от его полного размера. В дальнейшем для краткости мы будем опускать слово "эмпирическая", говоря о ёмкости контейнера. При использовании в качестве контейнера BMP файла его ёмкость приближённо равна 1/8, или. 12.5 %, от размера файла (мы пренебрегаем размером заголовка, который много меньше размера изображения). Как правило, стеганографические программы позволяют задать степень заполнения контейнера в виде некоторой доли от ёмкости. Предполагается, что при малых значениях степени заполнения изображение меньше искажается и факт наличия встроенного сообщения определить труднее.

Первое условие, которое выполняют все стеганографические программы, состоит в том, что искажения, вносимые в контейнер при внедрении сообщения, должны быть незаметны для человека. Так, даже при стопроцентном заполнении, т.е. при замене всех младших бит

[1] от англ. Least-Significant Bit — наименее значимый бит.

матрицы изображения, мы обычно не видим искажений, особенно, если под рукой нет исходного файла для сравнения. Однако более тонкие компьютерные методы стегоанализа могут всё же определить наличие встроенного сообщения. Наиболее новым и мощным средством такого анализа является использование методов универсального кодирования, применяющихся для сжатия данных. Этот подход был предложен Б. Я. Рябко и развит в ряде работ для решения множества статистических задач [1–5]. Основная идея его применения в целях стегоанализа состоит в том, что внедряемое сообщение нарушает статистическую структуру контейнера, повышает его энтропию, поэтому заполненный контейнер будет сжиматься хуже, чем исходный (незаполненный). По сути дела, для выявления факта наличия скрытого сообщения нам нужно протестировать статистическую независимость последовательности, образованной младшими битами, от остальной информации в контейнере. Это одна из задач математической статистики, решение которой с помощью методов сжатия данных было предложено в [4]. В работе М. Ю. Жилкина [6] этот метод был с успехом применён для стегоанализа графических файлов. Ему удалось построить систему, надёжно выявляющую скрытые сообщения в BMP-файлах при уровне заполнения больше 40 %.

Вместе с тем в ряде работ [7–10] была показана принципиальная возможность построения так называемых *идеальных* стегосистем, т.е. систем, в которых наличие скрытых сообщений выявить невозможно. Один из способов построения таких систем состоит в особом кодировании сообщений перед внедрением, обеспечивающем вероятности появления кодовых символов, в точности равные вероятностям заменяемых элементов контейнера. В результате статистическая структура контейнера не искажается, он становится статистически неотличимым от всех других (пустых) контейнеров, а следовательно, и факт наличия в нём скрытого сообщения выявить невозможно. Для осуществления указанного особого вида кодирования могут быть применены так называемые коды со стоимостью, хорошо изученные в теории кодирования источников, в частности, омофонные коды со стоимостью [10,11].

Цель настоящей работы состоит в построении стегосистемы, приближающейся по своим свойствам к идеальной и использующей в качестве контейнеров растровые изображения. Основная практическая трудность построения такой системы заключается в том, что статистика младших бит в цветовых составляющих и их зависимость от остальной информации в изображении изначально не известны. Разработанные в теории кодирования источников подходы позволяют оценивать вероятности с помощью стчётчиков встречаемости, формируемых в некотором контексте предшествующих символов. Специфика стеганографии, однако, заключается в том, что если внедрять данные непосредственно вместо бит с оценённой статистикой, то декодер не сможет воспроизвести оценки, полученные при кодировании, что приведёт к невозможности декодирования сообщения. Поэтому мы предлагаем делить изображение на две части — одну, используемую для получения оценок статистики, и вторую – для внедрения данных. Основное эвристическое предположение состоит при этом в том, что статистические структуры обеих частей близки. При равенстве размеров частей (что предполагается в настоящей работе) эффективная ёмкость контейнера сразу падает вдвое, т.е. составляет 50 % от его полной ёмкости. Построение надёжных стегосистем при таком уровне заполнения всё ещё актуально в виду упомянутых выше данных стегоанализа BMP-файлов [6].

В результате проведённых исследований был построен алгоритм внедрения скрытых сообщений в растровые изображения, учитывающий вероятности младших бит, оцениваемые с помощью определённой статистической модели. Стегоанализ Жилкина, проведённый на случайной выборке из 150 BMP-файлов, показал, что при уровне заполнения контейнера около 50 % предложенный алгоритм позволяет скрыть наличие встроенного сообщения в 15 % файлов. В это же время для других известных методов (при таком же уровне заполнения) в случайной выборке не нашлось ни одного файла, в котором факт наличия скрытого сообщения не удалось бы выявить. Полученный результат свидетельствует о целесообразности учёта статистики младших бит при внедрении сообщений, однако построенная стегоси-

стема ещё далека от идеальной. Ясно, что результат применения разработанного алгоритма может быть значительно улучшен, если в качестве контейнеров используются только "удобные" файлы, относящиеся к указанным выше 15 %.

Стегоанализ Жилкина является универсальным методом, не принимающим во внимание детали реализации алгоритма встраивания информации. Вместе с тем для предложенного алгоритма легко построить специфический метод стегоанализа, заключающийся в сравнении степени сжатия двух частей изображения. В силу неидеальности алгоритма степень сжатия первой части, которая использовалась только для оценивания статистики, будет заметно больше, чем степень сжатия второй, в которую внедрялась информация. Для предотвращения возможности такого стегоанализа был предложен метод перемешивания при разделении изображения на две части, определяемый псевдослучайной последовательностью, производной от секретного ключа, с помощью которого шифруется само встраиваемое сообщение. Без знания ключа невозможно разделить изображение на две части таким же образом, как это делалось при внедрении сообщения, и описанная специфическая атака становится неприменимой. Как показали эксперименты, перемешивание при разделении изображения практически не влияет на результаты универсального стегоанализа Жилкина.

План статьи следующий. В разд. 2 описываются этапы построения предлагаемого алгоритма: выбор методов кодирования, разделения изображения и оценки статистики. В разд. 3 проводится анализ алгоритма с помощью универсального стегоанализа. Наконец в разд. 4 рассматривается специфический стегоанализ предложенного алгоритма.

