Метод стеганографического преобразования сообщения со свойством частичной неизвлекаемости*

И. В. Нечта
Сибирский государственный университет телекоммуникаций и информатики, Новосибирск, Россия
Контактный e-mail: ivannechta@gmail.com

Разработан новый метод стеганографического преобразования двоичного сообщения, который позволяет внедрять в него скрытые данные. Для минимизации искажений статистических свойств контейнера, вносимых при внедрении, предложено преобразовывать сообщение, взятое из пустого контейнера. Алгоритм используется непосредственно перед применением стеганографического метода внедрения. Преобразование позволяет противодействовать методам стегоанализа, базирующимся на изучении статистики извлеченных из контейнеров сообщений. При использовании указанного алгоритма стегоаналитик не может корректно извлечь до 32 % бит сообщения из контейнера, и соответственно, реализовать эффективный стегоанализ. Объем внедрения скрытых данных составляет 19 % от длины исходного сообщения. В работе рассмотрены различные алгоритмы стегоанализа текстовых данных, показана их низкая эффективность после применения нового метода.

Ключевые слова: стеганография, стегоанализ, передача скрытых данных, неизвлекаемость.

Библиографическая ссылка: Нечта И.В. Метод стеганографического преобразования сообщения со свойством частичной неизвлекаемости // Вычислительные технологии. 2019. Т. 24, № 3. С. 75–87. DOI: 10.25743/ICT.2019.24.3.006.

Введение

В сетях связи вследствие увеличения объема передаваемой информации возрастает доля данных, к которым предъявляются повышенные требования по информационной безопасности. В одном случае необходимо ограничить доступ к данным от третьих лиц с помощью некоторого секретного ключа, например при передаче конфиденциальных документов по открытому каналу связи. В других ситуациях, в частности в задачах идентификации лицензионных копий цифровых объектов, требуется скрыть сам факт передачи секретных данных. Для создания тайного канала связи применяют методы стеганографии, которые позволяют встраивать скрытое сообщение в цифровой объект данных — так называемый контейнер. Сам факт передачи по сети такого объекта не является чем-то подозрительным. В качестве контейнера могут выступать файлы изображений, видео- и аудиофайлы, текстовые и исполняемые файлы.

*Title translation and abstract in English can be found on page 86.
© ИВТ СО РАН, 2019.

Рассмотрим метод встраивания скрытых сообщений в изображения LSB±1, описанный в работе [1]. Согласно этому алгоритму в изображение, которое является матрицей пикселов, сообщение записывается в последний значащий бит цвета. Если последний значащий бит уже равен биту скрываемого сообщения, то изменение цвета не производится, в противоположном случае значение цвета пикселя уменьшается или увеличивается на единицу в случайном порядке.

Метод встраивания скрытых сообщений в тексты на английском языке [2] производит замену слов в предложении на эквивалентные им по смыслу слова (синонимы) в соответствии со скрываемым сообщением. Смысл текста после преобразования не меняется, однако в ряде случаев возможны нарушения идиоматических выражений или статистической взаимосвязи слов с контекстом предложения.

Метод стеганографии исполняемых файлов [3] встраивает скрытое сообщение в неиспользуемые места секций файлов формата Portable Executable. Размер кода и данных в таких файлах выравнивается до размера, кратного 512 байтам. Выравнивание производится с помощью записи нулевых байтов, которые могут быть заменены на байты секретного сообщения. Более подробно методы стеганографии описаны в обзорах [4, 5].

Обратной к стеганографии задачей является стегоанализ, цели которого — выявление факта наличия секретного сообщения в контейнере и оценка объема внедрения. Стегоаналитик собирает некоторую статистику по заранее отобранным пустым и заполненным контейнерам, затем по анализируемому (подозрительному) контейнеру собирает аналогичную статистику и устанавливает, к какой категории относится анализируемый контейнер (к пустым или заполненным).

Эффективность стегоанализа напрямую определяется тем, какую статистику собирает стегоаналитик. Для оценки эффективности стегоанализа применяются два подхода. Первый предполагает сбор и анализ статистической взаимосвязи элементов контейнера, например связи цветов соседних пикселей изображения или соответствие слов (имеющих синонимы) контексту предложения, в котором они находятся. Второй подход используется, когда существует возможность непосредственно извлечь и проанализировать сообщение из контейнера (в том числе из пустого). Извлеченное из пустых текстовых контейнеров сообщение, как правило, выглядит как низкоэнтропийная последовательность, в отличие от сообщения из заполненного контейнера. Указанный способ использован в работе [6]. Метод стегоанализа текстовых файлов [7] обнаруживает факт применения кодирования длин серий синонимов [8] с вероятностью не менее 98.5 %. Из подозрительного контейнера извлекается сообщение, затем анализируется распределение вероятностей появления серий (последовательностей одинаковых битов) заданной длины, затем производится сравнение собранной статистики с эталонной, соответствующей пустому и заполненному контейнерам.

