УДК 535.317.1

# Стеганографические методы встраивания и обнаружения сокрытых сообщений, использующие GIF-изображения в качестве файлов-контейнеров

*И.В. Пономарев, Д.И. Строкин*

Алтайский государственный университет (Барнаул, Россия)

# Steganographic Methods for Embedding and Detecting Hidden Messages Using GIF Images as Container Files

*I.V. Ponomarev, D.I. Strokin*

Altai State University (Barnaul, Russia)

За последние несколько десятилетий в связи с повсеместным развитием информационных технологий и средств мультимедиа значительную актуальность приобретает разработка новых методов хранения, передачи, анализа и воспроизведения данных. К числу таких методов также относятся средства обеспечения надежности, защищенности, безопасности и конфиденциальности информации при ее передаче по различным каналам связи. На сегодняшний день выполнение необходимых требований конфиденциальности данных развивается в направлении криптографии и стеганографии.

В статье рассматривается метод обеспечения конфиденциальности данных средствами цифровой стеганографии, использующий в качестве файлов-контейнеров изображения формата GIF. Рассматриваются особенности данного метода, описываются основные шаги алгоритма и проводится анализ полученного файла. Во второй части работы изучается метод, оценивающий вероятность сокрытия информации в блоках файла-контейнера. Этот метод основан на применении критерия согласия Пирсона. Для проверки работоспособности данного метода проводился ряд тестовых испытаний алгоритма. Как результат была создана компьютерная программа на базе среды программирования Microsoft Visual Studio 2019 Community и языка программирования C# (платформа .NET Framework).

***Ключевые слова:*** контейнер, сообщение, палитра, индекс цвета, наименее значащий бит, гистограмма.

Over the past few decades, the development of new methods of storing, transferring, analyzing, and reproducing data has become increasingly important, considering the widespread development of information technologies and multimedia means. These methods also include means of ensuring the reliability, security, safety, and confidentiality of information when it is transmitted through various communication channels. Today, the fulfillment of the necessary data confidentiality requirements is developing in the direction of cryptography and steganography.

The article discusses a method of ensuring data confidentiality by means of Digital Steganography, using GIF images as container files. The features of this method are considered, the main steps of the algorithm are described, and the resulting file is analyzed. The second part of the article studies a method that estimates the probability of hiding information in the blocks of a container file. This method utilizes Pearson's goodness-of-fit test. A number of test tests of the algorithm have been carried out to check the efficiency of this method. As a result, a computer program has been created using the Microsoft Visual Studio 2019 Community programming environment and the C# programming language (.NET Framework).

***Keywords:*** container, message, palette, color index, least significant bit, histogram.

**DOI: 10.14258/izvasu(2022)1-18**

**Введение, постановка задачи.** Основная задача стеганографических методов заключается не только в том, что необходимо произвести сокрытие конфиденциальной информации, но и в сокрытии самого факта существования данной информации при передаче, хранении или обработке [1, 2].

**Определение 1.** Контейнером (носителем) называют несекретные данные, которые используют для сокрытия сообщений. Пустой контей-

нер — контейнер без встроенного сообщения; заполненный контейнер, или стего-контейнер, содержит встроенную информацию [3].

В интернете широко распространены программные реализации различных стеганографических методов сокрытия сообщений, использующие файлы различных форматов: видеофайлы (.MPEG, .AVI), аудиофайлы (.MP3, .WAV), изображения (.BMP, .JPEG, .PNG, .GIF). Например, программа Hide4PGP позволяет встраивать информацию внутрь изображений .BMP и аудиофайлов .WAV, а свободно распространяемая OpenStego поддерживает форматы .BMP, .JPEG, .PNG, .GIF и WBMP, но заполненные контейнеры всегда сохраняет в формате .PNG [4, 5].

Такие программы чаще всего используют самый распространенный метод сокрытия информации — метод замены Наименее Значащего Бита (Least Significant Bit), основной принцип которого заключается в том, что передаваемая информация встраивается в значения младших битов изображения [5]. Модификация именно таких битов не способна восприниматься человеческим зрением, так как они несут в себе меньше всего информации [3, 6].

