Адаптивный алгоритм стеганографического встраивания данных в цифровые изображения, использующий некриптографические хеш-функции для их извлечения

М.А. Дрюченко [1]
[1] *Воронежский государственный университет,*
394018, Россия, г. Воронеж, Университетская площадь, д. 1

Аннотация

Рассматривается адаптивный алгоритм стеганографического скрытия информации, основанный на итеративном внесении малозначительных искажений в блоки полноцветных изображений-контейнеров и использующий быстродействующие некриптографические хеш-функции для последующего извлечения скрытых данных. Особенностью алгоритма является модификация минимального числа элементов контейнера по сравнению с длиной скрываемых в него фрагментов сообщений, что позволяет увеличить показатели скрытой пропускной способности и снизить визуальную и статистическую заметность скрытых данных. Проводится сравнение алгоритма с современными алгоритмами адаптивного пространственного стегоскрытия в части оценки уровня искажающих изменений контейнеров. Рассматривается вариант повышения пропускной способности алгоритма за счёт мультиплексирования скрытых каналов, использующих общее подмножество элементов контейнера при встраивании в них различных сообщений.

Ключевые слова: стеганографическое скрытие информации, некриптографические хеш-коды, циклические избыточные коды.

Цитирование: **Дрюченко, М.А.** Адаптивный алгоритм стеганографического встраивания данных в цифровые изображения, использующий некриптографические хеш-функции для их извлечения / М.А. Дрюченко // Компьютерная оптика. – 2023. – Т. 47, № 3. – С. 415-425. – DOI: 10.18287/2412-6179-CO-1215.

Citation: Dryuchenko MA. An adaptive image steganography algorithm based on the use of non-cryptographic hash functions for data extraction. Computer Optics 2023; 47(3): 415-425. DOI: 10.18287/2412-6179-CO-1215.

Введение

Современные технологии компьютерной и цифровой стеганографии позволяют решать целый ряд задач, связанных с защитой конфиденциальной информации, находящейся в публичном доступе, а также задач, связанных с контролем распространения и использования различных объектов цифрового контента. Методы стеганографии реализуют скрытное внедрение пользовательских, аутентификационных или идентификационных данных (сообщений) в произвольные цифровые объекты (контейнеры), которые могут храниться в открытом виде или передаваться по незащищённым каналам коммуникации.

Одной из основных задач, которую приходится решать при разработке стегоалгоритмов, является минимизация неизбежных искажающих изменений в контейнерах, возникающих при встраивании в них сообщений. Вносимые искажения должны быть визуально и статистически незаметны и не должны нарушать функциональности контейнеров. Вторая задача, которая может ставиться при разработке стегоалгоритмов, связана с повышением их пропускной способности (ПС) $U=|M|/|I|$, оцениваемой как отношение размера скрываемого сообщения M к размеру контейнера I. В классических стегосистемах, предполагающих наличие контейнера и операций по его модификации, увеличение ПС почти неизбежно сопровождается увеличением искажающих изменений носителя.

Современные алгоритмы пространственной стеганографии, такие как WOW [1], S-UNIWARD [2], HUGO [3], реализуют идею адаптивного внедрения сообщений, решая задачу минимизации искажений путём целенаправленного выбора для внедрения данных лишь наиболее стохастичных областей контейнера, искажения в которых невозможно или достаточно сложно обнаружить даже с привлечением современных методов стегоанализа. К ограничениям данных и подобных им алгоритмов можно отнести сравнительно невысокую ПС, а также (зачастую) необходимость наличия исходных незаполненных контейнеров для извлечения данных. Среди алгоритмов адаптивной пространственной стеганографии, отличающихся повышенной ПС, можно отметить алгоритм MPBDH [4], реализующий блочное скрытие и модифицирующий до нескольких младших битовых плоскостей блоков изображения-контейнера в зависимости от их «сложности», а также алгоритм [5], реализующий встраивание информации лишь в наиболее шумные участки пространственного представления носителя путём перезаписи их младших разрядов (не более 4) битами сообщения. К объективным недостаткам ал-

горитмов с высокой ПС можно отнести их меньшую устойчивость по отношению к современным методам стегоанализа.

Алгоритмы стохастической модуляции [6, 7] минимизируют не абсолютные искажения контейнеров, а их статистические отличия от естественных незаполненных контейнеров путём внесения в них дополнительных шумов с заданными вероятностными распределениями, имитирующих естественные шумы контейнеров соответствующих форматов. В работе [8] минимизация статистических искажений графических контейнеров реализуется путём выделения высокочастотных стохастических составляющих самих контейнеров с последующим маскированием встроенных сообщений данными составляющими. Общий уровень искажений контейнеров, вносимых алгоритмами стохастической модуляции, может быть существенно выше, чем для алгоритмов адаптивного пространственного скрытия.

PVD-алгоритмы (Pixel Value Difference) пространственной стеганографии кодируют биты скрываемых сообщений как разность модифицированных значений пикселей блоков заданной конфигурации. В работах [9, 10] предложены адаптивные схемы скрытия на основе PVD, учитывающие информацию о вертикальных, горизонтальных и диагональных краях объектов на изображении, что позволяет минимизировать искажающие изменения маркированных контейнеров и обеспечить высокую устойчивость к PDH и RS-стегоанализу.

Особняком стоят появившиеся в последние годы алгоритмы [11 – 14], реализующие принципы «стеганография без контейнеров» (Coverless Steganography). Как следует из названия, они не используют традиционные контейнеры и не решают задачу минимизации их искажений. Скрываемая информация «кодируется» с использованием выборки объектов из предварительно сформированной большой коллекции объектов, например, набора изображений, каждый из которых может быть преобразован в короткую бинарную строку, соответствующую определённой части сообщения. На практике для преобразования объектов из выборки во фрагменты сообщения могут применяться различные алгоритмы классического и робастного хеширования [15]. В случае использования робастного хеширования передаваемые по открытым каналам объекты, кодирующие сообщение, могут быть дополнительно искажены, но это не должно существенным образом отразиться на возможности восстановления сообщения. Основным достоинством подобной схемы скрытной коммуникации является отсутствие любых стеганографических модификаций передаваемых объектов, т.е. абсолютная стеганографическая стойкость. К числу недостатков coverless-алгоритмов можно отнести невысокую ПС.

Также стоит отметить набирающие популярность стегоалгоритмы, реализующие принцип «стеганография без встраивания» (Steganography Without Embedding) и использующие современные модели машинного обучения для создания заполненных контейнеров на основе векторов скрываемых данных [16, 17]. Фактически данные алгоритмы не реализуют классических операций модификации или встраивания данных в контейнер, не маскируют присутствие полезного сигнала под «естественными» шумами, а лишь формируют синтетические заполненные контейнеры-изображения, из которых сообщение может быть извлечено обученной сетью-экстрактором. Заполненные контейнеры формируются с помощью специально обученных глубоких свёрточных генеративно состязательных сетей. Подобные алгоритмы показывают хорошие результаты в части противодействия современным статистическим и обучаемым методам стегоанализа. Несмотря на новизну и общую перспективность данной группы алгоритмов, на практике существуют определенные ограничения, связанные с необходимостью обучения, хранения и передачи достаточно больших моделей и сложностями с использованием имеющихся контейнеров.

