АЛГОРИТМЫ ОБРАБОТКИ ЦИФРОВОГО ВОДЯНОГО ЗНАКА ДЛЯ ЗАЩИТЫ АВТОРСКИХ ПРАВ НА ГРАФИЧЕСКИЕ ФАЙЛЫ

Сысоев В. В.[1], Быков А. Ю.[2]

DOI: 10.21681/2311-3456-2025-5-139-148

Цель исследования: *создание стеганографических алгоритмов, стойких к различным видам воздействий, для встраивания цифрового водяного знака в графический файл и извлечения его.*

Методы исследования: *быстрое дискретное преобразование Фурье, алгоритм Блюстейна, действия над комплексными числами, действия над матрицами.*

Результат исследования: *описаны этапы работы алгоритма создания стеганограммы на основе графического файла-контейнера, содержащей цифровой водяной знак (ЦВЗ), а также алгоритма извлечения ЦВЗ из созданной стеганограммы. При встраивании ЦВЗ в изображение было использовано быстрое дискретное преобразование Фурье по алгоритму Блюстейна, встраивание рекомендовано в область спектра низких или средних частот. Проведено моделирование работы алгоритма с помощью данных, полученных на имитаторе. При моделировании показаны этапы работы алгоритма и продемонстрированы особенности изменения данных контейнера и ЦВЗ при встраивании и извлечении. Представлены примеры работы алгоритмов встраивания и извлечения ЦВЗ. Проведено исследование зависимости изменений исходного контейнера от размера ЦВЗ после его встраивания и сравнение нового алгоритма с популярными стеганографическими алгоритмами. Показана стойкость алгоритма к различным видам воздействий на стеганограмму.*

Научная новизна: *заключается в разработке стеганографического алгоритма, пригодного для защиты авторского права на изображения, на основе встраивания ЦВЗ в яркостную составляющую цвета графического файла-контейнера с использованием быстрого преобразования Фурье.*

Ключевые слова: *стеганография, цифровое графическое изображение, файл-контейнер, матрица пикселей.*

Введение

Проблема защиты авторских или эксклюзивных прав на информацию является актуальной в коммерческих и государственных организациях в силу ценности как самой информации, так и прямым требованием действующего законодательства РФ. Несанкционированное/неправомерное применение данной информации может приводить для организаций как к репутационному ущербу, так и к прямым финансовым убыткам.

Предлагаемые в статье алгоритмы делают возможным защиту цифрового изображения графического файла путем встраивания в него ЦВЗ и последующей проверки ЦВЗ. Алгоритм встраивания является стойким к различным видам воздействий, при этом изображение после встраивания остается визуально не отличимым для человека от оригинального изображения. Основы предложенных алгоритмов встраивания и проверки ЦВЗ рассмотрены в [1], ниже алгоритмы описаны более подробно, а также приведены обоснование работы алгоритмов и результаты экспериментов.

Одна из задач, поставленная в рамках этой статьи, – создание простого, незаметного и надежного способа внедрения ЦВЗ в цифровое изображение. В конечном счете, результатом, достигаемым при решении вышеуказанной задачи, является обеспечение возможности автоматизированного формирования и внедрения в цифровое изображение цифровой метки, содержащей информацию, необходимую для проведения расследования и установления обстоятельств в случае несанкционированного использования графических материалов сторонними лицами.

Другая задача, которую нужно было решить в рамках данной статьи, – создание алгоритма для получения стеганограммы, стойкой к изменениям формата изображения, обрезанию изображения, сжатию, обработке шумами. Желательно, чтобы стеганограмма, созданная алгоритмом, имела стойкость ко всем перечисленным возможным изменениям одновременно [2].

В настоящий момент наиболее продвинутыми алгоритмами для встраивания ЦВЗ являются алгоритмы, основанные на методе NeRF [3]. Данный метод имеет много вариаций, например, IW-NeRF [4], RWNeRF [5], CopyNerRF [6], StegaNeRF [7]. Общий

1 Сысоев Валентин Валерьевич, аспирант кафедры «Информационная безопасность», МГТУ им. Н. Э. Баумана, г. Москва, Россия. E-mail: valsus88@mail.ru
2 Быков Александр Юрьевич, кандидат технических наук, доцент, доцент кафедры «Информационная безопасность», МГТУ им. Н. Э. Баумана, г. Москва, Россия. E-mail: abykov@bmstu.ru

принцип работы вышеназванных алгоритмов основан на том, что из объекта извлекаются его цветовые параметры, на которых обучается нейронная сеть MLP (MultiLayer Perceptron). Ключ, который представляет собой встраиваемый ЦВЗ у каждого алгоритма, создается и встраивается по его оригинальному методу. Достоинства алгоритмов семейства NeRF заключается в том, что они применимы как для 2D, так и 3D объектов и обладают относительно хорошей помехоустойчивостью. Недостатки вытекают из особенностей метода. Так как метод использует цветовые параметры объекта, при внесении в него значительных изменений информация о ключе теряется. Кроме того, создание пар «объект – нейросеть», вычислительно затратно, поскольку требует значительные вычислительные мощности для создания нейросети.

Поэтому необходим такой алгоритм, который бы, с одной стороны, работал по классическому принципу внесения ЦВЗ в изображение, но в то же время работал с цветовым представлением всего объекта, а не какой-то его части, так как это снижает эффективность инструментов стеганоанализа [8].