Рассмотрим пример. Допустим, имеется 8-битное изображение в градациях серого. 00 (00000000) обозначает черный цвет, FF (11111111) — белый. Всего имеется 256 градаций. Также предположим, что сообщение состоит из 1 байта, например, 01101011.

При использовании 2 младших бит в описаниях пикселей нам потребуется 4 пикселя. Допустим, они черного цвета. Тогда пиксели, содержащие скрытое сообщение, будут выглядеть следующим образом: 00000001 00000010 00000010 00000011.

Из достоинств данного метода можно выделить высокую пропускную способность (то есть максимальное количество информации, которое может быть встроено в один элемент контейнера), а из недостатков — низкую стеганографическую стойкость [7, 8].

Также существует ряд программных реализаций стеганографических методов, основанных на особенностях представления информации в той или иной информационной среде. Так, например, в статье [9] рассматривается математическая модель стеганографической системы, использующая в качестве файлов-контейнеров изображения формата BMP и проводящая их последующее сжатие к формату JPEG без потери скрываемой информации.

Поэтому для исследования в качестве носителей информации были выбраны изображения формата .GIF. Данный выбор обусловлен тем, что при передаче на сервер файлы данного формата, как правило, не подвергаются дополнительным модификациям и компрессии в связи с особенностями алгоритма сжатия LZW, являющегося основой выбранного формата. Это, в свою очередь, гарантирует нам дополнительную сохранность конфиденциальной информации.

Реализуемая программа будет производить полную декомпрессию файла-контейнера, последующее внедрение информации, которую необходимо засекретить, и компрессию полученных данных в новый заполненный GIF-файл.

Ниже приведены еще несколько причин выбора GIF-изображений в качестве файлов-контейнеров:
1) большой объем пространства сокрытия, то есть участков, в которых система может скрыть информацию;
2) заранее известный размер контейнера;
3) наличие в большинстве изображений текстурных областей, имеющих шумовую структуру и хорошо подходящих для встраивания информации;
4) задачи защиты фотографий, картин, видео от незаконного тиражирования и распространения;
5) широкая распространенность файлов-изображений GIF в сети интернет.

Отличительными чертами данного формата являются [10]:
1) использование блочной структуры данных;
2) использование палитры цветов — фиксированного набора (диапазона) цветов и оттенков, имеющего физическую или цифровую реализацию в том или ином виде.

При использовании палитры каждая точка изображения содержит лишь номер цвета из палитры, а не информацию о ее цвете в цветовом пространстве.

**1. Разработка метода сокрытия информации.** Разрабатываемый алгоритм базируется на разобранном выше методе замены Наименее Значащего Бита. Однако заметим, что пиксели GIF-изображений — это поток индексов из цветовой палитры. Это значит, что мы не можем напрямую менять значения пикселей, так как если элементы, близкие по индексу, будут иметь совершенно разные представления в цветовом пространстве, изменения младшего бита могут привести к заметным изменениям самого изображения.

Наилучшим решением в подобной ситуации будет использование «подобных» элементов палитры [3]. Под подобными в данном случае понимаются пары элементов, цветовая интенсивность которых отличается на незначительное число $d$. Например, значения яркости для цветов (255, 255, 255) и (255, 254, 253) будут некритично отличаться, а значит, индекс одного элемента можно легко заменить индексом другого.

Предлагаемая процедура включает в себя следующие шаги:

*Шаг 1.* Сортировка палитры цветов по возрастанию веса $W$, где:

$$W = R \cdot 65536 + G \cdot 256 + B.$$

*Шаг 2.* Поиск пар элементов в отсортированной палитре, для которых разность весов $W$ меньше заданной пороговой величины $d$. Обозначим одну такую пару за $(j_i, j_k)$, где $i$ и $k$ — это индексы элементов в неотсортированной палитре, причем в отсортированной таблице $j_i$ от $j_k$ отличается на 1.