В данной работе рассматривается вариант развития ранее предложенного в [18] алгоритма скрытия данных в цифровые изображения, позволяющего встраивать сообщения, длина которых существенно превышает число модифицируемых элементов носителя, что позволяет повысить ПС при сохранении искажений контейнера на низком уровне. Алгоритм предусматривает разбиение контейнера-изображения на непересекающиеся блоки с поиском для каждого из них такого минимально искажённого представления, которое обеспечивает корректное извлечение фрагмента сообщения – слова заданной разрядности. Для снижения визуальной и статистической заметности факта скрытия разрядность каждого встраиваемого слова адаптируется с учётом характера внутреннего наполнения блока. В более шумные блоки встраиваются слова большей длины, а в блоки с однородным содержимым – слова меньшей длины. Процедура извлечения скрытых данных предполагает использование т.н. функции «свёртки» H, примеряемой к блокам пикселей и преобразующей входные данные произвольной длины в значения фиксированной длины (слова сообщения). Используемая в алгоритме функция H должна отвечать следующим требованиям:

– высокая скорость вычисления (с увеличением разрядности встраиваемых слов количество вызовов функции H при поиске подходящих вариантов искажения блока возрастает экспоненциально);

– равномерное распределение значений на выходе функции H, лавинный эффект и минимизация частоты коллизий.

В качестве функции H, обеспечивающей требуемое преобразование входных данных, можно рассматривать криптографические и некриптографические хеш-функции или циклические избыточные ко-

ды. Криптографические хеш-функции были исключены из рассмотрения по причине избыточности для решаемой задачи как по времени работы, так и по разрядности формируемых ими кодов (≥ 128 бит), существенно превышающей обычную разрядность скрываемых слов сообщения (от 4 до 16 бит). Циклические избыточные коды характеризуются достаточно высоким быстродействием, но не всегда обеспечивают равномерность данных на выходе и приемлемую вероятность коллизий. В свою очередь, некриптографические хеш-функции, используемые в структурах данных для построения хеш-таблиц, характеризуются высоким быстродействием и хорошими статистическими свойствами формируемых выходных значений, поэтому при реализации стегоалгоритма предпочтение было отдано одной из наиболее производительных, обеспечивающих хорошую случайность выхода, мощный лавинный эффект и высокую устойчивость к коллизиям функции Murmur3 [19].

Отдельное внимание при разработке стегоалгоритма было уделено вопросам мультиплексирования скрытых каналов. В работе рассмотрен вариант создания пары скрытых каналов, формируемых одновременно с использованием одного и того же подмножества элементов контейнера, позволяющих в несколько раз увеличить фактическую ПС алгоритма без необходимости внесения в блоки контейнера дополнительных существенных искажений. В качестве контейнеров рассматривались цветные изображения в модели RGB. В качестве встраиваемых сообщений рассматривались обычные бинарные файлы.

Алгоритм встраивания данных

Укрупнённая блок-схема алгоритма встраивания данных приведена на рис. 1. Она включает следующие шаги.

Шаг 1. Загрузка файла-сообщения M и изображения-контейнера I размером $W\times H$ пикселей. Задание основных параметров встраивания, включающих размер обрабатываемых блоков ($w\times h$ пикселей), компоненты стеганографического ключа, включающие параметры для инициализации начального состояния генераторов псевдослучайных числовых последовательностей (ГПСЧП), используемых для определения порядка обхода блоков I и для рандомизации встраиваемых слов. Разбиение контейнера на непересекающиеся блоки $b_i \in I$, $b_i \cap b_j = \varnothing$, $\forall i \neq j$, $i, j = \overline{1,(W \cdot H)/(w \cdot h)}$.

Шаг 2. Добавление целочисленного значения длины сообщения L_M в его начало.

Шаг 3. Выбор в соответствии с ГПСЧП-1 на I очередного блока $b_i \in I$ и оценка его гладкости с целью определения оптимальной длины скрываемого в блок слова. Внесение при стегоскрытии сколь бы то ни было значимых искажений в гладкие блоки контейнера негативно отражается на визуальной и статистической заметности факта скрытия. В то же время шумные блоки кадра допускают потенциально бóльшие искажения, которые с меньшей вероятностью могут быть обнаружены при стегоанализе. В общем случае гладкость изображения можно рассматривать как функцию цветовых градиентов. В качестве параметра гладкости блока использовалось среднее значение дисперсий, вычисленных для модулей градиента блока в каждом его канале

$$d_i = \frac{1}{3}\sum\nolimits_{ch=R,G,B} D\left(grad\left(b_{i,ch}\right)\right).$$

Эмпирическим путем была предложена следующая шкала «гладкости», определяющая разрядность n встраиваемых и извлекаемых из блоков контейнера слов:

$$c_i = \begin{cases} 0,\ d_i < 0{,}15,\ n = 0; \\ 1,\ 0{,}15 \le d_i < 0{,}3,\ n = 4; \\ 2,\ 0{,}3 \le d_i < 2,\ n = 8; \\ 3,\ 2 \le d_i < 7,\ n = 12; \\ 4,\ d_i \ge 7,\ n = 16. \end{cases}$$

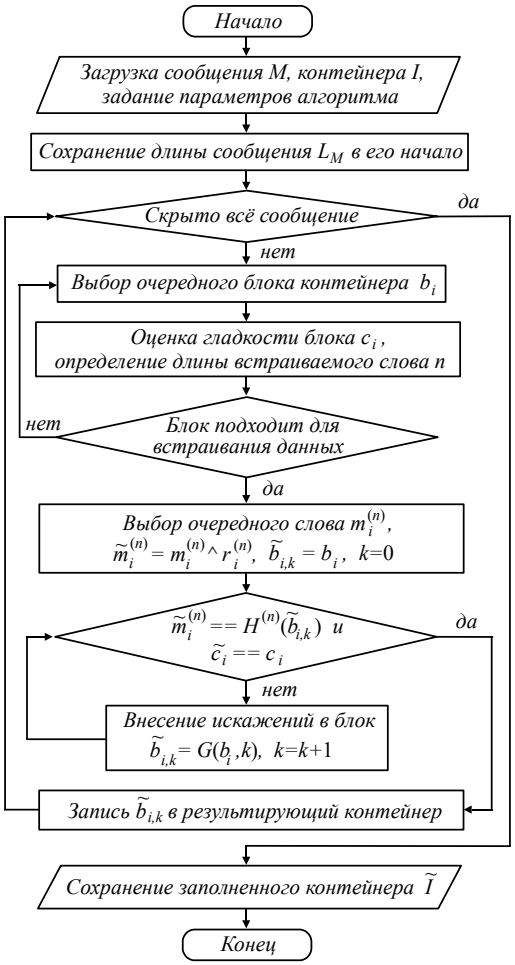

Рис. 1. Обобщённая схема работы алгоритма встраивания данных

Пример разметки блоков изображения «Lena» в соответствии с предложенной шкалой «гладкости» приведён на рис. 2*б*.

а)

б) ■ n=0; ■ n=4; ■ n=8; ■ n=12; □ n=16

Рис. 2. Исходное (а) и размеченное по уровню гладкости блоков (б) изображение «Lena»

При сохранении размеров блока и увеличении разрядности скрываемых слов будет увеличиваться вероятность искажений блока как по числу модифицируемых элементов, так и по амплитуде вносимых искажений. Отметим, что уменьшение разрядности встраиваемых слов в первую очередь оправдано при работе с небольшими блоками ($w \le 4$, $h \le 4$,). Для блоков большей размерности количество получаемых на шаге 5 алгоритма модифицированных вариантов блока с малым уровнем искажений будет достаточно большим, чтобы выбрать необходимый вариант, обеспечивающий извлечение данных.

Шаг 4. Выбор очередного n-битного слова $m_i^{(n)} \in M$ и его рандомизация сложением по модулю 2 со значением $r_i^{(n)}$, формируемым на выходе ГПСЧП-2, $\tilde{m}_i^{(n)} = m_i^{(n)} \oplus r_i^{(n)}$.