Одна из особенностей алгоритма, приведенного в данной статье в том, что он использует быстрое дискретное преобразование Фурье (ДПФ) в процессе создания стеганограммы. Быстрое ДПФ вычисляется посредством алгоритма Блюстейна[3]. Алгоритм Блюстейна отличается от других алгоритмов быстрого ДПФ тем, что является более гибким и быстрым алгоритмом, в котором нужно проводить меньше вычислений[4]. Алгоритм Блюстейна имеет ту же асимптотическую сложность, что и все алгоритмы быстрого ДПФ: $O(n\ \log(n))$, выбор его использования обусловлен более простой реализацией, которая позволяет задействовать его на устройствах с незначительными вычислительными ресурсами.

Имеются так же алгоритмы, схожие с предложенным, основанные на вейвлет-преобразовании [9]. Однако данная реализация сложнее, имеет сложность $O(n^2)$ и потребляет больше памяти. Другие особенности стеганографических алгоритмов рассмотрены в [10].

Математическая постановка задачи

В рамках данной статьи рассматриваются две задачи – внедрение цифрового водяного знака (ЦВЗ) в контейнер (создание стеганограммы) и извлечение из стеганограммы внедренного ЦВЗ.

3 Блеихут Р. Быстрые алгоритмы цифровой обработки сигналов. - М.: Мир, 1989. 448 с.

4 Sirin S. CZT vs FFT: Flexibility vs Speed. Embedded Systems Programming magazine. 2003. DOI: https://www.researchgate.net/publication/241978861_CZT_vs_FFT_Flexibility_vs_Speed.

1.1. Внедрение цифрового водяного знака в контейнер

Дано следующее:

Изображение C – контейнер в произвольном формате размерностью $h \times w$, где w – ширина изображения в пикселях, h – высота изображения в пикселях. Таким образом, что изображение C можно изобразить матрицей пикселей

$$C_{h\times w} = \begin{pmatrix} p_{11}^{(C)} & \cdots & p_{1w}^{(C)} \\ \vdots & \ddots & \vdots \\ p_{h1}^{(C)} & \cdots & p_{hw}^{(C)} \end{pmatrix},$$

где p_{ij} – пиксель изображения в цветовом пространстве RGB, характеризирующийся тремя цветовыми спектрами p_{ij} (r_{ij},g_{ij},b_{ij}), где r_{ij}, g_{ij}, b_{ij} – интенсивности составляющих трех цветов (красный, зеленый, синий).

Сообщение M в текстово-числовом виде предстоит внедрить в контейнер. Сообщение обычно является идентификатором автора (цифровым водяным знаком) для подтверждения авторского права.

Необходимо найти:

Стеганограмму CM, представляющую собой изображение в цветовом пространстве RGB, той же размерности $h \times w$, со встроенным сообщением M, визуально неотличимое от изображения C. Такое, что

$$CM_{h\times w} = \begin{pmatrix} p_{11}^{(CM)} & \cdots & p_{1w}^{(CM)} \\ \vdots & \ddots & \vdots \\ p_{h1}^{(CM)} & \cdots & p_{hw}^{(CM)} \end{pmatrix}.$$

1.2. Извлечение ЦВЗ из стеганоконтейнера

Дано следующее:

Изображение в цветовом пространстве RGB, являющееся стеганограммой со встроенным сообщением M, заданное матрицей $CM_{h\times w}$.

Необходимо найти: сообщение M в текстово-числовом виде.

2. Описание алгоритмов

2.1. Предварительные алгоритмы

Взаимная конверсия изображений из пикселей формата RGB в формат HSV и обратно.

По пиксельный перевод изображения из формата RGB в формат HSV обусловлен тем, что будем размещать метку не в каком-то одном цветовом диапазоне, а во всех диапазонах сразу, это повысит ее стойкость.

Пусть изображение Im представляет собой матрицу пикселей размерностью $h \times w$, где w – ширина изображения в пикселях, h – высота изображения в пикселях

$$Im_{h\times w} = \begin{pmatrix} p_{11}^{(Im)}(r_{11},g_{11},b_{11}) & \cdots & p_{1w}^{(Im)}(r_{1w},g_{1w},b_{1w}) \\ \vdots & \ddots & \vdots \\ p_{h1}^{(Im)}(r_{h1},g_{h1},b_{h1}) & \cdots & p_{hw}^{(Im)}(r_{hw},g_{hw},b_{hw}) \end{pmatrix}.$$

Для каждого пикселя $p_{ij}^{(Im)}(r_{ij},g_{ij},b_{ij})$ выполним приведение к виду $p_{ij}^{(Im')}(h_{ij},s_{ij},v_{ij})$ по общеизвестным соотношениям, где h_{ij}, s_{ij}, v_{ij} – составляющие тона, насыщенности и яркости

$$Im_{h\times w}=\begin{pmatrix} p_{11}^{(Im')}(h_{11},s_{11},v_{11}) & \cdots & p_{1w}^{(Im')}(h_{1w},s_{1w},v_{1w}) \\ \vdots & \ddots & \vdots \\ p_{h1}^{(Im')}(h_{h1},s_{h1},v_{h1}) & \cdots & p_{hw}^{(Im')}(h_{hw},s_{hw},v_{hw}) \end{pmatrix}.$$

Перевод матрицы $Imp'_{h\times w}$ изображения Im' из цветового пространства HSV в цветовое пространство RGB выполняется также по известным преобразованиям.