*Шаг 3.* Сокрытие сообщения. Последовательно просматриваются все точки изображения, по значению точки k определяется соответствующий номер $j_k$. Если элемент отсортированной палитры $j_k$ пригоден для сокрытия, то значение его наименее значащего бита заменяется очередным битом сообщения. Затем по получившемуся номеру $j_{k'}$ определяется связанный с ним элемент исходной таблицы $k'$, который и присваивается текущей точке.

*Шаг 4.* Извлечение сообщения происходит аналогичным способом. Для текущей точки $k$ ищется номер $j_k$ в отсортированной по весу $W$ палитре цветов, и если:
— младший бит индекса $j_k$ равен нулю, смотрим, удовлетворяет ли пара $(j_k, j_k + 1)$ условию: $W_{j_k+1} - W_{j_k} < d$. Если удовлетворяет, значит, из индекса $j_k$ извлекаем младший бит и записываем его в сообщение;
— младший бит индекса $j_k$ равен единице, смотрим, удовлетворяет ли пара $(j_k - 1, j_k)$ условию: $W_{j_k} - W_{j_k-1} < d$. Если удовлетворяет, значит, из индекса $j_k$ извлекаем младший бит и записываем его в сообщение.

Для проверки данного алгоритма была создана компьютерная программа на базе среды программирования Microsoft Visual Studio 2019 Community и языка программирования C# (платформа .NET Framework). Графический интерфейс программы представлен на рисунке 1.

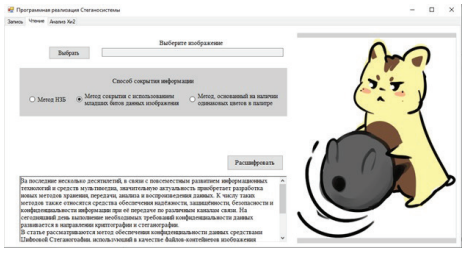


Рис. 1. Результат извлечения скрытого сообщения из некоторого заполненного файла-контейнера

**2. Алгоритм обнаружения факта сокрытия сообщения.** Выявление факта сокрытия информации внутри файла-контейнера — отдельный вид стеганографических атак, часто основывающихся на различных статистических закономерностях контейнеров. Модификацию одной из таких атак мы и рассмотрим.

Гистограммный метод, или метод, основанный на критерии $\chi^2$ [1, 3, 11], предполагает, что вероятность одновременного появления соседних (то есть отличных на наименее значащий бит) цветов в незаполненном контейнере крайне мала. А при последовательном встраивании равномерного сообщения пиксели изображения, напротив, приобретают равномерное распределение.

Поэтому степень различия между вероятностными распределениями элементов естественных контейнеров и полученных из них стего может быть использована для оценки вероятности существования стегоканала. Для удобства оценки вероятности внедрения секретной информации изображение необходимо разбить на отдельные блоки.

Полученный алгоритм имеет следующие этапы:

*Шаг 1.* Сортировка палитры цветов по возрастанию веса $W$, где:

$$W = R \cdot 65536 + G \cdot 256 + B$$

*Шаг 2.* Разбиение изображения на отдельные блоки.

*Шаг 3.* Для текущего блока строится эмпирическая гистограмма по количеству вхождений для каждого элемента палитры, т.е. подсчитывается, сколько раз $(n_i^*)$ элемент палитры $x_i$ принял рассматриваемые значения цветов палитры.

*Шаг 4.* На основе полученной эмпирической гистограммы строится теоретическая путем нахождения среднего арифметического количества пикселей элементов с соседними номерами:

$$n_0 = n_1 = \frac{n_0^* + n_1^*}{2}$$

*Шаг 5.* Величина $\chi^2$ для сравниваемых распределений последовательности и ожидаемого распределения стего равна:

$$\chi^2 = \sum_{i=1}^{v} \frac{(n_i - n_i^*)^2}{n_i^*}.$$

*Шаг 6.* Вычисляется вероятность $p$ того, что два распределения одинаковы [11]:

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int\limits_0^{\chi^2_{k-1}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1}\, dx,$$

где $\Gamma$ — гамма-функция Эйлера, $k$ — количество цветов в палитре.