Шаг 5. В цикле поиск подходящего варианта искажения блока $\tilde{b}_{i,k}$ такого, что

$$H^{(n)}\left(\tilde{b}_{i,k}\right) = \tilde{m}_i^{(n)}, \ \left\| b_i - \tilde{b}_{i,k} \right\| \to \min \ \text{и} \ c_i = \tilde{c}_i. \qquad (1)$$

Т.е. совпадение результата вычисления некриптографической хеш-функции $H^{(n)}$ от искажённого блока $\tilde{b}_{i,k}$ обеспечивает возможность корректного извлечения слова $\tilde{m}_i^{(n)}$ при условии сохранения прежнего уровня «гладкости» блока. Для модификации элементов b_i применяется функция G вида

$$\begin{aligned} \tilde{b}_{i,k} &= G\left(b_i, k\right) = \\ &= \left(b_i\left(1\right) + \psi_{1,k}, b_i\left(2\right) + \psi_{2,k}, \ldots, b_i\left(L_b\right) + \psi_{L_b,k}\right), \end{aligned} \qquad (2)$$

которая в зависимости от текущего значения переменной счётчика k добавляет к значениям цветов пикселей блока элементы вектора $\Psi_k = (\psi_{j,k}, \ldots, \psi_{L_b,k})^T$, где $\psi_{j,k}$, $j = \overline{1, L_b}$ могут принимать значения $\{0, \pm 1, \pm 2, \ldots, \pm \lambda\}$, λ – максимальное значение, добавляемое к значению цвета пикселя (обычно $\lambda \le 3$), $L_b = 3 \cdot w \cdot h$ – число доступных для модификации элементов блока для полноцветных контейнеров. Векторы $\Psi_k, k = \overline{1, K}$ формируются таким образом, чтобы при малых k количество и уровень вносимых изменений в b_i были минимальными, при этом в первую очередь затрагиваются пиксели в синем цветовом канале, потом в красном и зелёном. С увеличением k число одновременно модифицируемых коэффициентов в b_i, а также абсолютные значения приращений $\psi_{j,k}$ постепенно увеличиваются.

После того как найдена подходящая модификация $\tilde{b}_{i,k}$, она записывается в результирующий контейнер. Если просмотрены не все слова из M, выполняется переход к шагу 2.

Шаг 6. Сохранение заполненного контейнера $\tilde{I}$.

Если принять гипотезу о том, что на выходе функции $H^{(n)}$ формируются случайные последовательности значений с равномерным распределением, то теоретически для успешной работы алгоритма встраивания количество векторов Ψ_k, определяемое числом возможных неповторяющихся комбинаций значений приращений, должно быть сопоставимо с 2^n. На практике с учётом возможных коллизий для некриптографических хеш-функций количество векторов Ψ_k должно в несколько раз превосходить размерности пространства решений 2^n.

Типичные гистограммы распределения числа итераций искажений блоков при встраивании слов разрядностью 8 и 16 бит приведены на рис. 3. В зависимости от используемой функции H, а именно от вероятности коллизий и стохастичности её выхода, гистограммы могут несколько отличаться. Для функции Murmur3 средние и медианные значения числа итераций, необходимых для встраивания 4-битных слов, составили 14 и 9, для 8-битных – 254 и 175, для 12-битных – 4082 и 2827, для 16-битных – 67280 и 44826 соответственно.

Общее число K векторов приращений Ψ_k зависит от размера модифицируемого блока, а также от максимального значения искажения цвета одного элемента λ, заданного максимального числа одновременно искажаемых элементов и от значений, на которые они искажаются. Для оценки общего количества возможных комбинаций искажений можно использовать формулы комбинаторики – число сочетаний и/или размещений без повторений, применяемых для подсчёта различных вариантов искажения. Так, для блока размером 3×3 для $\lambda \le 2$ и количестве одновременно модифицируемых значений яркости в $\tilde{b}_{i,k}$ не более трех $K = 157824$. Для блока размером 5×5 и аналогичных значений максимального уровня искажений

и числа одновременно модифицируемых элементов $K = 3556000$. Для рассматриваемых в данной работе значений разрядности скрываемых слов $4 \le n \le 16$ количество формируемых вариантов $\tilde{b}_{i,k}$ с малыми относительными искажениями оказывается достаточным для выполнения (1) даже для блоков малого размера. С увеличением размеров блока естественным образом увеличивается число возможных комбинаций вносимых искажений и, соответственно, необходимый результат (1) можно получить для комбинаций, соответствующих минимальным по уровню и числу подвергшихся модификации элементов, что позволяет встраивать слова, длина которых в битовом представлении существенно больше числа модифицируемых элементов блока.

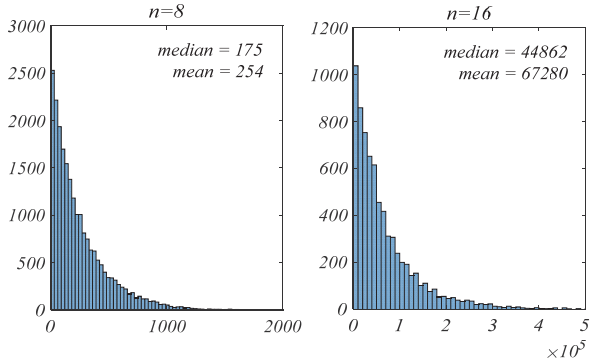

Рис. 3. Гистограммы распределения числа итераций искажений блоков при встраивании слов разрядностью 8 и 16 бит

Также стоит отметить, что на шаге 1 алгоритма встраивания возможен вариант формирования блоков из пикселей, выбранных в псевдослучайном порядке равномерно по всему контейнеру, что потенциально позволяет усложнить процедуру стегоанализа. Однако вследствие некоррелированности добавляемых в блоки пикселей из разных участков кадра количество сформированных блоков, имеющих малую гладкость и предполагающих встраивание слов большей разрядности, будет существенно выше, чем при формировании блоков из смежных пикселей, что негативным образом отразится на уровне искажения заполненных контейнеров. Поэтому при формировании содержимого блоков лучше рассматривать смежные пиксели, а форма блоков и порядок их обхода на кадре могут дополнительно рандомизироваться с использованием ГПСЧП.

Алгоритм извлечения данных

Укрупнённая блок-схема алгоритма извлечения данных приведена на рис. 4. Данный алгоритм не требует наличия исходного незаполненного контейнера, отличается вычислительной простотой и включает следующие шаги.

Шаг 1. Загрузка заполненного контейнера $\tilde{I}$, задание основных параметров алгоритма, включающих размер блоков и начальные состояния используемых ГПСЧП.

Шаг 2. Выбор в соответствии с ГПСЧП-1 на $\tilde{I}$ очередного блока $\tilde{b}_i$.

Шаг 3. Оценка гладкости блока $\tilde{b}_i$ и определение длины извлекаемого слова n. Если блок гладкий ($\tilde{c}_i = 0$), то он пропускается и осуществляется возврат к шагу 2.

Шаг 4. Извлечение очередного слова путём вычисления некриптографической хеш-функции от блока $\tilde{m}_i^{(n)} = H^{(n)}(\tilde{b}_i)$. Декодирование исходного значения слова путём его сложения по модулю два с выходом ГПСЧП-2 $m_i^{(n)} = \tilde{m}_i^{(n)} \oplus r_i^{(n)}$.

Если длина сообщения M не сформирована, то $m_i^{(n)}$ записывается в соответствующую целочисленную переменную L_M, иначе $m_i^{(n)}$ добавляется в результирующее сообщение M. Если число извлечённых байт меньше L_M, выполняется переход к шагу 2.

Шаг 5. Сохранение извлечённого сообщения M.