2.2. Преобразование Блюстейна

Основная идея вычислений заключается в замене прямого вычисления дискретного преобразования Фурье через использование свёрток. Вместо непосредственного вычисления суммы произведений, алгоритм Блюстейна преобразует её в форму, удобную для вычислений методом быстрого ДПФ. Используя за основу формулы из [8], составим формулы, наиболее применимые для нашего случая.

Пусть $X = |x_0,x_1,\ldots,x_{N-1}|$ — это входная последовательность (дискретная функция) длины N. Тогда выходная последовательность комплексных чисел $Y = |y_0,y_1,\ldots,y_{N-1}|$ такой же длины, вычисленная по ДПФ, определяется следующим образом:

$$y_k=\sum_{n=0}^{N-1} x_n\, e^{-i\frac{2\pi}{N}kn}, \qquad (1)$$

где $k = 0, 1, \ldots, N-1$ – индексы выходной последовательности.

В общем виде можно записать $Y = F(X)$, где F – оператор дискретного преобразования Фурье. Вычисление последовательности по алгоритму Блюстейна $Y = CZT(Y)$ может быть выражено через сумму свёрток, умноженную на N фазовых коэффициентов, и представлено в следующем виде:

$$y_k = b_k^{*}\sum_{n=0}^{N-1} a_n \times b_{|k-n|},\ k = 0, 1, \ldots, N-1, \qquad (2)$$

где $a_n = x_n\, e^{-i\frac{\pi}{N}n^2}$, $b_n = e^{-i\frac{\pi}{N}n^2}$, $b_k^{*} = e^{-i\frac{\pi}{N}k^2}$, $*$ – обозначение свертки.

Обратное преобразование Фурье $X' = FCZT(Y)$ вычисляется по следующей формуле:

$$x'_k=\frac{1}{N}\sum_{n=0}^{N-1} y_j\, e^{i\frac{2\pi}{N}kj},\ k = 0, 1, \ldots, N-1. \qquad (3)$$

В общем виде $X' = \mathcal{F}^{-1}Y$, где $\mathcal{F}^{-1}$ – оператор обратного дискретного преобразования Фурье.

Причина использования преобразования Блюстейна заключается в том, что оно удобно для вычисления дискретного преобразования Фурье любой входной длины. Особенность преобразования в том, что оно позволяет в уравнении свертки заполнять, при выполнении определенных условий, последовательность a_n нулями, тем самым, упрощая вычисление.

2.3. Внедрение ЦВЗ в контейнер

Алгоритм делится на несколько этапов:

ЭТАП I:

Этап I представляет собой предварительные действия над сообщением и контейнером, в который нужно внедрить сообщение.

Цепочка А – представляет собой действия над сообщением, которое нужно внедрить в контейнер.

1А: Сообщение из текстово-числового вида преобразуется в метку – изображение M произвольного размера $k \times l$, где l – ширина изображения метки, k – высота изображения метки,

$$k < h, l < w\colon M_{k\times l}=\begin{pmatrix} p_{11}^{(M)} & \cdots & p_{1l}^{(M)} \\ \vdots & \ddots & \vdots \\ p_{k1}^{(M)} & \cdots & p_{kl}^{(M)} \end{pmatrix}.$$

Метка формируется по следующему правилу:

Фон метки черный, т.е. всем пикселям $p_{ij}^{(M)}$ метки M устанавливаются значения $p_{ij}^{(M)}(0,0,0)$.

В произвольном месте метки наносится изображение, аналогичное графическому образу сообщения M, т.е. пиксели устанавливаются в белый цвет, $p_{ij}^{(M)}(1,1,1)$, если использовать интервал для задания цвета $[0, 1]$.

2А: Осуществляется попиксельный перевод изображения M из цветового пространства RGB в цветовое пространство HSV,

$$M'_{k\times l}=\begin{pmatrix} p_{11}^{(M')}(h_{11},s_{11},v_{11}) & \cdots & p_{1l}^{(M')}(h_{1l},s_{1l},v_{1l}) \\ \vdots & \ddots & \vdots \\ p_{k1}^{(M')}(h_{k1},s_{k1},v_{k1}) & \cdots & p_{kl}^{(M')}(h_{kl},s_{kl},v_{kl}) \end{pmatrix}.$$

3А: Из каждого пикселя $p_{ij}^{(M')}$ матрицы $M'_{k\times l}$ извлекаем соответствующую компоненту яркости v_{ij} и составляем соответствующую матрицу яркости метки

$$V_{k\times l}^{(M')}=\begin{pmatrix} v_{11}^{(M)} & \cdots & v_{1l}^{(M)} \\ \vdots & \ddots & \vdots \\ v_{k1}^{(M)} & \cdots & v_{kl}^{(M)} \end{pmatrix}.$$

Эта матрица состоит из 0 и 1 для черных и белых пикселей, соответственно.

Выбор значения яростной характеристики для кодирования метки обусловлено тем, что яркостная характеристика, при переводе из одного одной цветовой модели в другую, меньше всего подвергается математическим преобразованиям, при этом она покрывает все изображение, является подобием математического ожидания для распределения интенсивностей цветов в каждом пикселе изображения.