2. Построение алгоритма

2.1. Выбор метода кодирования

Для предлагаемого алгоритма внедрения информации необходим метод кодирования, преобразующий зашифрованное сообщение в поток бит, в котором нули и единицы появляются с определёнными задаваемыми вероятностями (биты зашифрованного сообщения считаются равновероятными и независимыми). В работах [9, 10] предложены методы кодирования, позволяющие точно решить эту задачу. Эти методы использовались нами на начальном этапе исследований, однако мы пришли к выводу, что их использование в рассматриваемой стегосистеме не оправдано, т.к. распределение вероятностей кодовых символов не является точным, а только оценивается, но, в то же время, методы [9, 10] сильно "растягивают" сообщение, стремясь достичь точного соответствия получаемого распределения кодовых символов заданному. Более эффективным оказалось приближённое решение задачи кодирования, допускающее некоторое (небольшое) отклонение распределения кодовых бит от заданного. Это решение достигается за счёт арифметического декодирования входного зашифрованного сообщения. Арифметическому декодеру указывается требуемое распределение вероятностей нуля и единицы для каждого следующего выходного символа, в соответствии с которым он и воспроизводит (декодирует) этот символ, рассматривая зашифрованное сообщение как код, построенный ранее соответствующим арифметическим кодером. Отклонение получаемого распределения вероятностей символов от заданного определяется избыточностью арифметического кодирования, которая на основании оценки, полученной в [12], составляет примерно 0.002 бита на символ при 16-разрядной арифметике и 0.00000003 бита на символ при 32-разрядной арифметике. Как будет видно в дальнейшем из общего результата, такая неточность воспроизведения распределения вероятностей пренебрежимо мала по сравнению с неточностью оценки самого распределения.

Чтобы восстановить зашифрованное сообщение из его кода, требуется, наоборот, применить арифметический кодер, которому указываются те же самые распределения вероятностей, что и декодеру.

2.2. Выбор способа разделения изображения

Как уже говорилось во введении, для оценки статистики младших бит и внедрения информации с учётом этой статистики необходимо разделить исходное изображение на две части: первую часть ("обучающую"), по которой будет оцениваться статистика, и вторую ("рабочую"), в которую будет внедряться сообщение. Отметим ещё раз, что мы не можем изменять биты первой части, т.к. в этом случае декодер не сможет получить те же самые оценки статистики, чтобы восстановить скрытое сообщение. Данная особенность значительно отличает кодирование в стеганографии от кодирования при сжатии данных. Разделение изображения на две части необходимо выполнить таким образом, чтобы статистические структуры частей были максимально близки. Интуитивно ясно, что если речь идёт об изображении, то деление его, скажем, на верхнюю и нижнюю половины – плохой способ. Действительно, если на верхней половине фотографии запечатлено небо, а на нижней трава, то вряд ли их статистические структуры будут близки. Наиболее целесообразным кажется такое разделение, при котором в разные части попадают близко идущие пикселы, которые в типичных изображениях (при достаточно высоком разрешении) почти одинаковы. Особенно свойство близости соседних пикселов характерно для изображения природных объектов, в которых нет резких переходов цвета и яркости, в отличие от искусственных объектов, например, черных линий на белом фоне. Основываясь на этих рассуждениях, мы исследуем 3 способа разделения, измеряя их статистическую близость как визуально, так и по разнице длин их сжатых представлений.

Обозначим матрицу исходного изображения размером $h \times w$ через S (здесь h – высота изображения, или число строк в матрице, а w – ширина, или число столбцов), будем начинать нумерацию элементов с единицы. Строго говоря, каждый элемент изображения содержит значения трёх цветовых составляющих (R, G, B). При построении статистической модели мы будем исходить из стандартного при обработке изображений предположения о независимости цветовых составляющих и производить над ними одинаковые действия. Поэтому для простоты описания условимся под элементом матрицы S понимать только одну обобщённую цветовую составляющую, определяющую действия над всеми тремя. Обозначим обучающую и рабочую части через E и W соответственно, число строк и столбцов в них будет зависеть от способа разбиения. Опять-таки, ради простоты описания, будем считать, что h и w – чётные числа.

Предлагаемые способы разделения частей исходной матрицы показаны схематично на рис. 1. Элементы исходной матрицы, пронумерованные символом 1, отходят к обучающей части, а элементы, обозначенные символом 2, к рабочей.

Способ 1

1	2	1	2
1	2	1	2
1	2	1	2
1	2	1	2

Способ 2

1	1	1	1
2	2	2	2
1	1	1	1
2	2	2	2

Способ 3

1	2	1	2
2	1	2	1
1	2	1	2
2	1	2	1

Рис. 1. Способы разделения исходного изображения

При первом способе разделения результирующие матрицы E и W состоят из h строк и $w/2$ столбцов (изображение в них сжато по горизонтали), при втором способе, наоборот, из $h/2$ строк и w столбцов (изображение в них сжато по вертикали). При третьем способе, когда элементы исходной матрицы выбираются в шахматном порядке, результирующие матрицы могут собираться как по вертикали, так и по горизонтали, но мы отдаём предпочтение вертикальной сборке, т.е. аналогично первому способу. Более формально,

$$E_{i,j} = S_{i,2j-1}, \quad W_{i,j} = S_{i,2j}, \quad j = 1, 2, \ldots, w/2; \quad i = 1, 3, \ldots, h-1;$$

$$E_{i,j} = S_{i,2j}, \quad W_{i,j} = S_{i,2j-1}, \quad j = 1, 2, \ldots, w/2; \quad i = 2, 4, \ldots, h.$$

Чтобы выбрать способ разделения, при котором получаются части с наиболее близкими статистическими свойствами, были проведены эксперименты на серии изображений. Каждое изображение разделялось тремя указанными способами, и полученные части независимо сжимались с помощью архиватора RAR. Анализировалась разность размеров сжатых частей. Результаты измерений для четырёх типичных изображений приведены в табл. 1.