В решаемой задаче используемые при извлечении сообщений в качестве $H^{(n)}$ некриптографические хеши не являются инъективными и допускается появление коллизий, когда для блока b_i существуют различные его модификации $\tilde{b}_{i,p} \ne \tilde{b}_{i,q}$, для которых $H^{(n)}(\tilde{b}_{i,p}) = H^{(n)}(\tilde{b}_{i,q})$. Наличие коллизий отражается на увеличении общего числа итераций искажения блоков контейнера, а значит, и времени работы алгоритма при работе в режиме поиска первой подходящей модификации блока (1). С другой стороны, наличие коллизий позволяет реализовать процедуру отбора из множества искажённых представлений блоков, обеспечивающих извлечение скрытых слов, таких вариантов, которые можно рассматривать в качестве прообразов для $H^{(v)}$ при извлечении v-битных слов в режиме мультиплексирования скрытых каналов.

Мультиплексирование скрытых каналов

Классические варианты мультиплексирования скрытых каналов, как правило, предполагают выделение в контейнере непересекающихся подмножеств модифицируемых элементов с последующим встраиванием в них информации, предназначенной для различных пользователей [18]. Описанные ниже принципы встраивания и извлечения информации могут быть использованы для организации нескольких скрытых каналов в рамках одного контейнера, предназначенных для скрытия различных сообщений, но, в отличие от классических вариантов мультиплексирования, работающих на одном и том же подмножестве элементов носителя.

Рассмотрим алгоритм мультиплексирования на примере создания в контейнере-изображении двух скрытых каналов C_M и C_S, используемых для передачи независимых сообщений M и S. Для простоты изложения рассмотрим алгоритм встраивания без адаптации длин скрываемых слов под гладкость каждого блока – будем считать, что во все пригодные для встраивания данных блоки контейнера I скрываются n-битные слова из M, а в пары различных блоков – v-

битные слова из S. $M = m_1^{(n)} \| m_2^{(n)} \| \ldots m_{N_M}^{(n)}$, N_M – число n-битных слов, составляющих скрываемое сообщение M. $S = s_1^{(v)} \| s_2^{(v)} \| \ldots s_{N_S}^{(v)}$, N_S – число v-битных слов, составляющих сообщение S.

Укрупнённая блок-схема алгоритма приведена на рис. 5. Она включает следующие шаги.

Шаг 1. Загрузка сообщений M и S. Добавление целочисленных значений длин каждого сообщения в его начало. Задание основных параметров алгоритма, включающих размер обрабатываемых блоков, разрядность слов n и v, компоненты стеганографических ключей, включающие параметры для инициализации используемых ГПСЧП, рандомизирующих встраиваемые слова, а также определяющие порядок обхода блоков при встраивании M и S.

Шаг 2. Выбор на I очередного блока b_i, оценка его гладкости c_i (если $c_i = 0$ – пропуск блока и переход в начало шага 2). Выбор очередного n-битного слова $m_i^{(n)} \in M$ и его рандомизация $\tilde{m}_i^{(n)} = m_i^{(n)} \oplus r_i^{(n)}$, $r_i^{(n)}$ – n-битное значение на выходе ГПСЧП-1.

Шаг 3. Вычисление множества различных вариантов искажений блока b_i $\tilde{B}_i = \{\tilde{b}_{i,1}, \ldots \tilde{b}_{i,Ni}\}$, таких, что

$$H^{(n)}\left(\tilde{b}_{i,k}\right) = \tilde{m}_i^{(n)}, \;\; \tilde{c}_i \neq 0, \;\; \forall k = \overline{1, N_i}. \tag{3}$$

В терминологии криптографии элементы $\tilde{B}_i$ являются коллизиями для функции $H^{(n)}$ (N_i – число коллизий) и каждый из $\tilde{b}_{i,k}$ обеспечивает корректное извлечение слова $\tilde{m}_i^{(n)}$. Для модификации элементов b_i применяется функция G. Если просмотрены не все слова из M, выполняется переход к шагу 2.

Шаг 4. Выбор очередного v-битного слова $s_l^{(v)} \in S$ и его рандомизация $\tilde{s}_l^{(v)} = s_l^{(v)} \oplus r_l^{(v)}$, $r_l^{(v)}$ – v-битное значение на выходе ГПСЧП-2.

Шаг 5. Согласно ГПСЧП-3 выбор очередной пары модифицировавшихся блоков b_i, b_j. Поиск в сформированных на шаге 3 множествах вариантов искажений $\tilde{B}_i$ и $\tilde{B}_j$ элементов, для которых справедливо

$$\begin{gathered} \tilde{s}_l^{(v)} = H^{(v)}\left(\tilde{b}_{i,p} \| \tilde{b}_{j,q}\right), \\ \left\|b_i - \tilde{b}_{i,p}\right\| \to \min, \;\; \left\|b_j - \tilde{b}_{j,q}\right\| \to \min, \end{gathered} \tag{4}$$

где $\|$ – операция конкатенации блоков (на уровне байтовых массивов), $H^{(v)}$– некриптографическая хеш-функция с выходом длиной v-бит. Запись содержимого модифицированных блоков $\tilde{b}_{i,p}, \tilde{b}_{j,q}$ на соответствующие позиции в результирующем контейнере $\tilde{I}$. Для обеспечения возможности встраивания и извлечения v-битных слов $\tilde{s}_l^{(v)}$ необходимо, чтобы число различных пар искажённых блоков b_i, b_j в (4), оцениваемое как произведение мощностей множеств $\tilde{B}_i$ и $\tilde{B}_j$, было не меньше, чем 2^v (на практике с учётом возможных коллизий в несколько раз больше).

Переход к шагу 4, если $1 < N_S$, т.е. встроены не все слова из S.

Шаг 6. Если остались модифицировавшиеся на шаге 3, но не использованные для встраивания S блоки, из соответствующих им множеств $\tilde{B}_i$ осуществляется выбор и запись в результирующий контейнер искажённых представлений блоков $\tilde{b}_{i,p}$, минимально отличающихся от исходных $\|b_i - \tilde{b}_{i,p}\| \to \min$.

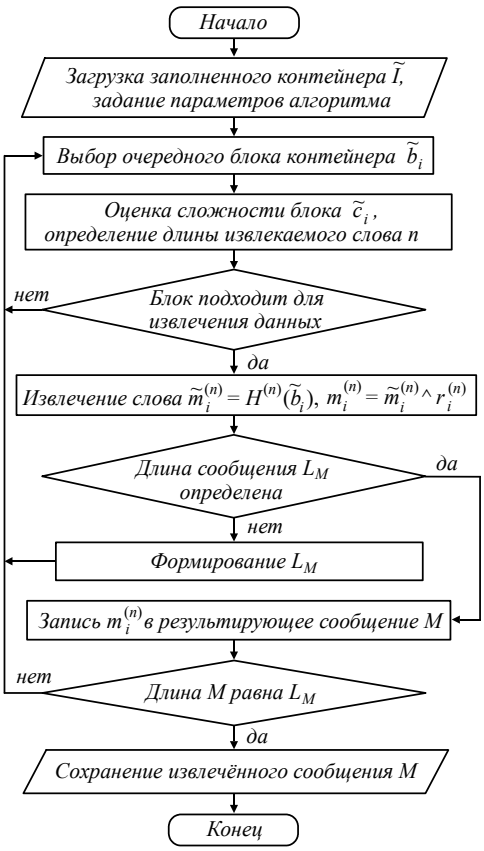

Рис. 4. Обобщённая схема работы алгоритма извлечения данных

Шаг 7. Сохранение заполненного контейнера $\tilde{I}$.