Также не маловажно, что изменение яркостной характеристики даже всего изображения на 2–4 %[5] остается незаметным для наблюдателя, что так же подтверждается и другими исследованиями [11].

5 David H. Hubel. Eye, Brain, and Vision. - Henry Holt and Company, 1995. 242 p.

Цепочка Б – представляет собой действия над контейнером.

1Б: В изображении – контейнере C переводим каждый пиксель из цветового пространства RGB в цветовое пространство HSV. Для каждого пикселя RGB $p_{ij}^{(C)}(r_{ij},g_{ij},b_{ij})$ получаем пиксель в формате HSV $p_{ij}^{(C)}(h_{ij},s_{ij},v_{ij})$. Получаем матрицу

$$C'_{h\times w}=\begin{pmatrix} p_{11}^{(C')} & \cdots & p_{1w}^{(C')} \\ \vdots & \ddots & \vdots \\ p_{h1}^{(C')} & \cdots & p_{hw}^{(C')} \end{pmatrix}.$$

Для краткости (r_{ij},g_{ij},b_{ij}) и (h_{ij},s_{ij},v_{ij}) далее опускаем.

2Б: Из каждого пикселя $p_{ij}^{(C')}$ матрицы C' извлекаем соответствующую компоненту яркости v_{ij} и составляем соответствующую матрицу яркости контейнера C':

$$V_{h\times w}^{(C')}=\begin{pmatrix} v_{11}^{(C')} & \cdots & v_{1w}^{(C')} \\ \vdots & \ddots & \vdots \\ v_{h1}^{(C')} & \cdots & v_{hw}^{(C')} \end{pmatrix}.$$

3Б: матрицу яркости контейнера $V_{h\times w}^{(C')}$ переводим в матрицу-строку $V_{hw}^{(C')}$ путем соединения всех h строк матрицы $V_{h\times w}^{(C')}$ в одну строку $V_{hw}^{(C')}=(V_1^{(C')}\ldots V_{hw}^{(C')})$, где $v_i\in[0,1]$.

4Б: над матрицей-строкой $V_{hw}^{(C')}$ проводится быстрее преобразование Фурье по формуле Блюстейна (2). В результате получаем: $CZT(V_{hw}^{(C')})=Z_{hw}^{(C')}=(z_1^{(C')}\ldots z_{hw}^{(C')})$, где $z_i^{(C')}\in C$ (множество комплексных чисел).

ЭТАП II:

На этапе II происходит непосредственное внедрение метки в контейнер:

1. Матрица-строка $Z_{hw}^{(C')}$ переводится в матрицу размерности $h\times w$ путем перевода w чисел из $Z_{hw}^{(C')}$ на новую строку:

$$Z_{h\times w}^{(C')}=\begin{pmatrix} z_{11}^{(C')} & \cdots & z_{1w}^{(C')} \\ \vdots & \ddots & \vdots \\ z_{h1}^{(C')} & \cdots & z_{hw}^{(C')} \end{pmatrix}.$$

2. Матрица $Z_{h\times w}^{(C')}$ складывается с матрицей $V_{k\times l}^{(M')}$ по определенным правилам:

$$Z_{h\times w}^{(CM')}=Z_{h\times w}^{(C')}+V_{k\times l}^{(M')}. \quad (4)$$

Складываем матрицы по следующему правилу: в матрице $Z_{h\times w}^{(C')}$ выбираем прямоугольник размерности $k\times l$, например, начиная с элемента $z_{ij}^{(C')}$, таким образом, чтобы не выйти за границы матрицы размерности $h\times w$, как правило, размерность $h\times w$ существенно выше, чем $k\times l$. К элементам этого прямоугольника прибавляем соответствующие элементы матрицы $V_{k\times l}^{(M')}$, в сложении участвуют действительные части комплексных чисел матрицы $Z_{h\times w}^{(C')}$. Выбор индексов i и j элемента, с которого начинается встраивание в матрицу контейнера, предлагается выполнять так, чтобы метка внедрялась в область спектра низких частот, в большей степени, и средних частот – в меньшей [12].

3. Полученную при сложении матрицу $Z_{h\times w}^{(CM')}$ переводим в матрицу-строку $Z_{hw}^{(CM')}$ путем соединения всех w строк матрицы $Z_{h\times w}^{(CM')}$ в одну строку $Z_{hw}^{(CM')}$.

4. Над матрицей-строкой $Z_{hw}^{(CM')}$ проводится обратное дискретное преобразование Фурье в соответствии с (3). В результате получаем матрицу-строку $V_{hw}^{(CM')}$: $FCZT(Z_{hw}^{(CM')})=V_{hw}^{(CM')}=(v_1^{(CM')}\ldots v_{hw}^{(CM')})$.

ЭТАП III:

На этапе III происходит восстановление стеганограммы из матричного вида обратно в визуальный вид.