Внедрение в исполняемый файл может быть обнаружено с помощью алгоритма, описаного в [9]. В этом случае для выявления участков с внедрением используется сжатие фрагментов исполняемого файла. Как правило, секретное сообщение шифруют перед внедрением. Одним из требований к шифру является статистическая неразличимость зашифрованной последовательности и “истинно случайной”, у которой вероятности битов “0” и “1” одинаковы и между их появлением отсутствуют какие-либо закономерности. Соответственно, зашифрованная последовательность не может быть сжата архиватором. Код программы, напротив, содержит часто повторяющиеся инструкции, что приводит к его заметному сжатию. Таким образом, сжимая фрагменты исполняемого файла, определяют наличие факта внедрения.

Большое внимание в методах стеганографии уделяется скрытности внедрения. Скрытность может быть определена через вероятность корректного обнаружения наличия или отсутствия факта внедрения. Так, в работе [7] используются вероятности ошибок первого рода: случай, когда заполненный контейнер признается пустым, и второго рода, когда пустой контейнер признается заполненным. Считается, что чем больше ошибок при стегоанализе, тем выше скрытность метода внедрения.

Скрытность можно увеличить двумя путями. Первый подход предполагает уменьшение объема внедрения, как, например, в алгоритмах, рассмотренных в статье [10]. Каждое изменение исходного контейнера нарушает его "естественную" статистическую структуру, что потенциально может быть выявлено в ходе стегоанализа. Второй подход [9] предполагает предварительное кодирование внедряемого сообщения так, чтобы уменьшить статистическое различие между сообщениями, взятыми из пустого и заполненного контейнеров. Стоит отметить, что кодирование, добавляющее избыточность, также уменьшает полезный объем внедрения в контейнер.

В настоящей работе предложен новый метод преобразования сообщения, взятого из пустого контейнера. Целью преобразования является увеличение скрытности внедрения секретного сообщения за счет ограничения возможности извлечь сообщение из контейнера. Без знания секретного параметра стегоаналитик не сможет корректно извлечь сообщение и, соответственно, произвести анализ его статистики. Такое преобразование значительно повышает устойчивость к анализу рядом известных методов, базирующихся на извлечении и анализе статистики сообщения в контейнере.

1. Метод преобразования сообщения

В настоящей работе описан алгоритм преобразования сообщения, который применяется перед непосредственным внедрением зашифрованного сообщения в контейнер. Разработка алгоритма направлена на уменьшение вероятности успешного стегоанализа методами, которые базируются на выявлении статистических различий сообщений, взятых из пустого и заполненного контейнеров [6, 7, 11]. Основное преимущество предлагаемого метода состоит в том, что атакующий стегоаналитик не может корректно извлечь сообщение из контейнера без знания некоторого секретного параметра, доступного только отправителю и адресату сообщения. Доля некорректно извлекаемых битов сообщения из контейнера, согласно экспериментальным исследованиям, составляет 32 %, что делает неэффективным практически любой стегоанализ рассматриваемого типа.

Остановимся более подробно на методах стегоанализа, основанных на выявлении статистических различий сообщения, взятого из пустого и заполненного контейнеров [6]. Алгоритм стегоанализа извлекает сообщение из контейнера и анализирует его на предмет отклонения от истинно случайной последовательности (в которой биты появляются равновероятно и независимо друг от друга). Зашифрованное сообщение, внедряемое в контейнер, выглядит как случайное (это одно из требований к современным шифрам), а сообщение из пустого контейнера имеет разного рода отклонения от случайности. Если отклонение не будет выявлено, то имеет место факт внедрения скрытого сообщения.

Для стегоанализа применяется "тест со смещением" [6]. Тест предполагает последовательное разбиение исходного сообщения на четырехбитные слова и проверку равновероятности их появления. На каждом последующем шаге исходное сообщение уменьша-

ется на один начальный бит. Разбиение и проверка выполняются пять раз. Если хотя бы один раз выявлено отклонение от случайности, то все сообщение признается неслучайным, соответственно, анализируемый контейнер пуст. В противном случае контейнер заполнен.