В результате формируется заполненный контейнер $\tilde{I}$ с парой скрытых каналов передачи информации C_M и C_S. Формирование данных каналов реализуется одновременно. В общем случае соотношение между размерами сообщений M и S может быть произвольным: $|M| > |S|$, $|M| < |S|$, $|M| = |S|$, все зависит от выбранной разрядности скрываемых слов каждого сообщения. Например, если $n = 4$, а $v = 16$ и для скрытия используются все подходящие блоки контейнера, то $|S| = 2|M|$. Соотношение между числом модифицируемых блоков контейнера подчиняется неравенству $N_S \le N_M / 2$.

В табл. 1 приводятся усреднённые по кадрам тестовой выборки значения частоты коллизий, формируемых для блоков заданного размера при встраивании в них n-битных слов. Каждая из коллизий обеспечивает корректное извлечение n-битного слова, но также характеризуется различным уровнем искажения соответствующего ей блока. С использованием данных из табл. 1 можно оценить примерную размерность пространства решений для задачи встраивания слов различной разрядности из сообщения S. Так, для блока размером 3×3 пикселя, $n = 8$, максимальной амплитуды вносимых искажений $\lambda = 1$ и максимальным числом

одновременно искажаемых пикселей блока $N_{corr} \le 4$ среднее число комбинаций пар искажённых представлений блока составит $(2575)^2$, что более чем достаточно для встраивания слова $s_l^{(v)}$ разрядностью 16 бит.

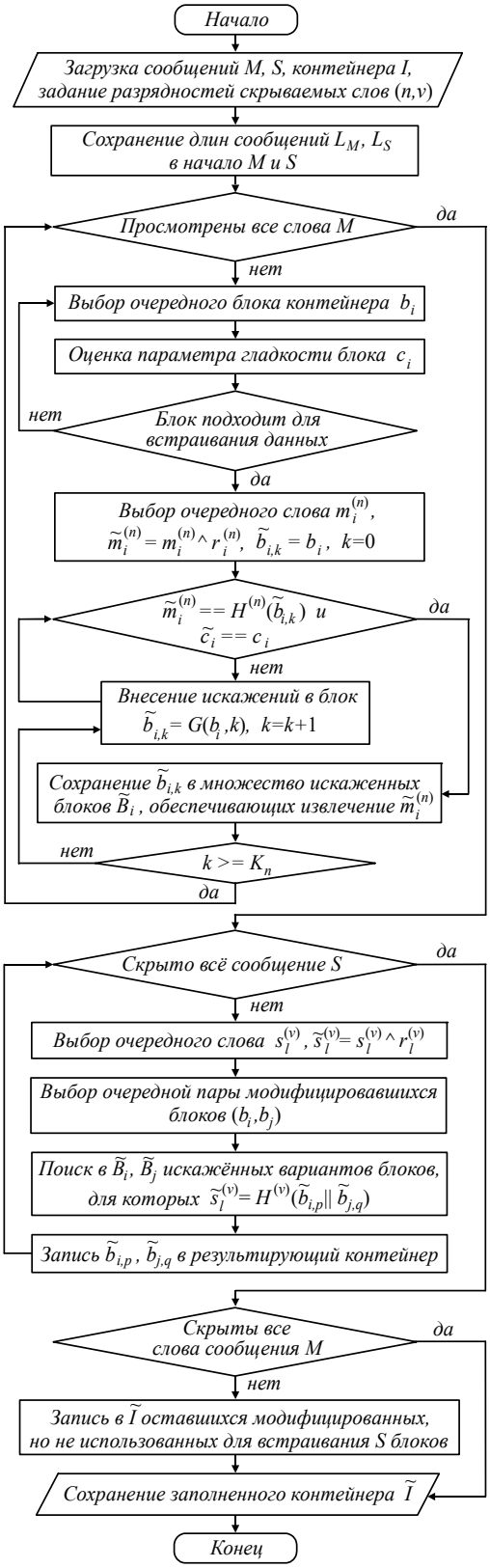

Рис. 5. Обобщённая схема алгоритма встраивания данных при создании двух скрытых каналов

Табл. 1. Среднее число искажённых вариантов блока, каждый из которых обеспечивает корректное извлечение n-битного слова

Размер и параметры искажения блока	$n=4$	$n=8$	$n=12$	$n=16$
3×3, $\lambda=1$, $N_{corr} \le 3$	1503	88	7	0,6
3×3, $\lambda \le 2$, $N_{corr} \le 3$	8352	526	34	3
3×3, $\lambda=1$, $N_{corr} \le 4$	39381	2575	167	14
4×4, $\lambda=1$, $N_{corr} \le 3$	8852	536	36	2
4×4, $\lambda \le 2$, $N_{corr} \le 3$	47405	3072	192	11
4×4, $\lambda=1$, $N_{corr} \le 4$	480102	30896	1867	146
5×5, $\lambda=1$, $N_{corr} \le 3$	34446	2 068	121	7
5×5, $\lambda \le 2$, $N_{corr} \le 3$	184329	11594	719	42
5×5, $\lambda=1$, $N_{corr} \le 4$	3166088	197318	12328	748

Алгоритм извлечения сообщения из канала C_M будет аналогичен представленному на рис. 4 с той разницей, что по результатам оценки гладкости блоков не производится определение разрядности извлекаемых слов. Если алгоритм скрытия с мультиплексированием каналов выполнялся с адаптацией длин встраиваемых слов, то схема извлечения данных из C_M будет идентична представленной на рис. 4.

Алгоритм извлечения сообщения из канала C_S включает следующие шаги.

Шаг 1. Загрузка заполненного контейнера $\tilde{I}$.

Шаг 2. Согласно ГПСЧП-3 выбор на $\tilde{I}$ очередной пары блоков $\tilde{b}_i, \tilde{b}_j$, оценка их гладкости (гладкие блоки пропускаются, и выполняется переход в начало шага 2).

Шаг 3. Извлечение и декодирование слова $\tilde{s}_l^{(v)} = H^{(v)}(\tilde{b}_i \| \tilde{b}_j)$, $s_l^{(v)} = \tilde{s}_l^{(v)} \oplus r_l^{(v)}$. Если длина сообщения S не сформирована, запись $s_l^{(v)}$ в соответствующую целочисленную переменную L_S. Иначе запись $s_l^{(v)}$ в результирующее сообщение S.

Если число извлечённых байт меньше L_S, переход к шагу 2.

Шаг 4. Сохранение сообщения S.

Схематичная графическая интерпретация операций встраивания/извлечения данных из первого и второго скрытых каналов приведена на рис. 6.

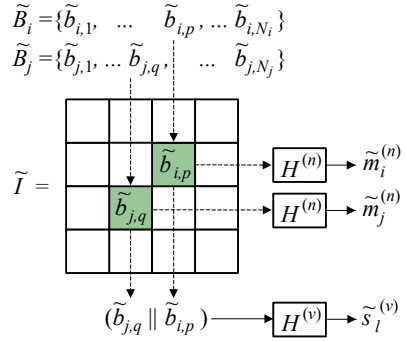

Рис. 6. Пример выбора искажённых блоков, обеспечивающих извлечения данных из первого и второго скрытых каналов

На рис. 7 приведён пример исходных и модифицированных в режиме мультиплексирования скрытых каналов блоков контейнера при встраивании пары слов длиной 8 бит в C_M и одного 8-битного слова в C_S. В ка-

честве $H^{(8)}$ использовалась некриптографическая хеш-функция Murmur3 с финализацией выхода по разрядности встраиваемых слов ($seed = 135$), $\lambda \le 2$, $N_{corr} \le 3$.