1. Переведем матрицу-строку $V_{hw}^{(CM')}$ в матрицу размерности $w\times h$ путем перевода строки из h чисел из $V_{hw}^{(CM)}$ на новую строку:

$$V_{h\times w}^{(CM')}=\begin{pmatrix} v_{11}^{(CM')} & \cdots & v_{1w}^{(CM')} \\ \vdots & \ddots & \vdots \\ v_{h1}^{(CM')} & \cdots & v_{hw}^{(CM')} \end{pmatrix}$$

2. В матрице $C'_{h\times w}$ заменим все компоненты яркости v_{ij} в каждом пикселе $p_{ij}^{(C')}$ на компоненты яркости из $V_{h\times w}^{(CM')}$:

$$CM'_{k\times l}=\begin{pmatrix} p_{11}^{(CM')}(h_{11},s_{11},v_{11}) & \cdots & p_{1w}^{(CM')}(h_{1w},s_{1w},v_{1w}) \\ \vdots & \ddots & \vdots \\ p_{h1}^{(CM')}(h_{h1},s_{h1},v_{h1}) & \cdots & p_{hw}^{(CM')}(h_{hw},s_{hw},v_{hw}) \end{pmatrix}.$$

Если компонента $v_{ij}^{(CM')}$ выходит за интервал $[0, 1]$, то значения, меньшие 0, заменяются на 0, а значения, большие 1, заменяются на 1.

3. При необходимости переведем пиксели в матрице $C'_{h\times w}$ из цветового пространства HSV в цветовое пространство RGB по известным преобразованиям, получаем на выходе алгоритма матрицу:

$$CM_{w\times h}=\begin{pmatrix} p_{11}^{(CM)} & \cdots & p_{1w}^{(CM)} \\ \vdots & \ddots & \vdots \\ p_{h1}^{(CM)} & \cdots & p_{hw}^{(CM)} \end{pmatrix}.$$

2.4. Извлечение метки из контейнера

1. Пиксели матрицы $CM_{w\times h}$ переводятся из цветового пространства RGB в цветовое пространство HSV, получаем матрицу $CM'_{h\times w}$.

2. Из матрицы $CM'_{h\times w}$ извлекаем компоненту яркости $v_{ij}^{(CM')}$, создавая таким образом новую матрицу яркостей стегоконтейнера $V_{h\times w}^{(CM')}$.

3. Матрица $V_{h\times w}^{(CM')}$ переводится в матрицу-строку $V_{hw}^{(CM')}$ путем соединения всех строк матрицы.

4. Над матрицей-строкой $V_{hw}^{(CM')}$ проводится быстрее преобразование Фурье по алгоритму Блюстейна в соответствии с (2). В результате получаем матрицу-строку $Z_{hw}^{(CM')}$ комплексных чисел.

5. Матрицу-строку $Z_{hw}^{(CM')}$ переводят в матрицу $Z_{h\times w}^{(CM')}$ размерности $h\times w$ путем перевода каждых h элементов матрицы из $Z_{wh}^{(CM')}$ на новую строку:

6. Из матрицы $Z_{wh}^{(CM')}$ составляется матрица $D_{wh}^{(CM')}$ путем выделения действительной части чисел $z_{ij}^{(CM')}$,

$$D_{w\times h}^{(CM')} = \begin{pmatrix} d_{11}^{(CM')} & \cdots & d_{1w}^{(CM')} \\ \vdots & \ddots & \vdots \\ d_{h1}^{(CM')} & \cdots & d_{hw}^{(CM')} \end{pmatrix}.$$

7. Создаем матрицу изображения извлеченной метки из контейнера $MC'_{w\times h}$, в которой пиксели заданы в формате HSV, значения h и s в которой выставлены в 0, а значение v взято из матрицы $D_{w\times h}^{(CM')}$, при этом, если компонента $d_{ij}^{(CM')}$ выходит за интервал $[0, 1]$, то значения, меньшие 0, заменяются на 0, а значения, большие 1 заменяются на 1,

$$MC'_{h\times w} = \begin{pmatrix} p_{11}^{(MC)}(0,0,d_{11}^{(CM')}) & \cdots & p_{1w}^{(MC)}(0,0,d_{1w}^{(CM')}) \\ \vdots & \ddots & \vdots \\ p_{h1}^{(MC)}(0,0,d_{h1}^{(CM')}) & \cdots & p_{hw}^{(MC)}(0,0,d_{hw}^{(CM')}) \end{pmatrix}.$$

8. Переводим пиксели матрицы $MC'_{h\times w}$ из формата HSV в формат RGB и получаем таким образом визуальное изображение $MC_{h\times w}$.

9. По визуальному изображению находим искомое сообщение M.

2.5. Обоснование работы алгоритма

Рассмотрим пример работы алгоритмов встраивания и извлечения на имитаторе исходных данных. По сути, алгоритм встраивает данные (вектор из 0 и 1, аналог метки) в матрицу-строку (вектор) $V_{hw}^{(C')}$. Для имитации был сгенерирован вектор исходных данных для встраивания из 200 вещественных чисел в интервале $[0, 1]$, для заполнения использовался следующий код на языке C++:

```
size_t N=200;
vector<double> c(N); // Контейнер
for(size_t i=1; i<=N; i++) c[i-1]=sin(i)/2+0.5;
```

Метка представляла из себя вектор такой же размерности, в котором все 0, в начале несколько элементов 1. Результаты имитации встраивания (табл. 1).

Таблица 1.