Стегоанализ может быть основан на различии в распределениях длин серий. Под серией понимается последовательность повторяющихся битов. Автором работы [7] экспериментально показано, что после внедрения методом кодирования длин серий синонимов [8] вероятность появления серий длиной 1 бит значительно уменьшается.

Известны методы стегоанализа исполняемых файлов, базирующиеся на сжатии данных, например [11]. В этом случае код программы, имеющий множество повторяющихся инструкций, сжимается более чем в два раза. Зашифрованное сообщение абсолютно несжимаемо, что позволяет его эффективно выявлять в ходе стегоанализа.

В рамках настоящего исследования для уменьшения статистических различий после внедрения предлагается использовать алгоритм SMT, который берет из пустого контейнера исходное сообщение и преобразует, добавляя в него скрытую информацию. Алгоритм SMT формально может быть записан следующим образом:

$$SMT(M, S, salt) = M'. \tag{1}$$

Здесь M — сообщение, взятое из пустого контейнера; S — внедряемое секретное сообщение; $salt$ — секретный инициализационный параметр для генератора случайных чисел; M' — преобразованное сообщение (с внедрением).

Рассмотрим сообщение M как последовательность двухбитных элементов (далее просто элементов). Назовем серией последовательность неповторяющихся элементов. Длину серии (количество элементов в серии) обозначим через Len. Пример: в исходном сообщении подчеркнуты серии, а жирным шрифтом выделены элементы после серий — так называемые терминальные элементы, не входящие в серию:

$$\underline{01\ 10\ 11}\ \mathbf{10}\ \underline{10}\ \mathbf{10}\ \underline{00\ 10}\ \ \mathbf{00}\ \underline{10\ 01\ 00\ 11}\ \ \mathbf{01}\ \ldots$$

Серии разделены между собой терминальными элементами. Внедрение скрытого сообщения осуществляется путем перестановки элементов в сериях. Номер перестановки в двоичном виде является скрытым сообщением.

2. Алгоритм *SMT*

Предлагаемый алгоритм прямого преобразования сообщения включает следующие шаги:

Шаг 1. Поиск серий.

Шаг 2. Выбор допустимых перестановок.

Шаг 3. Нумерация перестановок и внедрение секретного сообщения.

На первом шаге в исходном сообщении M производится поиск серий с применением следующего алгоритма.

Алгоритм “Поиск серий”

Вход: M — сообщение.
Выход: $Result$ — можество пар: {номер начального элемента серии в M, длина серии}.

$M_{Size} \leftarrow$ кол-во элементов в M
$Result \leftarrow \{\emptyset\}$, $i \leftarrow 1$
пока $i < M_{Size}$ **цикл**
    $i_{Start} \leftarrow i$
    $Len \leftarrow 1$
    пока $(i < M_{Size} - 1)$ и $(M[i] = M[i+1])$ **цикл**
        $i \leftarrow i + 1$
        $Len \leftarrow Len + 1$
    к-цикл
    $Result \leftarrow Result \cup \{i_{Start}, Len\}$
    $i \leftarrow i + 1$
к-цикл
возврат $Result$

Внедрение осуществляется через перестановку элементов внутри серии. Номер перестановки в двоичном представлении является частью секретного сообщения S, которое записывается последовательно в каждую серию. Строго говоря, существуют серии длины 1, 2, 3 и 4 элемента. Применив формулу Хартли

$$I = \log_2 N,$$

где I — длина сообщения, биты; N — количество различных вариантов сообщений (в нашем случае число возможных перестановок элементов в серии), получим соотношения длины внедренного сообщения и длины серии (табл. 1). Здесь было произведено округление длины сообщения до целого значения в меньшую сторону, так как дробные биты не могут быть записаны в сообщение.

Второй шаг — отбор допустимых перестановок, которые могут быть использованы для внедрения сообщения по следующему алгоритму. Если серия имеет длину два, то этот шаг алгоритма пропускается.