Таблица 1. Абсолютная разность размеров частей (кбайт) после сжатия архиватором RAR

Изображение	Способ 1	Способ 2	Способ 3
1	6.71	2.62	0.69
2	0.82	1.02	0.63
3	0.14	0.23	0.14
4	0.55	0.14	0.08
среднее	2.06	1.00	0.39

Как видно из табл. 1, третий способ даёт наименьшую разность размеров сжатых частей. Это означает, что при этом способе разделения статистика частей наиболее близка. Именно третий способ мы будем использовать в дальнейшем. Анализируя данные таблицы, отметим также, что первые два представленных в ней изображения – это фотографии искусственных объектов (назовём их объектами класса A), а последние два – фотографии природного типа (класс B). Мы видим, что для объектов класса B разница в степени сжатия частей заметно меньше, чем для объектов класса A. Это означает, что предлагаемый метод внедрения скрытой информации будет более эффективен для файлов класса B, т.к. искажение статистики при разделении частей для таких файлов менее существенно, чем для файлов класса A.

Визуально оценивая разницу между изображениями E и W можно утверждать, что отличие заметно на глаз только на мелких контрастных деталях картинки. Рисунок 2 иллюстрирует пример того, как выглядят обучающая и рабочая части для объекта класса B. Их визуальное сходство, а также представленные выше исследования с помощью архиватора, позволяют нам сделать главное эвристическое предположение, на котором основываются все дальнейшие построения: статистическая модель, построенная для обучающей части изображения E может быть использована как основа для внедрения данных в рабочую часть W.

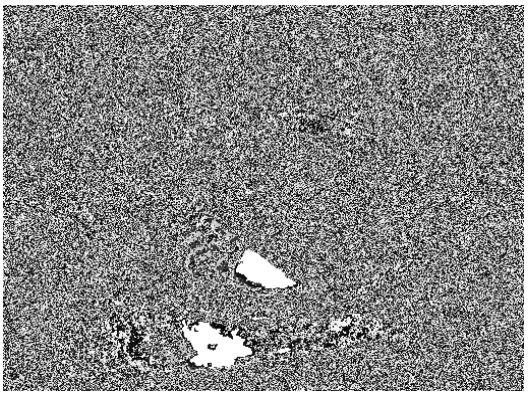
Рис. 2. Две части изображения при разбиении в шахматном порядке

2.3. Построение статистической модели

В данном разделе описывается построение статистической модели обучающей матрицы изображения *E,* с помощью которой будут оцениваться вероятности появления нулей и единиц в младших битах цветовых составляющих.

Прежде всего, рассмотрим распределение младших бит, взятых изолировано. На рис. 3 и 4 показана визуализация младших бит всех трёх цветовых составляющих для изображений класса A и B. Младшие биты визуализированы путём максимального усиления их яркости, т.е. значение 0 не изменяется, а 1 заменяется на 255. Мы видим, что младшие биты в изображении класса A имеют ярко выраженные зоны, в которых они совершенно не случайны. В изображении класса B таких зон нет, хотя местами видны некоторые отклонения от чисто случайного поведения. В большей же части изображений младшие биты (взятые изолированно) выглядят как совершенно случайные. Вместе с тем, известно, что замена младших бит на полностью случайные (что происходит при записи на их место зашифрованного сообщения) приводит к заметному уменьшению степени сжатия файла (об этом свидетельствуют, в частности, результаты работы [6]). Это означает, что существует статистическая зависимость младших бит от остальных элементов изображения. Эта зависимость разрушается при записи в младшие биты независимых данных, что и приводит к росту размера сжатого файла.

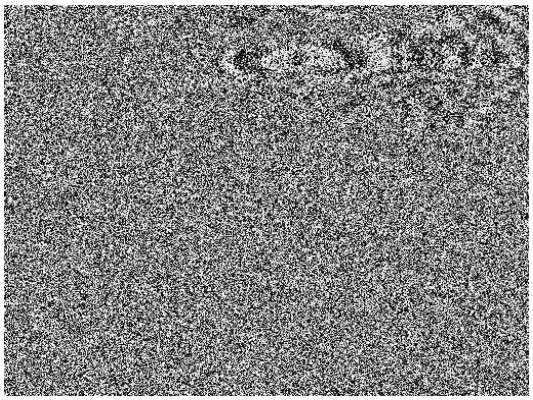
а Исходное изображение



б Младшие биты

Рис. 3. Визуализация младших бит для изображения класса A (табл.1, файл № 2)

а Исходное изображение



б Младшие биты

Рис. 4. Визуализация младших бит для изображения класса B (табл.1, файл № 4)

Оценка статистики младших бит с учётом возможных зависимостей может быть простроена стандартными методами теории кодирования источников. Например, возможна следующая модель. Мы формируем счётчики встречаемости значений 0 и 1 младших бит в контексте более старших бит, а также окружающих пикселов. Если в некотором контексте X значение 0 встретилось $n_{X,0}$ раз, а $1 - n_{X,1}$ раз, то можно получить оценку условных вероятностей по формулам

$$P(0 \mid X) = \frac{n_{X,0} + 1}{n_{X,0} + n_{X,1} + 2}, \quad P(1 \mid X) = 1 - P(0 \mid X) \tag{1}$$

(оценка по Лапласу), либо

$$P(0 \mid X) = \frac{2n_{X,0} + 1}{2(n_{X,0} + n_{X,1}) + 2}, \quad P(1 \mid X) = 1 - P(0 \mid X) \tag{2}$$

(оценка по Кричевскому – Трофимову; прибавление единицы в числителе в обоих случаях необходимо для предотвращения нулевых оценок). Однако на практике возникает следующая проблема: количество различных контекстов (размер пространства состояний модели) слишком велико. Действительно, если учитывать 7 оставшихся бит в байте и 8 окружающих

пикселов, то размер пространства состояний получается равным 2^{71}. Размер изображения, тем более, его половины, не настолько большой, чтобы сформировать представительную статистику в таком огромном пространстве. Большинство состояний будут встречаться только единожды, причём состояния в обучающей части могут не совпадать с состояниями в рабочей, в результате чего мы практически получим модель, приписывающую равные вероятности нулям и единицам. Проблема усугубляется тем, что изображение формируется нестационарным процессом. Статистика может сильно меняться при переходе от одной области изображения к другой (небо и трава на фотографии). Поэтому нужна высокоадаптивная модель, быстро перестраивающая оценки вероятностей при переходе от одного фрагмента к другому. Это приводит к тому, что вероятности должны оцениваться не глобально по всей обучающей части, а в окне сравнительно небольшого размера. Поэтому возникает задача существенного уменьшения размера пространства состояний путём поиска самых существенных элементов контекста, в наибольшей степени статистически связанных с младшими битами.