Результаты имитации встраивания данных

Индекс элемента	Данные контейнера	Действительная часть после прямого ДПФ	Встраиваемая метка	Действительная часть после прямого ДПФ с меткой	Результат обратного ДПФ (контейнер с меткой)	Результат прямого ДПФ для извлечения метки
1	2	3	4	5	6	7
0	0,9207	100,0163	0,0000	100,0163	0,9407	1,0000
1	0,9546	0,0161	0,0000	0,0161	0,9745	0,0135
2	0,5706	0,0154	1,0000	1,0154	0,5900	0,4935
3	0,1216	0,0143	1,0000	1,0143	0,1404	0,4915
4	0,0205	0,0127	1,0000	1,0127	0,0385	0,5054
5	0,3603	0,0106	1,0000	1,0106	0,3771	0,5026
6	0,8285	0,0080	0,0000	0,0080	0,8439	0,0220
7	0,9947	0,0048	0,0000	0,0048	1,0000	0,0209
8	0,7061	0,0010	0,0000	0,0010	0,7183	0,0027
9	0,2280	–0,0035	0,0000	–0,0035	0,2384	0,0000
...						
190	0,7975	–0,0086	0,0000	–0,0086	0,8060	0,0000
191	0,3225	–0,0035	0,0000	–0,0035	0,3330	0,0000
192	0,0108	0,0010	0,0000	0,0010	0,0230	0,0027
193	0,1488	0,0048	0,0000	0,0048	0,1627	0,0209
194	0,6097	0,0080	0,0000	0,0080	0,6252	0,0220
195	0,9698	0,0106	0,0000	0,0106	0,9866	0,5026
196	0,8979	0,0127	0,0000	0,0127	0,9158	0,5054
197	0,4602	0,0143	0,0000	0,0143	0,4790	0,4915
198	0,0591	0,0154	0,0000	0,0154	0,0786	0,4935
199	0,0634	0,0161	0,0000	0,0161	0,0832	0,0135

Рис. 1. Исходный контейнер и он же со встроенной стеганограммой

В таблице 1 представлены первые и последние 10 элементов вектора, на которых показано встраивание и извлечение метки. В первом столбце заданы индексы значений вектора данных, во втором – сами данные, полученные имитатором. В третьем столбце – действительные части комплексных значений, полученных после прямого ДПФ. В четвертом столбце – данные вектора встраиваемой метки, в векторе метки элементы, равные 1, имеют индексы от 2 до 5, остальные равны 0. В пятом столбце – результат сложения третьего и четвертого столбцов, аналог выполнения операции, заданной в (4). Элементы, соответствующие значениям единиц в векторе метки, выделены цветом фона в таблице.

Следует отметить, что при сложении значений единиц из метки они складываются со значениями из третьего столбца по модулю существенно меньше 1, и большинство значений в третьем столбце, кроме константы по индексу 0, по модулю меньше единицы. Также вторая половина из третьего столбца повторяют первую, кроме константы по индексу 0, если смотреть в обратном порядке, это свойство прямого ДПФ.

После обратного ДПФ значений пятого столбца получаем имитацию стенограммы с меткой, данные ее в столбце 6. Эти данные близки с исходными данными второго столбца, они бы совпали, если бы не прибавили к данным метку. Изменения, внесённые меткой в отдельные значения, распределяются по всем данным шестого столбца. В седьмом столбце представлены данные, полученные после прямого ДПФ, для извлечения метки. Если бы данные пятого столбца не были бы изменены меткой, то седьмой столбец совпал бы с пятым. При этом элементы седьмого столбца с индексами, в которых у вектора-метки находятся 1, существенно больше соседних элементов этого столбца, что позволяет визуально отличать метку. Учитывая то, что при прямом ДПФ вторая половина полученных данных, совпадает с первой, то происходит дублирование метки во второй половине – элементы с индексами 195–198.

3. Эксперименты

3.1. Примеры работы алгоритмов

Рассмотрим результаты работы алгоритма на примере рисунка 1024 x 1024 пикселей (рис. 1).

В контейнер встроим сообщение M: «Авторское право». Визуализация матрицы после прямого ДПФ со встроенной меткой – матрица действительных частей из $Z_{h\times w}^{(CM')}$ (рис. 2).

Рис. 2. Визуализация контейнера с меткой после обратного ДПФ

Визуализация матрицы предполагает, что по ней создаем в начале изображение в формате HSV, где элементы матрицы – это значения яркости, причем, если значение элемента выходит за пределы $[0, 1]$, то значения, меньшие 0, заменяются на 0, а больше 1 – на 1, значения тона и насыщенности равны 0, при необходимости переводим в формат RGB. Как видно, метка хорошо выделяется, так как остальные пиксели в основном черные (значения яркости близки к 0), идет небольшая засветка по краям контейнера.

Далее после обратного преобразования получаем контейнер с меткой, он визуально для человека

аналогичен (рис. 1). Для извлечения метки выполняем прямое ДПФ, получаем матрицу $MC'_{h\times w}$, ей соответствует изображение (рис. 3).

Рис.3. Изображение с извлеченной меткой

Из рисунка видно, что несмотря на то, что яркость метки уменьшилась, она вполне читаема, также произошло дублирование метки во второй половине контейнера, если смотреть в обратном порядке, что является свойством прямого ДПФ.