Эффективность полученного метода была проверена на различных модельных примерах. Например, на рисунке 2 представлен результат оценки одного из тестовых изображений.

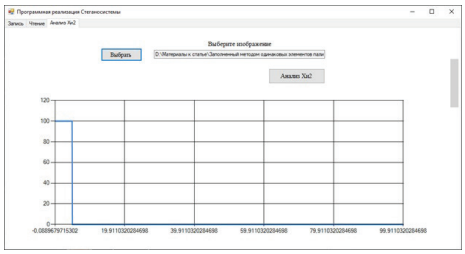

Рис. 2. Результат анализа файла-контейнера на наличие скрытого сообщения

В этом случае метод $\chi^2$ свидетельствует о том, что в 5% контейнера есть скрытая информация.

По результатам анализа алгоритма можно выделить следующие способы сведения вероятности нахождения скрытого сообщения к минимуму:
1. Снижение вероятности обнаружения стегосообщения атаками, основанными на критерии $\chi^2$, можно добиться использованием псевдослучайных интервалов при сокрытии очередного бита сообщения. Иными словами, лучше всего встраивать сообщение не последовательно, а рассеянно по всему контейнеру.
2. Использование $\pm 1$ кодирования при замене Наименее Значащих Битов изображения. Суть такого кодирования заключается в том, что вместо прямого встраивания бита сообщения в Наименее Значащий Бит лучше менять все значение байта, уменьшая или увеличивая его на единицу. Увеличивать текущий байт или уменьшать, определяется псевдослучайным образом.
3. Использовать как можно меньшее количество пар элементов палитры, подходящих для сокрытия информации.
4. Использовать изображения со сложной структурой и зашумленностью.

**Заключение и выводы.** Приведенные в работе методы позволяют решать задачи сохранения конфиденциальности сообщения при передаче и хранении. Основанный на данном методе алгоритм имеет оптимальную сложность для кодирования информации больших объемов. Разработанная компьютерная программа, реализующая указанный подход, имеет приемлемую ресурсоемкость и сложность вычислений — общее время сокрытия сообщений зависит лишь от количества отдельных кадров GIF-изображения.

Также выбранный метод оказывает минимальные визуальные воздействия на исходный файл-контейнер, а в случаях когда в палитре присутствуют элементы с разными индексами, но кодирующие один и тот же цвет, заполненный файл-контейнер становится визуально идентичен оригиналу.

## Библиографический список

1. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. М., 2009.

2. Хорошко В.А., Чекатков А.А. Методы и средства защиты информации. К., 2003.

3. Аргановский А.В., Балакин А.В., Грибунин В.Г., Сапожников С.А. Стеганография, цифровые водяные знаки и стеганоанализ. М., 2009.

4. Барсуков В.С., Романцов А.П. Компьютерная стеганография вчера, сегодня, завтра // Специальная техника. 1998. № 4, 5.

5. Изычева А.В., Сидоренко В.Г. Стеганографические методы защиты информации. М., 2017.

6. Shannon C.E. The communication theory of secrecy systems // Bell Sys. Tech. J. 1949. Vol. 28. № 4.

7. Конахович Г.Ф., Пузыренко А Ю. Компьютерная стеганография. Теория и практика / гл. ред. Ю.А. Шпак. К., 2006.

8. Урбанович П.П. Защита информации методами криптографии, стеганографии и обфускации. Минск, 2016

9. Куркина М.В., Пономарев И.В., Строкин Д.И. Стеганографические методы, устойчивые к JPEG сжатию // Известия Алт. гос. ун-та 2021, № 1 (117), DOI: 10.14258/izvasu(2021)1-17.

10. Сэломон Д. Сжатие данных, изображений и звука. М., 2004.

11. Westfeld A., Pfitzmann A. Attacks on Steganographic Systems //Lecture Notes in Computer Science. 2000, Vol. 1768. DOI: 10.1007/10719724_5.