В качестве критерия при поиске наиболее подходящего контекста будем использовать эмпирическую (частотную) энтропию, которую мы сейчас определим. Пусть n – общее число пикселов, величины $n_{X,0}$, $n_{X,1}$ – как и ранее, число нулей и единиц в некотором контексте X. Обозначим через n_X число появлений самого контекста X. Очевидно, что

$$n_X = n_{X,0} + n_{X,1} \quad \text{и} \quad \sum_X n_X = n$$

(суммирование проводится по всем контекстам X). Тогда частотная энтропия F определяется следующим образом:

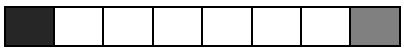
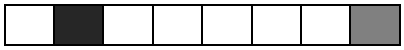
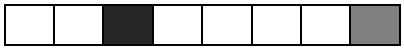
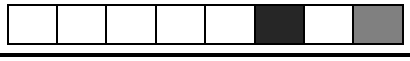
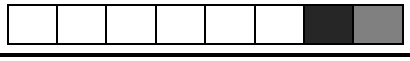
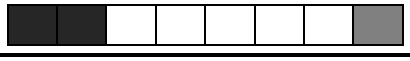
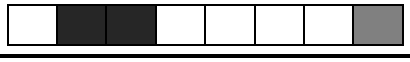
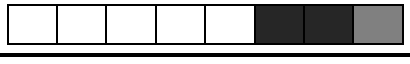
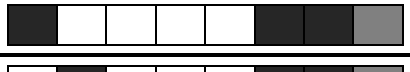
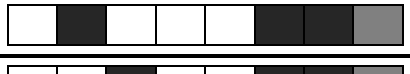
$$F = \sum_X \frac{n_X}{n} \left(\frac{n_{X,0}}{n_X} \log \frac{n_X}{n_{X,0}} + \frac{n_{X,1}}{n_X} \log \frac{n_X}{n_{X,1}} \right)$$

(log обозначает логарифм по основанию 2).

Эта энтропия показывает среднюю длину кодового слова идеального (безызбыточного) кода, который может быть построен при заданных оценках статистики. Чем меньше длина кода, тем "лучше" оценки статистики, т.е. тем ближе статистическая модель к реальному источнику. Таким образом, задача поиска наилучшего способа оценивания статистики младших бит сводится к задаче минимизации частотной энтропии F. Эта задача многопараметрическая: имеется множество возможностей выбора контекста, размера и алгоритма движения окна, в котором оценивается статистика, наконец, многое зависит от характера самого изображения. В настоящей работе мы сделали попытку найти некоторое приближение к действительному минимуму частотной энтропии.

Вначале определим, какие биты в самом пикселе в наибольшей степени влияют на значение его младшего бита. Оценку статистики и вычисление частотной энтропии будем производить в окне (квадрате) размером 8×8 пикселов. Мы выбрали такой размера окна исходя из того, что он используется при сжатии изображений в стандарте JPEG. Авторы стандарта тщательно оптимизировали все параметры алгоритма и их выбор довольно хорошо обоснован. Успехи стандарта JPEG в области сжатия изображений свидетельствуют о том, что окно 8×8 пикселов является достаточно репрезентативной выборкой; с другой стороны, оно довольно мало и позволяет хорошо адаптироваться к изменению характера изображения. Подобно тому как это делается в стандарте JPEG, мы будем передвигать окно на 8 пикселов, т.е. без перекрытия с его предыдущим положением. Эксперименты проводились на десяти изображениях класса B, результаты вычислений по всем окнам и всем изображениям усреднялись для того, чтобы найти наиболее характерные зависимости, свойственные большинству изображений и их фрагментов.

Таблица 2. Значения частотной энтропии F в различных контекстах

№	Контекст	F
1		0.99
2		0.99
3		0.98
4		0.98
5		0.97
6		0.97
7		0.97
8		0.99
9		0.98
10		0.97
11		0.97
12		0.96
13		0.95
14		0.95
15		0.95
16		0.94
17		0.94
18		0.93
19		0.93
20		0.93
21		0.93
22		0.93

Результаты экспериментов приведены в табл. 2. Вначале в качестве контекста выступали одиночные биты (биты, образующие контекст, показаны в таблице темным цветом, младший бит находится справа). Мы видим, что значение F уменьшается при переходе от старших бит к младшим (строки 1–7 таблицы). Это означает, что биты, расположенные "ближе" к младшему, влияют на него сильнее. Данный результат вполне согласуется с "шумовой теорией" младшего бита [13]. Такая же тенденция просматривается, если в качестве контекста брать по два соседних бита (строки 8–13 таблицы). За счёт увеличения размера контекста энтропия несколько снижается. Энтропия ещё немного уменьшается, если рассматривать контексты из трёх смежных бит (строки 14 и 18). Мы приводим результаты не всех проведённых экспериментов, а только наиболее показательные. Так, выяснилось, что контексты, состоящие из несмежных бит, работают хуже (в строках 15–17 приведены примеры таких

контекстов из трёх бит). В результате, для выбранных параметров модели (размера и способа передвижения окна) наилучший результат дал контекст из трёх бит, примыкающих к младшему (строка 18). Попытки дальнейшего увеличения размера контекста при условии, что он строится из бит, примыкающих к младшему (строки 19–22), не привели к существенному уменьшению энтропии, что объясняется недостаточным размером окна для формирования статистики в таких больших контекстах. Контекст, показанный в строке 18, был выбран для дальнейшей работы как самый короткий из всех, обеспечивающих минимум энтропии.

Теперь вспомним о том, что наша конечная цель – используя статистическую модель, построенную на основе матрицы *E*, записать сообщение в матрицу *W*. Алгоритм записи предлагается следующий. Устанавливаем начальное окно в *E* и аналогичное окно в *W* (матрицы *E* и *W* одинакового размера и окна в них располагаются одинаково). В окне матрицы *E* накапливаем статистику, формируя счётчики нулей и единиц в каждом встреченном контексте. От счётчиков переходим к оценкам вероятностей, используя формулы (1) или (2) (экспериментальные результаты, приводимые ниже, получены при использовании оценки по Кричевскому – Трофимову (2)). Затем формируем последовательно каждый младший бит в окне матрицы *W* путём арифметического декодирования внедряемого сообщения с распределением вероятностей, определяемым соответствующим контекстом.