3.2. Оценка степени изменения исходного контейнера при встраивании цифрового водяного знака

Рассмотрим оценку изменения исходного контейнера при встраивании в него ЦВЗ в зависимости от размера ЦВЗ. Размер ЦВЗ знака будем измерять отношением числа единиц в метке или изменяемых пикселей при встраивании к числу пикселей в контейнере, заданных в процентах. Изменение в контейнере будем измерять средней квадратичной ошибкой на один пиксель, вычисляемой по формуле:

$$E = \sqrt{\frac{\sum_{i=1}^{h}\sum_{j=1}^{w}(v_{ij}^{(C')} - v_{ij}^{(CM')})^2}{hw}}. \quad (5)$$

График зависимости этой ошибки от числа измененных пикселей в процентах (рис. 4). Увеличение измененных пикселей производится путем добавления символов к встраиваемой метке, измерения проводилось на контейнере (рис. 1).

Рис. 4. Зависимость ошибки от числа измененных пикселей

Из рисунка 4 можно сделать вывод о том, что зависимость изменения в контейнере от размера метки, скорее всего, является линейной или близкой к линейной.

3.3. Сравнение с другими алгоритмами

Сравнение предложенного алгоритма (в табл. 2 заголовок – алгоритм) производилось со следующими популярными стеганографическими алгоритмами: LSB [13], F3 [14], Jsteg [15], дифференциально яркостной ячеистый алгоритм (Digital Brightness Cell Algoritm, DBCA) [16], Куттера [17], Patchwork [18], преобразования Фурье в синем спектре XYZ (FXYZ).

В качестве оценки выбралось способность алгоритмов противостоять различным видам воздействий на стеганограмму. В поле стойкость ставится:

«+» – если стеганограмма, полученная данным алгоритмом, выдержала воздействие, метка была успешно извлечена.

«–» – если стеганограмма, полученная данным алгоритмом, не выдержала воздействие, метка не была успешно извлечена.

«∓» – если стеганограмма, полученная данным алгоритмом, выдержала воздействие, но при извлечении

Таблица 2.

Результаты стойкости стеганографических алгоритмов к различным преобразованиям

№ п/п	Преобразования стеганограммы	Стеганографические алгоритмы							
		LSB	F3	Jsteg	DBCA	Куттера	Patchwork	FXYZ	Алгоритм
1	Конверсия изображения в другие форматы	+	–	–	+	∓	∓	+	+
2	Вырезание частей	–	–	–	+	∓	∓	+	+
3	Обрезание части	–	–	–	+	∓	∓	-	+
4	Ухудшение качества путем сжатия	–	–	–	–	∓	∓	+	+
5	Изменение характеристик	–	–	–	–	∓	∓	–	+
6	Наложение шума	–	–	–	–	∓	∓	–	+

получились неясные результаты, метка была извлечена частично, или в некоторых изображениях она извлекается полностью, в некоторых не извлекается вообще.

Были произведены эксперименты над 10-ю изображениями с различными тематическими содержаниями. В результате получили следующие результаты (табл. 2):

Особенности преобразований стеганограммы:

1. Конверсия исходного изображения стеганограммы в другие форматы.

Перевод изображения из одного формата в другие, например, исходное изображение в формате .png в формат .jpg.

2. Вырезание частей из стеганограммы.

Изменение представляет собой вырезание части, изображения стеганограммы, но не менее 1/4 от нее, метка находится в вырезанной части. В дальнейшем предпринимается попытка извлечь из вырезанной части изображения скрытую метку.

3. Обрезание части стеганограммы.

Изменение похоже на предыдущее, но в данном случае ведется обрезание части изображения с двух сторон стеганограммы (например, обрезание изображения для перехода в формат от 16:10 к 16:9) так, чтобы от полученной стеганограммы осталось не менее 1/8 от нее оригинала, метка находится в оставшейся части. В дальнейшем предпринимается попытка извлечь из оставшейся части изображения скрытую метку.

4. Ухудшение качества стеганограммы путем сжатия изображения.

Стеганограмма подвергается алгоритмам сжатия. В дальнейшем предпринимается попытка извлечь скрытую метку.

5. Изменение характеристик стеганограммы.

В стеганограмме увеличивается и уменьшается яркость и/или контраст изображения. В дальнейшем предпринимается попытка извлечь скрытую метку.

6. Наложение шума на стеганограмму.

На стеганографию наносится цифровой шум – намеренно искажаются хаотично разбросанные пиксели. В дальнейшем предпринимается попытка извлечь скрытую метку.

Выводы

В статье был предложен алгоритм встраивания ЦВЗ в графический файл на основе быстрого ДПФ по алгоритму Блюстейна, а также был приведен алгоритм извлечения ЦВЗ из созданного контейнера.

Продемонстрированы особенности работы алгоритма на данных, полученных с помощью имитатора. Представлены результаты работы описанного алгоритма на примере встраивания и извлечения ЦВЗ в контейнер.

Проведено исследование зависимости изменений в контейнере при встраивании от размера ЦВЗ (отношение числа изменяемых пикселей при встраивании к числу пикселей в контейнере). Изменение контейнера измерялось средней квадратической ошибкой на один пиксель после встраивания ЦВЗ. По результатам эксперимента можно сделать вывод о том, что эта зависимость является близкой к линейной.

Проведено сравнение предложенного алгоритма с семью популярными стеганографическими алгоритмами. Показана стойкость алгоритма к шести различным видам изменений стеганограммы.

Работоспособность предложенных алгоритмов получила практическое подтверждение на проведенных экспериментах. Достоверность предлагаемых научных решений подтверждена анализом свойств прямого и обратного дискретных преобразований Фурье.