В общем случае каналы C_M и C_S могут использоваться для скрытной передачи данных любого характера, предназначенных различным абонентам. Особенность формирования скрытых каналов на основе одного и того же подмножества элементов контейнера допускает использование C_M как второстепенного несекретного канала, маскирующего наличие канала C_S, содержащего конфиденциальную информацию. В классической «проблеме заключённых» или в задаче о взаимодействующих агентах при обнаружении скрытого канала обмена, прямые его участники Алиса и/или Боб в условиях непреодолимых обстоятельств по требованию третьей стороны (охранника Вилли) теоретически могут раскрыть скрытно передаваемое «несекретное» сообщение M, объяснив соответствующие искажения в контейнере внесением в него конкретных элементов M в процедуре стеганографического встраивания. Естественно, что при этом не должен разглашаться тот факт, что эти же самые искажения вносились в контейнер и для встраивания второго «секретного» сообщения S. Используя предложенные принципы создания каналов C_M, C_S и зная принципы работы алгоритма встраивания/извлечения данных, при детальном анализе заполненного контейнера не возникает расхождений в числе модифицированных блоков и числе скрытых слов сообщения M, что исключает или минимизирует подозрения о существовании S.

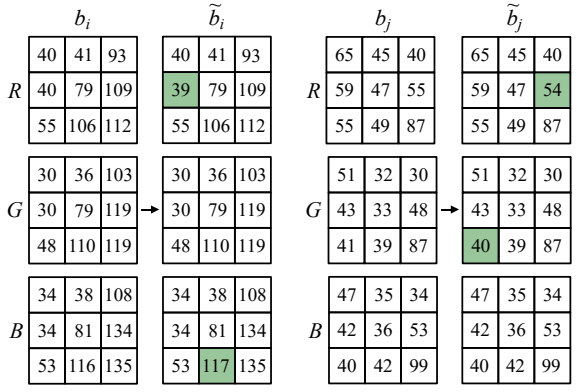

Рис. 7. Пример пары искажённых блоков $\tilde{b}_i$, $\tilde{b}_j$, обеспечивающих извлечение тройки слов $m_i^{(8)}$, $m_j^{(8)}$, $s_l^{(8)}$

Результаты экспериментальных исследований и их обсуждение

Экспериментальный анализ алгоритма стегоскрытия проводился в части оценки зависимостей среднего числа модифицируемых элементов и уровня их искажений в блоках различного размера от длин встраиваемых в данные блоки слов, итогового уровня искажений стегоконтейнеров, а также времени, затрачиваемого алгоритмом на создание заполненных контейнеров. Тестирование проводилось на выборке из 50 полноцветных изображений в модели RGB из базы данных PPG-LIRMM-COLOR [20]. В базе представлены кадры размером 512×512 пикселей, хранящиеся в формате bmp. В качестве скрываемых сообщений рассматривались псевдослучайные последовательности чисел требуемой длины, которые формировались алгоритмом MT19937.

На рис. 8 приводятся зависимости среднего числа модифицируемых значений в блоках различного размера от разрядности встраиваемых в данные блоки слов. С уменьшением размеров блоков и увеличением длины скрываемых слов естественным образом увеличивается число потенциально модифицируемых элементов. Так, для встраивания 16-битного слова в блок размером 3×3 в среднем необходимо модифицировать три элемента. Для встраивания 8-битного слова в блок того же размера в среднем необходимо модифицировать менее двух элементов.

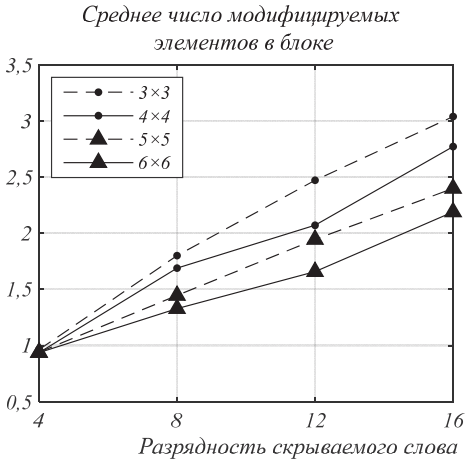

Рис. 8. Зависимость среднего числа модифицируемых элементов блока от разрядности скрываемых слов

На рис. 9 приводятся зависимости среднего значения амплитуды искажения модифицируемых элементов в блоках от разрядности встраиваемых в них слов. Амплитуда искажений определяется приращениями λ в функции искажения и измеряется в уровнях яркости/цветности. Даже для блоков малого размера средний уровень искажения подмножества модифицируемых при встраивании элементов не превышает 1,15. Для встраиваемых слов разрядностью 8 бит средний уровень искажений примерно равен 1 для всех рассмотренных размерностей блоков.

Приведённые на рис. 8, 9 зависимости получены для стегоконтейнеров, сформированных без мультиплексирования скрытых каналов. В режиме создания двух каналов для малых разрядностей скрываемых слов $n = 4$ и $v \le 8$ значения среднего числа модифицируемых элементов, а также средний уровень вносимых искажений будут сопоставимы с приведёнными выше. Для $n \ge 8$ и $v \ge 8$ минимальное необходимое для скрытия M и S число модифицируемых элементов и ам-

плитуда модификации в среднем оказываются несколько больше. Насколько – зависит от отношения разрядностей скрываемых слов, размеров обрабатываемых блоков и заданных максимальных значений λ и N_{corr}.

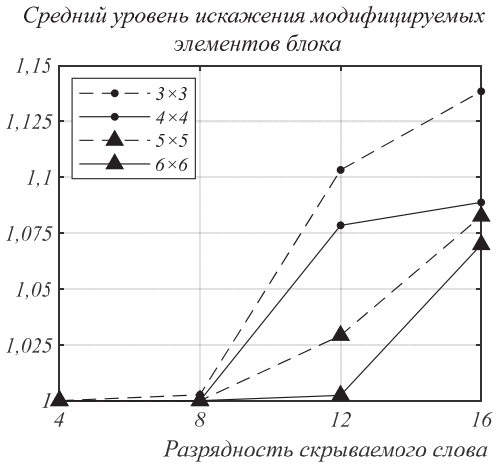

Рис. 9. Зависимость среднего уровня искажения модифицируемых элементов блока от разрядности скрываемых слов

Для оценки степени искажения заполненных контейнеров рассматривались традиционные метрики – пиковое отношение сигнала к шуму (PSNR), индекс структурного сходства (SSIM) и максимальная абсолютная ошибка (MAE). В табл. 2 представлены результаты оценки данных показателей для рассматриваемого в работе алгоритма, использующего некриптографические хеш-функции для извлечения скрытых данных (обозначен аббревиатурой AHX) с параметрами размера блока 4×4, $\lambda \le 2$, $N_{corr} \le 3$, а также для пары современных алгоритмов адаптивной пространственной стеганографии WOW и S-UNIWARD. Для всех алгоритмов реализовывалось встраивание в RGB-изображения с фактическим одинаковым объёмом полезной нагрузки α, определяемым средним количеством встраиваемых бит стегосообщения, приходящихся на пиксель контейнера. При оценке объёма полезной нагрузки факт пропуска блоков с малой стохастичностью в выбранных для сравнения алгоритмах также учитывался. Для всех α предложенный алгоритм показал меньшие искажения, чем WOW и S-UNIWARD, по показателям PSNR и SSIM, что объясняется используемыми принципами встраивания, позволяющими скрывать больше бит, чем число фактически модифицируемых бит носителя. В то же время при $\alpha > 0{,}5$ в маркированных предложенным алгоритмом контейнерах встречаются отдельные блоки с бóльшим уровнем абсолютных искажений, чем в контейнерах, сформированных алгоритмами WOW и S-UNIWARD.