Как уже отмечалось, статистическая структура матриц *E* и *W* близка, но не идентична. Если бы мы строили оценки статистики по матрице *W*, то они были бы немного другие. Нам предстоит оценить конечный эффект от внедрения сообщения в матрицу *W* на основе статистики матрицы *E*. Будем использовать подход, предложенный и обоснованный в [3].

Пусть φ – универсальный кодер (архиватор). Через $\varphi(S)$ будем обозначать код, построенный для исходного изображения *S*, а через $|\varphi(S)|$ – его длину. Обозначим через S^* изображение с внедрённым в него сообщением. Мерой сходства статистических структур изображений (согласно [3]) является разность длин их сжатых представлений. Обозначим эту разность через $\delta = |\varphi(S^*)| - |\varphi(S)|$. Значение δ, близкое к нулю, означает сходство статистических структур: чем δ больше, тем больше статистических различий, т.е. тем хуже скрыта внедрённая информация. Оценим описанный метод внедрения сообщения, используя в качестве кодера стандартный архиватор RAR. Результаты экспериментов на 10 файлах класса B приведены в табл. 3. Все файлы имели исходной размер 576 кбайт и в каждый файл было внедрено 30 кбайт полностью случайных данных (примерно 50 % от ёмкости).

Таблица 3. Разность длин сжатых представлений пустых и заполненных файлов

№	δ, кбайт	№	δ, кбайт
1	1.36	6	4.66
2	0.61	7	1.89
3	0.38	8	1.00
4	5.38	9	5.23
5	3.60	10	0.95
Среднее значение:			2.55

Анализируя данные табл. 3, можно сделать вывод о том, что рассматриваемый метод внедрения сообщений далёк от идеального. Однако, если соотнести разность δ с размером исходного файла, можно сделать предположение, что наш метод всё же затруднит обнаружение факта наличия встроенного сообщения, если стегоаналитику будет представлен только

один файл (без исходного). Но прежде чем перейти к стегоанализу метода, попытаемся улучшить оценку статистики за счёт включения в контекст информации из соседних пикселов.

По аналогии с тем, как это описывалось выше, была проведена серия экспериментов с целью выявления отдельных бит из окружающих пикселов, которые в наибольшей степени влияют на младшие биты. Наилучшим среди исследованных оказался контекст, схематично представленный на рис. 5. К ранее выбранному контексту добавились младшие биты из четырёх окружающих пикселов. Заметим, что с учётом шахматного порядка при разделении исходного изображения, после обратной сборки двух частей левые верхний и нижний пикселы будут расположены соответственно над и под текущим пикселом, для которого определяется зависимость. Отметим одну важную особенность (смешанный контекст): если три бита, примыкающих к младшему, берутся из текущего пиксела рабочей матрицы, то остальные биты контекста берутся из обучающей матрицы, т.к. именно они расположены наиболее близко к текущему пикселу в исходном изображении.

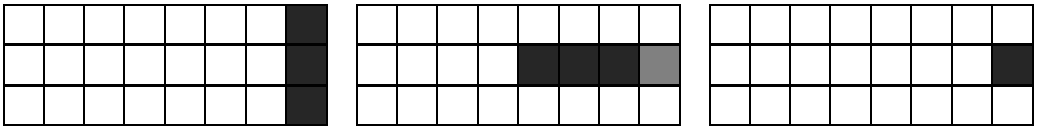


Рис. 5. Расширенный контекст

В связи с тем, что размер контекста увеличился, имеет смысл увеличить размер окна, в котором оценивается статистика. Мы рассмотрели вариант увеличения размера окна до величины 16×16 точек. Такое окно можно сдвигать на его полный размер, но мы также проанализировали возможность его сдвига на 8 точек. В режиме сдвига на 8 точек (см. рис. 6) оценивание статистики по матрице *E* идёт в окне 16×16, а запись данных на основе этой статистики производится в правую нижнюю четверть аналогичного окна в матрице *W*. После этого окно сдвигается по горизонтали на 8 точек, и процесс повторяется, пока не будет пройдена вся горизонтальная полоса изображения. Затем окно содвигается на 8 точек вниз по вертикали и проходится следующая полоса и т.д. (самая верхняя и самая левая полосы изображения обрабатываются немного иначе, но это не имеет большого значения).

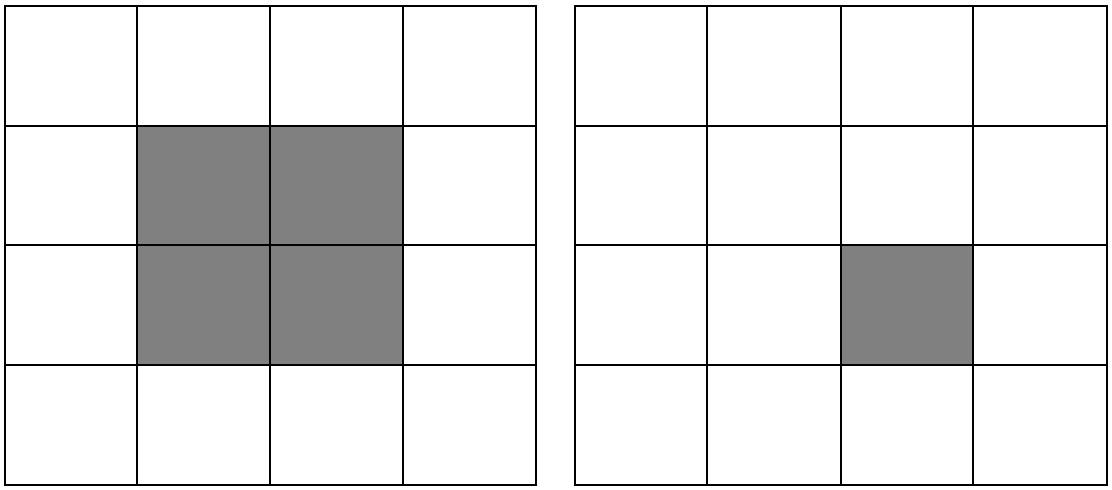


Рис. 6. Схема оценивания статистики и записи данных при размере окна 16×16 и сдвиге на 8 точек

Как и ранее, оценим построенные схемы, используя архиватор RAR. Результаты экспериментов на тех же 10 файлах приведены в табл. 4 (в последней строке указаны средние значения). Во все файлы так же было внедрено по 30 кбайт полностью случайных данных. Мы видим, что средняя разность δ при увеличенном размере окна уменьшилась по сравнению с данными табл. 3, причём лучший результат получен при сдвиге окна на 8 точек. Именно на основе этого метода оценивания статистики и был построен алгоритм и разработана программа, осуществляющая внедрение данных в файлы формата BMP, стойкость которой по отношению к стегоанализу будет рассматриваться в следующих разделах.