Алгоритм "Выбор допустимых перестановок"

Вход: P — эталонное распределение вероятностей, $FULL$ — множество всех перестановок в текущей серии.
Выход: *Allowable* — множество допустимых перестановок.
$Q[1] \leftarrow 0$
для i от 2 до $|P|$ **цикл**
    $Q[i] \leftarrow Q[i] + P[i-1]$ //Рассчитаем кумулятивные вероятности
к-цикл

Т а б л и ц а 1. Соотношения длин серии и внедряемого секретного сообщения

Длина серии, биты	Количество элементов	Количество перестановок	Длина внедряемого сообщения, биты
2	1	1	0
4	2	2	1
6	3	6	2
8	4	24	4

$i \leftarrow 0$
пока $i < 2^{\lfloor log_2 |FULL_i| \rfloor}$ **цикл**
    $Rand \leftarrow$ случайное число в диапазоне $(0; 1)$
// Найдем номер интервала, в который попало число $Rand$
    $r \leftarrow |Q|$
    для j от 1 до $|Q| - 1$ **цикл_1**
        если $(Q[j] \leqslant Rand)$и$(Rand < Q[j+1])$ **то**
            $r \leftarrow j$, **выход-цикл_1**
        к-если
        $j \leftarrow j + 1$
    к-цикл_1
// Выбранный интервал определяет перестановку, которая признается допустимой
    если $FULL[r] \notin Allowable$ **то**
        $Allowable \leftarrow Allowable \cup FULL[r]$
        $i \leftarrow i + 1$
    к-если
к-цикл
возврат $Allowable$

Процесс построения множества $Allowable_i$ производится с учетом эталонного распределения вероятностей перестановок элементов в сообщении (обозначенном как P), которое извлечено из типичного пустого контейнера. Для получения типичного пустого контейнера (для примера взят текстовый контейнер) использованы тексты архива Gutenberg Project [12]. Сообщение извлекалось с помощью программы Tyrannosaurus Lex [2]. Размер извлеченного сообщения составил 1.79 Мбит. Далее производился статистический анализ серий сообщения. В итоге получены распределения, указанные в табл. 2.

Рассмотрим процесс выбора допустимых перестановок на примере трехэлементной серии, представленный на рис. 1. Пусть имеется серия i из трех элементов $\{a, b, c\}$. Полным набором будем называть множество всех перестановок в текущей серии

$$FULL_i = \Big[\{a,b,c\}; \{a,c,b\}; \{b,a,c\}; \{b,c,a\}, \{c,a,b\}, \{c,b,a\}\Big].$$

Множество допустимых перестановок для текущей серии обозначим $Allowable_i$. Причем $Allowable_i \subsetneq FULL_i$ и

$$|Allowable_i| = 2^{\lfloor log_2 |FULL_i| \rfloor}. \tag{2}$$

Возьмем единичный отрезок, на котором последовательно отложены интервалы, соответствующие перестановкам множества $FULL_i$. В общем случае размер интервалов

Таблица 2. Эталонные распределения вероятностей пустого текстового контейнера

Длина серии, элементы	Распределения вероятностей, %
3	{16.93, 16.68, 16.71, 16.54, 16.32, 16.82}
4	{4.565, 4.384, 4.371, 4.01, 3.91, 4.09, 4.15, 3.96, 3.94, 4, 3.89, 4.77, 4.33, 3.92, 4.22, 3.91, 4.17, 3.8, 4.04, 4.28, 4.06, 4.44, 4.59, 4.2}

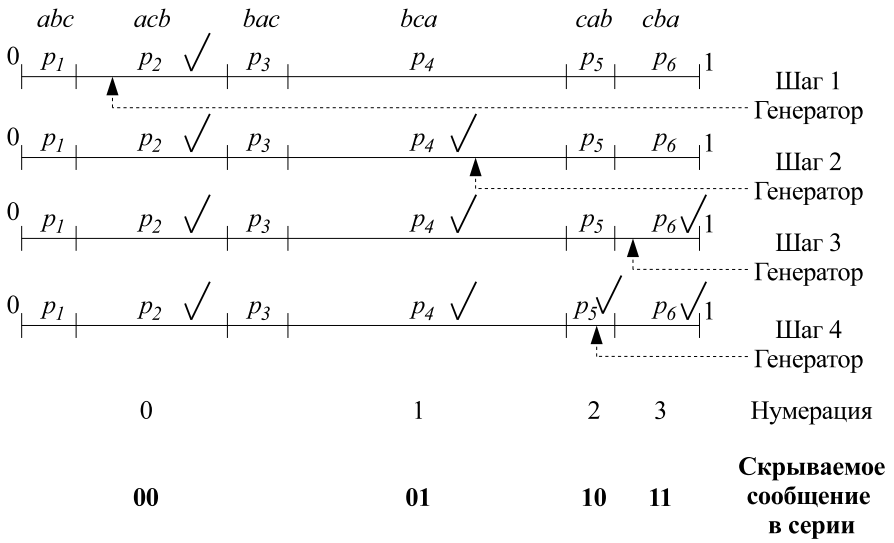

Рис. 1. Процесс выбора допустимых перестановок для трехэлементных серий

равен вероятностям, представленным в табл. 2. Здесь для наглядности взято произвольное распределение $P = \{p_1, \ldots, p_6\}$.