В табл. 3 приведены усреднённые по тестовой выборке значения времени (в секундах) создания заполненного полноцветного контейнера размером 512×512 пикселей для различных вариантов алгоритма стегоскрытия, предусматривающих фиксированную или автоматически определяемую разрядность встраиваемых слов. При задании фиксированной разрядности скрытие n-битных слов происходило во все блоки размером 4×4 пикселя, «гладкость» которых превышала 1. Контейнеры создавались из расчёта 100 % их заполнения скрытой информацией. Результаты получены для двух вариантов функции H – некриптографической хеш-функции Murmur3 с финализацией длины её выхода под размер слова, а также для быстрых табличных реализаций циклических избыточных кодов CRC-8 и CRC16-CITT, используемых при скрытии слов разрядностью 4, 8 и 12, 16 бит соответственно.

Табл. 2. Результаты сравнения алгоритмов стегоскрытия по показателям искажения заполненных контейнеров

Наименование алгоритма	PSNR	SSIM	MAE
	$\alpha = 0{,}1$		
WOW	66,433	0,9999	1
S-UNIWARD	66,778	0,9999	1
AHX	69,846	0,9999	1
	$\alpha = 0{,}5$		
WOW	59,319	0,9996	1
S-UNIWARD	59,032	0,9994	1
AHX	62,988	0,9999	1
	$\alpha = 1$		
WOW	55,348	0,9986	1
S-UNIWARD	54,796	0,9973	1
AHX	59,959	0,9997	2

Табл. 3. Среднее время (в секундах) создания заполненного контейнера размером 512×512 пикселей при 100 % его заполнении скрытой информацией

Разрядность встраиваемых слов	Тип функции H	
	Murmur3	CRC-8, CRC-16-CCITT
$n = 4$	0,018	0,019
$n = 8$	0,041	0,075
$n = 12$	0,375	1,896
$n = 16$	3,386	27,303
Определяется автоматически	0,759	5,732

В таблице представлены результаты для распараллеленного варианта алгоритма скрытия (многопоточная реализация на языке C++, компилятор MinGW gcc-7.3.0, CPU Intel Core i5-10400 2.90ГГц, 16Гб ОЗУ). Несмотря на достаточно большое число итераций искажения блоков при встраивании в них 16-битных слов, общее время создания заполненного контейнера не превышает 3,5 секунд для используемой по умолчанию некриптографической хеш-функции Murmur3. При аналогичных условиях реализация алгоритма с циклическими избыточными кодами оказалась в 8 раз медленнее. При меньших разрядностях встраиваемых слов, а также в варианте алгоритма с адаптацией длин скрываемых слов под гладкость блоков среднее время работы алгоритма встраивания с функцией Murmur3 составило менее 1 секунды на кадр, что в среднем в 1,8 – 7,5 раз быстрее, чем для реализации с CRC.

Усреднённые значения времени создания контейнеров для алгоритма, работающего в режиме мультиплексирования скрытых каналов, приведены в табл. 4.

Для ускорения работы алгоритма в режиме создания пары скрытых каналов полные множества искажённых вариантов блоков $\tilde{B}_i$ и $\tilde{B}_j$ (см. табл. 1) не формируются. Добавление подходящих искажённых вариантов блоков в соответствующие множества реализуется последовательно с одновременной проверкой выполнения условия (4) для всех пар блоков, которые можно сформировать с текущим добавляемым искажённым блоком. При формировании возможных пар множества $\tilde{B}_i$ и $\tilde{B}_j$ просматриваются сначала. Если найдена подходящая пара блоков, то их содержимое сохраняется в результирующий контейнер и алгоритм переходит к встраиванию следующей тройки слов. Таким образом удаётся существенно сократить время работы алгоритма и обеспечить минимальные искажения итогового контейнера.

Табл. 4. Среднее время (в секундах) создания заполненного контейнера размером 512×512 пикселей при 100 % его заполнении скрытой информацией в режиме мультиплексирования скрытых каналов

Разрядность встраиваемых слов	Размер блоков 4×4, $\lambda \le 2$, $N_{corr} \le 4$
$n=4$, $v=4$	0,637
$n=4$, $v=8$	0,828
$n=4$, $v=12$	1,470
$n=4$, $v=16$	4,055
$n=8$, $v=\,=\,=4$	0,607
$n=8$, $v=8$	1,785
$n=8$, $v=12$	7,621
$n=8$, $v=16$	28,869

Подробное рассмотрение вопросов стеганографической стойкости предложенного алгоритма выходило за рамки данной работы. Тем не менее с использованием глубокой свёрточной модели Yedrouj-Net [21] было проведено тестирование в части сравнения стеганографической стойкости предложенного алгоритма (размер блока – 4×4, $\lambda \le 2$, $N_{corr} \le 3$,) с алгоритмами WOW и S-UNIWARD для двух вариантов «загрузки» контейнеров $\alpha = 0{,}1$ и $\alpha = 1$. Yedrouj-Net входит в число современных эффективных моделей, применяемых для стегоанализа в пространственном представлении контейнеров. Её архитектура предполагает использование шести свёрточных и трёх полносвязных слоев. В первых свёрточных слоях используются нелинейные активации в виде функций abs и trunk. При проведении тестирования использовалась база изображений PPG-LIRMM-COLOR. В ходе исследований для анализа каждого алгоритма скрытия при заданном значении объёма полезной нагрузки обучалась отдельная модель. Каждым алгоритмом формировалось по 2000 заполненных контейнеров. В качестве скрываемых данных рассматривался выход ГПСЧП. Тренировочные наборы включали по 1500 заполненных и соответствующих им исходных контейнеров, для валидации и тестирования использовались по 250 оставшихся заполненных и соответствующих им исходных контейнеров, гиперпараметры задавались согласно [21]. При $\alpha = 0{,}1$ усреднённые значения точности детектирования факта стегоскрытия для всех алгоритмов оказались примерно одинаковы и составили 60 % для предложенного алгоритма и WOW и 58 % для S-UNIWARD. При $\alpha = 1$ наибольшая усредненная точность обнаружения в 94 % была зафиксирована для предложенного алгоритма, в то время как для алгоритмов WOW и S-UNIWARD соответствующие значения составили 87 и 83 %. Отчасти меньшая устойчивость предложенного алгоритма к стегоанализу при увеличении объемов полезной нагрузки обуславливается бóльшим уровнем абсолютных искажений (MAE ≥ 2), вносимых в отдельные немногочисленные элементы контейнера, а также менее избирательным характером самого алгоритма выбора пикселей для модификации в рамках обрабатываемых блоков. Следует обратить внимание на то, что WOW и S-UNIWARD в первую очередь разрабатывались как алгоритмы, способные эффективно противодействовать стегоанализу. Подробное рассмотрение вопросов стеганографической стойкости предложенного алгоритма и проведение более масштабных его исследований планируется в следующей статье по данной тематике.

Заключение

Описанные в работе принципы встраивания/извлечения данных являются универсальными и могут применяться как в пространственном, так и в частотном представлении контейнеров различных форматов. Предложенный алгоритм не является робастным и предназначен для создания стегоконтейнеров, не подвергающихся дополнительным искажениям. Алгоритм имеет настраиваемую ПС, зависящую от размеров искажаемых блоков, разрядности встраиваемых слов, характера содержимого самого контейнера, определяющего процент пропускаемых блоков или блоков, для которых длина встраиваемых слов автоматически уменьшается. На практике алгоритм способен обеспечить соотношение числа фактически модифицируемых бит контейнера к числу встраиваемых бит сообщения порядка 0,06 (для большинства классических стегоалгоритмов этот показатель находится на уровне от 0,5 и выше), что положительным образом отражается на минимизации итоговых искажений стегоконтейнеров.

Отличительной особенностью предложенного варианта мультиплексирования скрытых каналов является использование одного и того же подмножества элементов, подвергающихся одинаковым искажениям при встраивании различных сообщений.