Таблица 4. Разность длин сжатых представлений пустых и заполненных файлов

№	δ, кбайт окно 16×16, сдвиг 16	δ, кбайт окно 16×16, сдвиг 8	δ, кбайт окно 8×8, сдвиг 8
1	1.30	0.63	1.45
2	0.59	0.32	0.75
3	0.38	0.34	0.38
4	6.39	5.22	6.42
5	3.26	2.93	3.45
6	2.64	4.33	4.76
7	1.90	1.80	1.98
8	0.79	1.20	1.70
9	4.62	3.88	4.16
10	1.13	0.75	0.91
Ср.	2.30	2.14	2.60

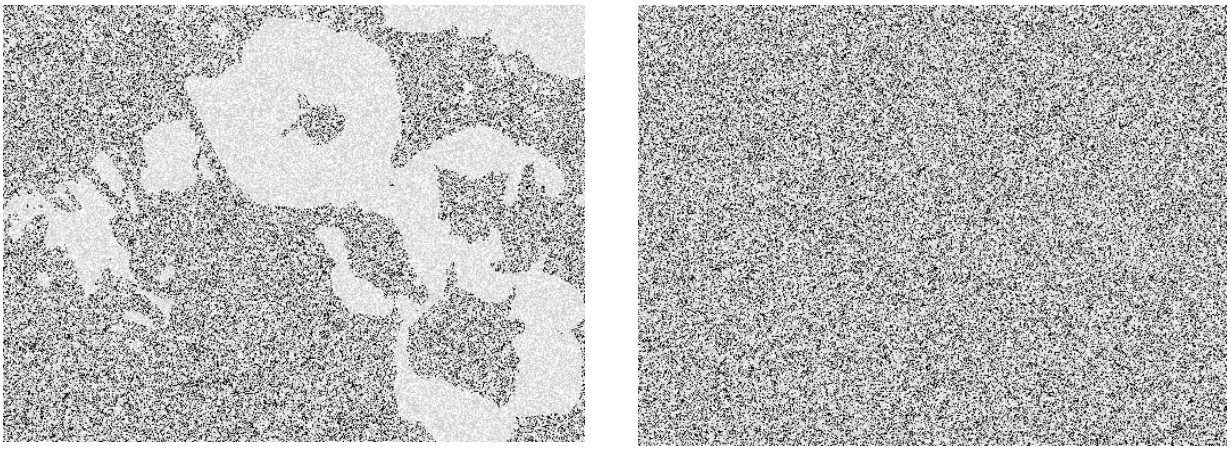
3. Универсальный стегоанализ алгоритма

Универсальный стегоанализ разработанного алгоритма проводился с помощью программной системы, разработанной М. Ю. Жилкиным [6]. Была сформирована выборка из 150 BMP-файлов. Далее эти файлы заполнялись случайными данными (имитирующими зашифрованные сообщения) примерно на 50 % от их ёмкости как разработанной нами программой, так и двумя общедоступными стеганографическими программами HIDE4PGP и STEGOTOOLS. При проведении анализа использовались 4 архиватора: RAR, ZIP, GZIP и BZIP2. Известно, что эти архиваторы построены на базе различных классов методов сжатия данных. Так, только один из них – RAR – применяет сжатие на основе статистической модели источника (алгоритм PPM). Архиваторы ZIP и GZIP базируются на словарных методах (различные варианты кодов Лемпела-Зива), BZIP2 использует преобразование Барроуза-Уилера (BWT).

Прежде чем мы перейдём к изложению полученных результатов, кажется интересным дать визуальное сопоставление работы нашего алгоритма и упомянутых выше программ. Различие в их работе отчётливо видно при внедрении сообщения в изображения класса A. На рис. 7–9 показано распределение изолированных младших бит (с усилением яркости).

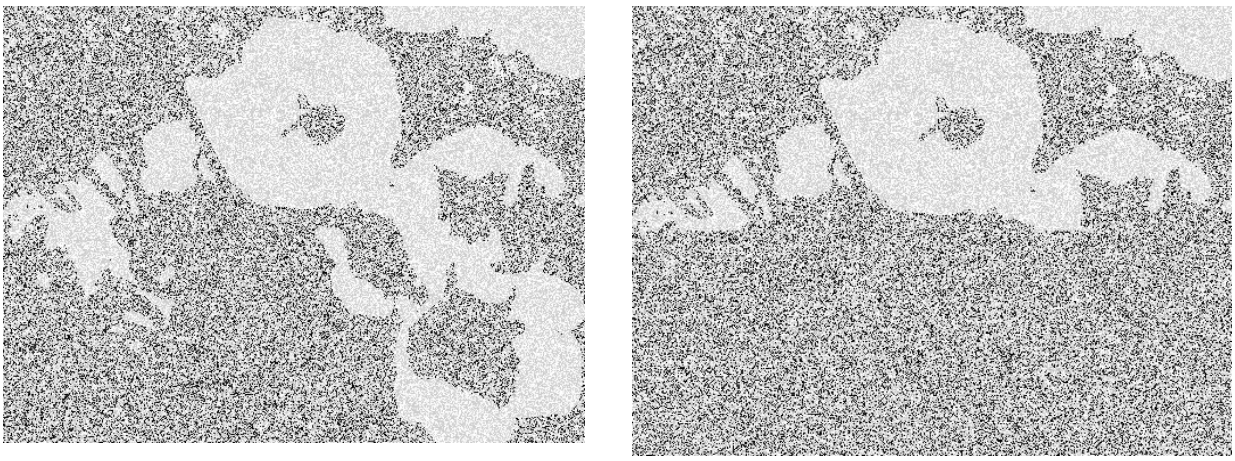
Программа HIDE4PGP (рис. 7) производит так называемое рассеянное заполнение, когда младшие биты замещаются новой информацией в порядке, определяемом некоторой псевдослучайной последовательностью. Мы видим, что даже при 50 % заполнении распределение младших бит полностью искажается и приближается к чисто случайному. Программа STEGOTOOLS (рис. 8) производит последовательное заполнение. Хорошо видно, что половина младших бит изменена (BMP-файл хранится в памяти начиная с нижних строк изображения). Наконец наша программа (рис. 9) в значительной степени сохранила рисунок младших бит несмотря на то, что 50 % младших бит изменено!