Изначально $Allowable_i = \varnothing$. Генератор случайных чисел, инициализированный параметром $salt$, порождает действительные числа в интервале $(0; 1)$. Сгенерированное число попадает в интервал, соответствующий перестановке из $FULL_i$, которая признается "допустимой" и добавляется во множество $Allowable_i$. Далее генерация чисел повторяется до тех пор, пока не будет достигнуто необходимое для внедрения количество перестановок (см. формулу (2)). Если число повторно попадает в тот же интервал ранее признанной "допустимой" комбинации, то число заново генерируется. Отбор четырех перестановок (в сериях, состоящих из элементов a, b и c) условно показан на рис. 1.

На первом шаге число из генератора попадает в интервал, соответствующий перестановке $\{a, c, b\}$, которая признается допустимой (отмечена галочкой). Далее процесс повторяется. На четвертом шаге получаются следующие допустимые перестановки:

$$Allowable_i = \Big[\{a, c, b\}; \{b, c, a\}; \{c, a, b\}; \{c, b, a\}\Big].$$

На третьем шаге работы алгоритма SMT нумерация допустимых перестановок происходит следующим образом.

Алгоритм "Нумерация перестановки"

Вход: $Allowable$ — множество допустимых перестановок, $X \in Allowable$ — нумеруемая перестановка.
Выход: $Numb$ — номер перестановки.
$i \leftarrow 1$
пока $i \leq |Allowable|$ **цикл**
  если $Allowable[i] = X$ **то**
    $Numb = i$, **возврат** $Numb$
  к-если
к-цикл

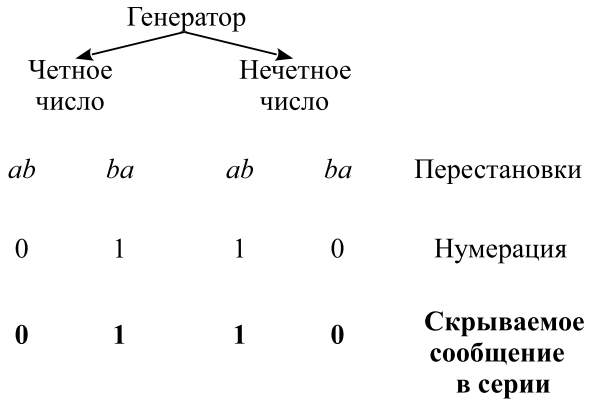

Рис. 2. Нумерация перестановок для двухэлементных серий

Для трех- и четырехэлементных серий перестановка осуществляется последовательно (от нуля) в порядке их расположения на единичном отрезке. Для нумерации двухэлементных серий генерируется натуральное число. Как показано на рис. 2, если получено четное число, то перестановка элементов, отсортированная по возрастанию, нумеруется нулем, по убыванию, — единицей. При генерации нечетного числа нумерация идет в противоположном порядке.

После нумерации секретное сообщение S встраивается последовательно по частям в каждую серию путем выбора нужной перестановки. В результате получается сообщение M' той же длины, что и M. На этом этапе работа алгоритма SMT завершена. Преобразованное сообщение M' записывается в контейнер заранее выбранным способом.

При извлечении адресатом сообщения из заполненного контейнера необходимо провести обратное преобразование:

$$SMT^{-1}(M', salt) = M.$$

Для этого выполняются шаги 1 и 2 алгоритма SMT: поиск серий, выбор допустимых перестановок и их нумерация. По имеющейся в сообщении перестановке определяются ее номер и передаваемое сообщение S. На этом этапе обратное преобразование завершено. Далее адресат расшифровывает и читает секретное сообщение в соответствии с применяемым стеганографическим алгоритмом.

3. Экспериментальное исследование свойств алгоритма SMT

Подход стегоанализа, базирующийся на статистических различиях сообщений, извлеченных из пустого и заполненного контейнеров, предложен и развит в работах [6, 7, 11]. Разработанный алгоритм SMT позволяет противодействовать методам стегоанализа, построенным в рамках указанного подхода. Покажем эффективность разработанного метода экспериментально. При анализе эффективности будем сравнивать статистические различия между M и M'. В качестве статистики применяются распределения вероятностей n-битных слов, статистика “теста со смещением” [6] и распределения вероятностей перестановок в сериях заданной длины. Совпадение статистики, в частности распределений вероятностей, будем определять при помощи теста хи-квадрат.