Стоит отметить, что алгоритм встраивания очень хорошо подходит для распараллеливания. Все слова скрываемого сообщения могут встраиваться в соот-

ветствующие им блоки кадра независимо друг от друга. В режиме мультиплексирования скрытых каналов зависимыми будут лишь пары одновременно искажаемых блоков.

В числе возможных направлений развития данного алгоритма можно отметить изучение возможностей использования GPU для ускорения вычислений при работе со словами разрядностью более 16 бит, а также подробное изучение вопросов устойчивости алгоритма к современным методам стегоанализа.

References

[1] Holub V, Fridrich J. Designing steganographic distortion using directional filters. 2012 IEEE Int Workshop on Information Forensics and Security (WIFS) 2012: 234-239. DOI: 10.1109/WIFS.2012.6412655.

[2] Holub V, Fridrich J. Digital image steganography using universal distortion. IH&MMSec '13: Proc 1st ACM Workshop on Information Hiding and Multimedia Security 2013: 59-68. DOI: 10.1145/2482513.2482514.

[3] Pevný T, Filler T, Bas P. Using high-dimensional image models to perform highly undetectable steganography. In Book: Böhme R, Fong PWL, Safavi-Naini R, eds. Information hiding. Berlin, Heidelberg: Springer-Verlag; 2010: 161-177. DOI: 10.1007/978-3-642-16435-4_13.

[4] Nguyen TD, Arch-int S, Arch-int N. An adaptive multi bit-plane image steganography using block data-hiding. Multimed Tools Appl 2015; 75(14): 8319-8345. DOI: 10.1007/s11042-015-2752-9.

[5] Paul G, et al. Keyless dynamic optimal multi-bit image steganography using energetic pixels. Multimed Tools Appl 2017; 76(5): 7445-7471. DOI: 10.1007/s11042-016-3319-0.

[6] Fridrich J, Goljan M. Digital image steganography using stochastic modulation. Proc SPIE 2003; 5020: 191-202. DOI: 10.1117/12.479739.

[7] Eltisheva EYu, Fionov AI. Creation of a steganographic system for raster images based on the principles of stochastic modulation considering the statistics of the least significant bits [In Russian]. Proc Siberian State University of Telecommunications and Informatics 2011; 2: 63-75.

[8] Sirota AA, Dryuchenko MA, Mitrofanova EYu. Digital watermarking method based on heteroassociative image compression and its realization with artificial neural networks. Computer Optics 2018; 42(3): 483-494. DOI: 10.18287/2412-6179-2018-42-3-483-494.

[9] Swain G. Adaptive pixel value differencing steganography using both vertical and horizontal edges. Multimed Tools Appl 2016; 75: 13541-13556. DOI: 10.1007/s11042-015-2937-2.

[10] Pradhan A, Krovi RS, Swain G. Adaptive PVD steganography using horizontal, vertical, and diagonal edges in six-pixel blocks. Secur Commun Netw 2017; 2017: 1924618. DOI: 10.1155/2017/1924618.

[11] Zhou Z, Sun H, Harit R, Chen X, Xingming S. Coverless image steganography without embedding. In Book: Huang Z, Sun X, Luo J, Wang J, eds. Cloud computing and security. Switzerland: Springer International Publishing; 2015: 123-132. DOI: 10.1007/978-3-319-27051-7_11.

[12] Zheng Sh, Wang L, Ling B, Hu D. Coverless information hiding based on robust image hashing. In Book: Huang D-S, Hussain A, Han K, Gromiha MM, eds. Intelligent computing methodologies. Cham: Springer International Publishing AG; 2017: 536-547. DOI: 10.1007/978-3-319-63315-2_47.

[13] Liu Q, Xiang X, Qin J, Tan Y, Tan J, Luo Y. Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping. Knowledge-Based Systems 2020; 192: 105375. DOI: 10.1016/j.knosys.2019.105375.

[14] Zou L, Sun J, Gao M, Wan W, Gupta BB. A novel coverless information hiding method based on the average pixel value of the sub-images. Multimed Tools Appl 2019; 78: 7965-7980. DOI: 10.1007/s11042-018-6444-0.

[15] Kozachok AV, Kopylov SA, Meshcheryakov RV, Evsutin OO. Review of the current methods for robust image hashing. Computer Optics 2017; 41(5): 743-755. DOI: 10.18287/2412-6179-2017-41-5-743-755.

[16] Hu D, Wang L, Jiang W, Zheng S, Li B. A novel image steganography method via deep convolutional generative adversarial networks. IEEE Access 2018; 6: 38303-38314. DOI: 10.1109/ACCESS.2018.2852771.

[17] Ping W, Sheng L, Xinpeng Z, Ge L, Zhenxing Q, Qing Z. Generative steganography network. MM '22: Proc 30th ACM Int Conf on Multimedia 2022: 1621-1629. DOI: 10.1145/3503161.3548217.

[18] Dryuchenko MA. Steganographic algorithm for data hiding in jpeg images based on the use of convolution functions [In Russian]. Proc Voronezh State University: Systems analysis and information technologies 2018; 3: 93-102. DOI: 10.17308/sait.2018.3/1235.

[19] Aappleby/smhasher. Source: <https://github.com/aappleby/smhasher/>.

[20] The 512×512 PPG-LIRMM-COLOR Database. Source: <https://www.lirmm.fr/~chaumont/PPG-LIRMM-COLOR.html>.

[21] Yedroudj M, Comby F, Chaumont M. Yedrouj-Net: An efficient CNN for spatial steganalysis. arXiv Preprint. 2018. Source: <https://arxiv.org/abs/1803.00407>. DOI: 10.48550/arXiv.1803.00407.

Сведения об авторе

Дрюченко Михаил Анатольевич, 1985 года рождения, в 2007 году окончил Воронежский государственный университет по специальности «Прикладная математика и информатика». Доцент кафедры технологий обработки и защиты информации Воронежского государственного университета. Область научных интересов: компьютерная стеганография и стегоанализ, компьютерная обработка изображений, программирование.
E-mail: *m_dryuchenko@mail.ru* .

ГРНТИ: 81.93.29
Поступила в редакцию 18 августа 2022 г. Окончательный вариант – 6 ноября 2022 г.

An adaptive image steganography algorithm based on the use of non-cryptographic hash functions for data extraction

M.A. Dryuchenko [1]
[1] *Voronezh State University, 394018, Voronezh, Russia, Universitetskaya pl.1*

Abstract

An adaptive steganography algorithm for data hiding in full-color images based on iterative introduction of minor distortions into blocks of container images and the use of high-speed non-cryptographic hash functions for data extraction is considered. Modification of the minimum number of container elements compared to the length of the hidden message is a distinctive feature of the algorithm. This feature allows the hidden throughput to be increased and the visual and statistical visibility of hidden data to be reduced. The algorithm is compared with modern algorithms of adaptive spatial steganography in terms of assessing the level of distorting changes in stego-containers. In addition, a modified version of the algorithm that implements covert channel multiplexing using a common subset of container elements when embedding various messages into them is considered.

Keywords: steganography, non-cryptographic hash codes, cyclic redundancy codes.

Citation: Dryuchenko MA. An adaptive image steganography algorithm based on the use of non-cryptographic hash functions for data extraction. Computer Optics 2023; 47(3): 415-425. DOI: 10.18287/2412-6179-CO-1215.

Author's information

Mikhail Anatolievich Dryuchenko (b. 1985) graduated from Voronezh State University in 2007, majoring in Applied Mathematics and Informatics. Currently docent of Information Processing and Security Technologies department at Voronezh State University. Research interests: steganography and steganalysis, computer graphics processing, programming. E-mail: *m_dryuchenko@mail.ru* .

Received August 18, 2022. The final version – November 6, 2022.