Результаты универсального стегоанализа М.Ю. Жилкина приведены в табл. 5. Следует отметить, что в выборку, на которой проводились испытания, были включены в основном изображения класса А. Это объясняется тем, что многие изображения класса В, более подходящие для предложенного алгоритма, изначально признавались "заполненными" (ошибка первого рода). Но и на изображениях класса А мы видим превосходство разработанной программы над аналогами. Особенно заметен результат действия программы по отношению к архиватору RAR (4 % обнаружения). Что касается результатов, показанных на других архиваторах, они, с одной стороны, свидетельствуют о недостаточном качестве используемой в программе статистической модели, а с другой – наводят на мысль о дальнейшем улучшении этой модели за счёт привлечения словарных методов и BWT.



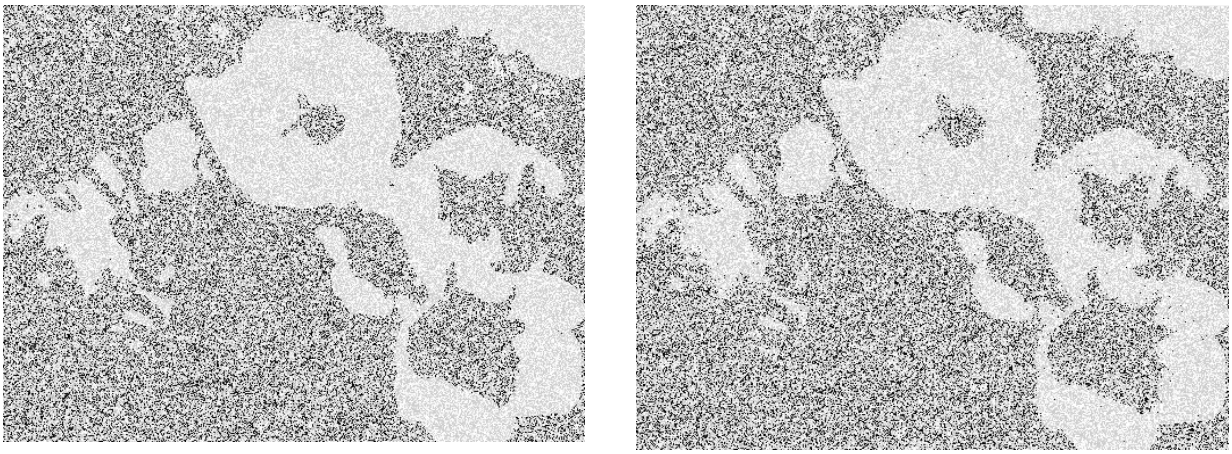
а Исходное изображение *б* Изображение с внедрённым сообщением

Рис. 7. Младшие биты при заполнении на 50 % программой HIDE4PGP



а Исходное изображение *б* Изображение с внедрённым сообщением

Рис. 8. Младшие биты при заполнении на 50 % программой STEGITOOLS



а Исходное изображение *б* Изображение с внедрённым сообщением

Рис. 9. Младшие биты при заполнении на 50 % разработанной программой

Таблица 5. Результаты универсального стегоанализа М.Ю. Жилкина при заполнении контейнеров на 50 %: доля файлов в процентах, в которых была обнаружена встроенная информация

	HIDE4PGP	STEGOTOOLS	Разработанный алгоритм
RAR	100 %	55 %	4 %
ZIP	100 %	100 %	59 %
GZIP	100 %	100 %	59 %
BZIP2	100 %	100 %	85 %

4. Специфический стегоанализ и его предотвращение

Стегоанализ М.Ю. Жилкина относится к классу универсальных методов, т.к. он строится без учёта деталей реализации стеганографических алгоритмов. Если всё-таки исходить из несекретности алгоритма (это стандартное требование), то для предложенной конструкции легко реализуется специфический метод анализа, сводящийся к оценке разности в степени сжатия двух частей изображения. Фактически, данные табл. 3, 4 показывают, что рабочая матрица W с записанной в неё информацией сжимается всегда хуже, чем обучающая матрица E. Это может служить надёжным критерием для обнаружения наличия скрытого сообщения. (Заметим, что при отсутствии скрытой информации также наблюдается различие в степени сжатия частей, как это представлено в табл. 1, но там идёт речь об *абсолютной* разности. Иногда, в зависимости от изображения, хуже сжимается матрица W, иногда – матрица E. Кроме того, сама разность заметно меньше, чем разности в табл. 3, 4). Всё это приводит к необходимости ещё одной доработки алгоритма.