Для экспериментов подготовлен текстовый контейнер, состоящий из различных текстов архива [12]. Из контейнера при помощи программы [2] извлекалось сообщение (1.79 Мбайт) и в дальнейшем обрабатывалось. Здесь преднамеренно взят один большой контейнер, а не множество контейнеров малого размера. Это связано с тем, что точность применяемых методов стегоанализа возрастает при увеличении объема анализируемого контейнера. Следовательно, анализ большого числа малых по объему контейнеров априори даст низкую точность стегоанализа, что ставит алгоритм SMT в заведомо выигрышную позицию.

Внедряемое сообщение имитировалось с помощью генератора случайных чисел, так как зашифрованное сообщение неотличимо от случайной последовательности. В качестве генератора случайных чисел использовалась функция ANSI C rand(). Далее производилось встраивание сообщения при помощи алгоритма SMT. Количество серий длины 2, 3 и 4 составило 84627, 72882 и 23463 соответственно.

В ходе анализа получены распределения n-битных слов для пустого и заполненного контейнеров (табл. 3). В качестве критерия совпадения вероятностных распределений использовался критерий согласия Пирсона (при уровне значимости $\alpha = 0.05$).

Результаты анализа показывают, что после применения алгоритма STM в распределениях n-битных слов сообщения для пустого и заполненного контейнеров статистически значимых различий не наблюдается. Нулевые значения χ^2 для $n = 1$ и $n = 2$ объясняются тем, что сообщение полностью не перезаписывалось, а только переставлялись его двухбитные элементы, следовательно, их соотношение не менялось.

Далее рассмотрим результаты "теста со смещением" (табл. 4). Здесь под ошибкой первого рода понимается случай, когда заполненный контейнер признается пустым, и ошибкой второго рода — случай, когда пустой контейнер воспринимается как заполненный. Видно, что тест со смещением изначально имеет малую долю ложных срабатываний, но после применения алгоритма SMT тест ошибается практически во всех случаях, что показывает высокую эффективность разработанного метода.

Отметим, что последовательности повторяющихся битов (цепочек), которые часто встречаются в сообщениях пустых текстовых контейнеров, остаются неизменными (так как алгоритм переставляет биты вне цепочек). Следовательно, предложенный в работе [7] стегоанализ, базирующийся на анализе распределения вероятностей длин упомянутых цепочек, будет неэффективным.

Т а б л и ц а 3. Результаты анализа совпадения распределений n-битных слов сообщения

n, биты	χ^2	Значение квантиля	Вывод
1	0	3.84	Отклонения не выявлены
2	0	7.81	« «
3	0.05	14.07	« «
4	0.27	25.00	« «
5	0.17	44.98	« «

Т а б л и ц а 4. Результаты работы теста со смещением

Тип ошибки	Вероятность ошибки, %
Ошибка первого рода	93.0
Ошибка второго рода	7.5

Т а б л и ц а 5. Результаты анализа совпадения распределений перестановок в сериях заданной длины

Длина серии, элементы	χ^2	Значение квантиля	Вывод
2	0.183	3.84	Отклонения не выявлены
3	2.850	11.07	« «
4	2.907	35.17	« «

Рассмотрим результаты анализа распределения вероятностей перестановок в сериях заданной длины. Как видно из табл. 5, до и после внедрения распределения вероятностей перестановок в сообщении существенно не меняются.

4. Особенности алгоритма *SMT*

Поясним некоторые особенности алгоритма. Так, на втором шаге работы алгоритма SMT производится выбор допустимых перестановок. Это необходимо, поскольку при внедрении произвольного сообщения не все перестановки будут использоваться. Например, для четырехэлементной серии возможны $P_4 = 24$ перестановки. Однако в серию такой длины, согласно табл. 1, можно внедрить только 4 бита, т. е. могут быть использованы только 16 вариантов перестановок. То, что часть перестановок не будет использована, создает угрозу эффективного стегоанализа. Указанная проблема встречается только в сериях длины 3 и 4 элемента.

Для ее устранения необходимо "использовать" все варианты перестановок, какие возможны в пустом контейнере. Для этого согласно алгоритму SMT для каждой серии отбирается множество допустимых перестановок (т. е. перестановки, которые могут использоваться в текущей серии) из полного набора (множества различных перестановок в данной серии). Выбор допустимых перестановок производится только для серий длины 3 и 4 элемента. Для серий длины 2 элемента все серии являются допустимыми, соответственно, шаг 2 алгоритма SMT пропускается.

При внедрении есть стремление минимизировать статистические искажения, вносимые при встраивании секретного сообщения. Так, алгоритм позволяет учитывать распределение вероятностей появления перестановок в сообщении типичного пустого контейнера, представленного в табл. 2. В частности, на втором шаге алгоритма SMT генератор случайных чисел, получающий равномерно распределенные числа из интервала $[0; 1]$, будет чаще генерировать числа, соответствующие тем перестановкам, которые имеют бóльшую вероятность. Такой подход, согласно экспериментальным результатам, представленным в табл. 5, не искажает распределение вероятностей серий после внедрения секретного сообщения.

Рассмотрим некоторые дополнительные свойства, которыми обладает предлагаемый алгоритм. При извлечении сообщения из контейнера без знания инициализационного параметра *salt* некорректно извлекаются 32 % бит сообщения. Это связано с тем, что стегоаналитик не может воспроизвести работу генератора случайных чисел и, соответственно, повторить выбор тех же "допустимых" перестановок. В ходе экспериментов обнаружено, что если не использовать генератор случайных чисел для нумерации двухэлементных перестановок, то доля некорректно извлеченных битов падает до 18 %.

Это свойство — невозможность корректно извлечь сообщение из контейнера — было названо неизвлекаемостью.

Стоит отметить, что в известных ранее методах стеганографии с контейнерами-изображениями указанное свойство имело место, так как внедрение секретного сообщения осуществлялось в случайно выбираемые пиксели и не могло быть воспроизведено стегоаналитиком. Однако в иных типах контейнеров (текст, исполняемые файлы и т. д.) указанное свойство ранее не встречалось. В идеальном случае доля неправильно извлеченных битов составит 50 %. Так как указанный предел еще не достигнут, будем считать, что разработанный алгоритм SMT обладает свойством "частичной неизвлекаемости" сообщения. Дальнейшие исследования будут направлены на достижение указанного предела.

Стеганографическое преобразование STM позволяет встраивать скрытые данные объемом 19 % от длины сообщения, извлеченного из пустого контейнера. Для сравнения известный алгоритм внедрения в текстовый контейнер, предложенный в работе [8], имеет аналогичный показатель — 30 %, что является его достоинством. Однако этот алгоритм эффективно обнаруживается стегоанализом, описанным в работе [7]. Таким образом, разработанный алгоритм SMT может активно применяться на практике.

Заключение

Предложен метод стеганографического преобразования сообщения, который позволяет встроить в сообщение скрытые данные при минимальном искажении статистических свойств исходного сообщения. Экспериментально показано, что разработанный алгоритм позволяет противодействовать ряду известных методов стегоанализа. Без знания некоторого секретного параметра стегоаналитик не может корректно извлечь до 32 % бит сообщения из контейнера, что приводит к значительным ошибкам при стегоанализе.

Представленный в настоящей работе алгоритм рекомендуется применять для внедрения в аудио-, текстовые контейнеры, исполняемые файлы и контейнеры-графы [13]. Применение алгоритма SMT для изображений не предусмотрено, так как биты сообщения привязаны к окружению соседних пикселей. Любая перестановка с высокой вероятностью нарушает эту взаимосвязь.

Список литературы / References

[1] **Zhang, W., Zhang, X., Wang, S.** A double layered "plus-minus one" data embedding scheme // IEEE Signal Proc. Lett. 2007. Vol. 14, No. 11. P. 848–851.

[2] **Winstein, K.** Lexical steganography. Available at: `http://web.mit.edu/keithw/tlex/` (accessed 14.01.2018).

[3] US Patent 7664967B2 (2010). Development system with methodology providing information hiding in executable programs / Thorpe, D. 17 p. Available at: `https://patents.google.com/patent/US7664967B2/en` (accessed 16.04.2019).

[4] **Koluguri, A., Gouse, S., Reddy, P. B.** Text steganography methods and its tools // Intern. J. of Advanced Sci. and Techn. Res. 2014. Vol. 2, No. 4. P. 888–902.

[5] **Subhedar, M.S., Mankar, V.H.** Current status and key issues in image steganography: A survey // Comput. Sci. Rev. 2014. Vol. 13. P. 95–113.

[6] **Нечта И.В.** Применение статистического анализа для обнаружения скрытых сообщений в текстовых данных // Вестн. СибГУТИ. 2012. № 1. С. 23–29.
Nechta, I.V. Applying statistical analysis for secret message detection in text data // Vestn. SibGUTI. 2012. No. 1. P. 23–29. (In Russ.)