Идея состоит в том, чтобы скрыть от стегоаналитика способ разделения частей изображения. Предлагается следующий метод. С помощью того же шифра, которым шифруется встраиваемое сообщение, мы генерируем секретную псевдослучайную битовую последовательность (точно такая же последовательность в дальнейшем может быть получена законным получателем сообщения, знающим секретный ключ). Обозначим эту последовательность через $z = z_1, z_2, z_3, \ldots$. Биты последовательности z используются для определения типа шахматного порядка. Например, если $z_i = 0$, то мы распределяем пикселы между частями в соответствии с шаблоном 1212 (сравн. с рис. 1); если $z_i = 1$, то используется шаблон 2121. Так как чередование этих шаблонов определяется секретной последовательностью, стегоаналитик не сможет разделить файл на обучающую и рабочую части, как это делал отправитель сообщения. Важно решить, в какой момент при разделении изображения можно менять шахматный порядок. Проблема здесь в том, что при переключении этого порядка возникает ситуация типа 1221, в результате чего в обучающую часть попадают два пиксела, отстоящие друг от друга на две позиции. Это может ухудшить оценку статистики, т.к. такие пикселы в меньшей степени связаны между собой. С точки зрения качества статистики, переключения должны быть как можно реже. С другой стороны, переключений должно быть достаточно много, чтобы было невозможно проверить все их комбинации методом перебора. Если мы сделаем порядка 100 переключений, то перебор будет практически невозможен. Но здесь оказывается более важным другое соображение. При 100 переключениях и типичном размере матрицы изображения в несколько сотен точек размер области с постоянным шахматным порядком будет порядка нескольких килобайт. Такой размер области может позволить произвести в ней независимый анализ частей и определить часть, содержащую сообщение. Поэтому более важно потребовать, чтобы области с постоянным шахматным порядком были небольшими по размеру, не позволяющими производить достоверный статистический анализ.

Мы провели анализ трёх вариантов: когда переключение может происходить после каждой пары, четвёрки или восьмёрки пикселов. На настоящем этапе этого оказалось достаточно. Качество оценки статистики, как и ранее, оценивалось путём сравнения степени сжатия исходного файла и файла с внедрённым сообщением. Результаты представлены в табл. 6. В этой таблице для удобства сравнения воспроизведены данные из табл. 4 для выбранного метода построения статистической модели. Через P обозначена длина области с неизменным порядком разделения пикселов (при отсутствии переключений считаем, что $P = \infty$). Мы видим, что при $P = 2$ происходит ухудшение качества оценки статистики. Однако уже при $P = 8$ ухудшения не наблюдается (на использованной выборке изображений даже произошло улучшение).

Таким образом, описанная специфическая атака на стегосистему предотвращена, причём качество алгоритма внедрения информации не пострадало.

Таблица 6. Разность длин сжатых представлений пустых и заполненных файлов

№	δ, кбайт окно 16×16, сдвиг 8 $P=\infty$	δ, кбайт окно 16×16, сдвиг 8 $P=2$	δ, кбайт окно 16×16, сдвиг 8 $P=8$
1	0.63	1.10	1.20
2	0.32	0.55	0.54
3	0.34	0.93	0.97
4	5.22	5.43	5.50
5	2.93	3.18	2.65
6	4.33	3.43	1.51
7	1.80	1.87	1.88
8	1.20	0.76	0.83
9	3.88	4.10	4.22
10	0.75	0.86	0.82
Ср.	2.14	2.22	2.11

Выражение благодарности

Авторы благодарят М. Ю. Жилкина за предоставленную возможность использования его системы стегоанализа BMP-файлов и помощь в проведении тестирования.

Литература

1. Ryabko B., Monarev V. Using information theory approach to randomness testing // Journal of Statistical Planning and Inference. 2005. V. 133, N. 1. P. 95–110.
2. Ryabko B., Monarev V. Experimental investigation of forecasting methods based on data compression algorithms // Problems of Information Transmission. 2005. V. 41, N. 1. P. 65–69 (in Russian: P. 74–78).
3. Ryabko B., Astola J. Universal codes as a basis for time series testing // Statistical Methodology. 2006. V. 3. P. 375–397.
4. Ryabko B, Astola J. Universal codes as a basis for nonparametric testing of serial independence for time series // Journal of Statistical Planning and Inference. 2006. V. 136, N. 12. P. 4119–4128.
5. Ryabko B. Compression-based methods for nonparametric density estimation, on-line prediction, regression and classification for time series // 2008 IEEE Information Theory Workshop. Porto, Portugal, May 5–9, 2008.
6. Жилкин М. Ю. Стегоанализ графических данных на основе методов сжатия // Вестник СибГУТИ. 2008. № 2. С. 62–66.
7. Cachin C. An information-theoretic model of steganography // Lecture Notes in Computer Science (Proc. 2nd Information Hiding Workshop). Springer Verlag, 1998. V. 1525. P. 306–318.

8. Ryabko B., Ryabko D. Information-theoretic approach to steganographic systems // IEEE International Symposium on Information Theory. Nice, France, 2007. P. 2461–2464.
9. Fionov A., Ryabko B. Simple ideal steganographic systems for containers with known statistics // XI International Symposium on Problems of Redundancy. St.-Petersburg, July 2–6, 2007. P. 184–188.
10. Рябко Б.Я., Фионов А.Н. Алгоритмы кодирования для идеальных стеганографических систем // Вестник НГУ, серия: Информационные технологии. 2008. № 2. С. 88–93.
11. Hoshi M., Han T. S. Interval algorithm for homophonic coding // IEEE Transactions on Information Theory. 2001. V. 47. P. 1021–1031.
12. Рябко Б. Я., Фионов А. Н. Эффективный метод адаптивного арифметического кодирования для источников с большими алфавитами // Проблемы передачи информации. 1999. Т. 35, № 4. С. 1–14.
13. Johnson N. F., Jajodia S. Steganography: seeing the unseen // IEEE Computer. 1998. N 2. P. 26–34.

Статья поступила в редакцию 15.12.2008.

Елтышева Екатерина Юрьевна
аспирант кафедры прикладной математики и кибернетики СибГУТИ,
тел. (383) 2-698-272, e-mail: katya@1nsk.ru.

Фионов Андрей Николаевич
д.т.н., профессор кафедры прикладной математики и кибернетики СибГУТИ,
тел. (383) 2-698-272, e-mail: fionov@sibsutis.ru.

Construction of Stegosystem on the Basis of Raster Images Using Least-Significant Bits Statistics

E. Yu. Eltysheva, A. N. Fionov

A method of data hiding in least-significant bits of raster images, represented as files in BMP, PNG and other formats which employ lossless data compression. In contrast to known analogs, the suggested method encodes embedded data according to least-significant bit probability estimates in order to preserve initial statistics of a container file as much as possible. Universal and specific steganalysis of the algorithm developed is carried out. An advantage of the suggested method over known analogs is demonstrated.

Keywords: steganography, steganalysis, LSB-methods, statistics model, arithmetic coding, ideal stegosystems.