[7] **Нечта И.В.** Новый метод стегоанализа текстовых данных, полученных кодированием длин серий синонимов // Безопасность информ. технологий. 2018. Т. 25, № 2. С. 114–120.
Nechta, I.V. New method of steganalysis for text data obtained by synonym run-length encoding // Bezopasnost' Inform. Tekhnologiy. 2018. Vol. 25, No. 2. P. 114–120. (In Russ.)

[8] **Xiang, L., Wang, X., Yang, C., Liu, P.** A novel linguistic steganography based on synonym run-length encoding // IEICE Transact. on Inform. and Syst. 2017. Vol. 100, No. 2. P. 313–322.

[9] **Нечта И.В.** Метод внедрения скрытых сообщений в исполняемые файлы // Вестн. СибГУТИ. 2011. № 2. С. 3–10.
Nechta, I.V. Method of secret messages embedding into executable files // Vestn. SibGUTI. 2011. No. 2. P. 3–10. (In Russ.)

[10] **Luo, X., Song, X., Li, X. et al.** Steganalysis of HUGO steganography based on parameter recognition of syndrome-trellis-codes // Multimedia Tools and Appl. 2016. Vol. 75, No. 21. P. 13557–13583.

[11] **Нечта И.В.** Эффективный метод стегоанализа, базирующийся на коде Хаффмана // Вестн. СибГУТИ 2010. № 4. С. 47–53.
Nechta, I.V. Effective method of steganalysis of executable files based on Huffman code // Vestn. SibGUTI. 2010. No. 4. P. 47–53. (In Russ.)

[12] Official website "Gutenberg Project". Available at: `http://www.gutenberg.org/wiki/Main_Page` (accessed 11.06.2018).

[13] **Нечта И.В.** Метод сокрытия информации в графоподобных структурах социальной сети // Вычисл. технологии. 2018. Т. 23, № 2. С. 55–62.
Nechta, I.V. A method of hidden messages embedding in graphlike structures of a social network // Comput. Technologies. 2018. Vol. 23, No. 2. P. 55–62.

Поступила в редакцию 2 июля 2018 г.,
с доработки — 6 ноября 2018 г.

A method of steganographic message transformation with the partial antidisturbance property

Nechta, Ivan V.

Siberian State University of Telecommunication and Information sciences, Novosibirsk, 630102, Russia,

Corresponding author: Nechta, Ivan V., e-mail: `ivannechta@gmail.com`

Purpose. This article addresses the development of a new method for embedding hidden messages in various containers (text, executable files).

Methods. During this research the methods of information theory, probability theory and mathematical statistics are used. Efficiency analysis of the new method was carried out by using the Chi-square test.

© ICT SB RAS, 2019

Results. A new method of steganographic transformation of a binary message, which allows to embed hidden data is presented. It is proposed to transform a message taken from an empty container. The original message is considered as a set of two-bit elements. A sequence of non-repeating elements, called series, is used to embed a secret message. Embedding was performed by permutations of elements within the series. The proposed algorithm allows simulating statistical features of a message taken from an empty container, which reduces the probability of successful stegoanalysis. A stegoanalyst who does not know some secret parameter cannot correctly extract the message from the suspicious container and analyze its statistical properties. Thus, the steganalysis approach, based on detection statistical differences in a message taken from an empty and filled container, becomes ineffective.

Conclusions. The new method of steganographic message transformation of an empty container allows embedding hidden information, with minimal distortion of the statistical properties of the original container. In the course of the experiment statistical properties of the message taken from container before and after embedding were investigated. The analysis was carried out using previously known methods of steganalysis, based on detecting statistical differences in messages taken from empty and filled containers. It is shown that without knowing a certain secret parameter, the stegoanalyst cannot correctly extract up to 32 % of the message bits from the container, which leads to significant errors in the stegoanalysis. The volume of the message being embedded is 19 % of the length of the original message in the empty container. The algorithm presented in this paper is recommended to employ for embedding into containers: text, audio, executable files, graph containers. Using of this algorithm for images is not accounted for.

Keywords: steganography, stegoanalysis, secret data transmission, antidisturbance.

Cite: Nechta, I.V. A method of steganographic message transformation with the partial antidisturbance property // Computational Technologies. 2019. Vol. 24, No. 3. P. 75–87. (In Russ.) DOI: 10.25743/ICT.2019.24.3.006.

Received July 2, 2018
Received in revised form November 6, 2018