Литература

1. Сысоев В. В., Быков А. Ю. Защита авторского права с использованием цифровой голографии // Сборник трудов XIII всероссийской научно-технической конференции «Безопасные информационные технологии» – М.: МГТУ им. Н. Э.Баумана. 2024. С. 207–213.
2. Макаренко С.И. Эталонная модель взаимодействия стеганографических систем и обоснование на ее основе новых направлений развития теории стеганографии // Вопросы кибербезопасности. 2014. № 2(3). С. 24–32.
3. Li D., Yang Z., Jin X. Zero watermarking scheme for 3D triangle mesh model based on global and local geometric features. Multimed. Tools Appl. 2023. Vol. 82. P. 43635–43648.
4. Chen L. and another. IW-NeRF: Using Implicit Watermarks to Protect the Copyright of Neural Radiation Fields. Applied Sciences. 2024., Vol. 14. No 6184. DOI:10.3390/app14146184.
5. Sun W. and another. RWNeRF: Robust Watermarking Scheme for Neural Radiance Fields Based on Invertible Neural Networks. Computers. Materials & Continua. 2024. Vol. 80. P. 4065–4083. DOI:10.32604/cmc.2024.053115.
6. Luo Z., Guo Q., Cheung K.C., See S., Wan R. CopyRNeRF: Protecting the CopyRight of Neural Radiance Fields. 2023. DOI:10.48550/arXiv.2307.11526.
7. Li C., Feng B.Y., Fan Z., Pan P., Wang Z. StegaNeRF: Embedding Invisible Information within Neural Radiance Fields. 2022. DOI:10.48550/arXiv.2212.01602.
8. Сивачев А. В., Прохожев Н. Н., Михайличенко О. В., Башмаков Д. А. Эффективность стеганоанализа на основе методов машинного обучения // Вопросы кибербезопасности. 2017. № 2(20). С. 53–60. DOI 10.21581/2311-3456-2017-2-53-60.

9. Глинская, Е. В., Чичварин Н. В. Информационная безопасность открытых каналов передачи проектной документации, продуцируемой в САПР // Вопросы кибербезопасности. 2014. № 4(7). С. 11–22.
10. Абасова, А. М., Бабенко Л. К. Защита информационного содержания изображений в условиях наличия деструктивного воздействия // Вопросы кибербезопасности. 2019. № 2(30). С. 50–57. DOI 10.21681/2311-3456-2019-2-50-57.
11. Козачок А. В., Копылов С. А., Бочков М. В. Оценка параметров необнаруживаемости разработанного подхода к маркированию текстовых электронных документов // Вопросы кибербезопасности. 2020. № 1(35). С. 62–73. DOI 10.21681/2311-3456-2020-01-62-73.
12. Морковин С. В. Алгоритмы и программные средства человеко-машинной обработки цифровых водяных знаков в видеопоследовательности // Моделирование, оптимизация и информационные технологии. 2022. Т. 10. № 3(38). С. 30–31. DOI: 10.26102/2310-6018/2022.38.3.024.
13. Кривошеев И. А., Линник М. А. Статический способ стеганографического встраивания информации на основе LSB // Системы и средства информатики. 2020. Т. 30, № 3. С. 56–66. DOI 10.14357/08696527200306.
14. Brūzgienė Rasa and another. Enhancing Steganography through Optimized Quantization Tables. Electronics. 2024. Vol.13. No 2415. DOI: 10.3390/electronics13122415.
15. Binmin P., Qiao T., and another. Novel Hidden Bit Location Method towards JPEG Steganography. Security and Communication Networks. 2022. Vol. 2022. No. 8230263. P. 13. DOI: 10.1155/2022/8230263.
16. Крамаренко С. М. Способ внесения цифровых меток в цифровое изображение и устройство для осуществления способа. 2020. Роспатент. RU2739936C1.
17. Zhigalov I. E., Ozerova M. I., Evstigneev A. V. Application of Cutter–Jordan–Bossen method for data hiding in the image spatial area // Вестник Южно-Уральского государственного университета. Серия Компьютерные технологии, управление, радиоэлектроника. 2022. Т. 23. С. 16–23. DOI: 10.14529/ctcr230302.
18. Кушнеревич П. М. Анализ эффективности алгоритмов Patchwork и LSB для защиты графических образов с помощью водяных знаков // Интеграция науки, общества, производства и промышленности: проблемы и перспективы: сборник статей по итогам Международной научно-практической конференции. Волгоград. 2021. С. 121–124.

DIGITAL WATERMARK PROCESSING ALGORITHMS FOR COPYRIGHT PROTECTION OF GRAPHIC FILES

Sysoev V. V.[6], Bykov A. Yu.[7]

Keywords: *steganography, digital graphic image, container file, pixel matrix.*

The purpose of the study: *creation of steganographic algorithms for embedding a digital watermark in a graphic file and extracting it, resistant to various types of influences.*

Methods of research: *fast Fourier transform, Bluestein's algorithm, actions on complex numbers, actions on matrices.*

Result(s): *the stages of the algorithm for creating a steganogram based on a graphic container file containing a digital watermark, as well as an algorithm for extracting a digital watermark from a created steganogram, are described. When embedding a digital watermark in an image, a fast discrete Fourier transform using the Bluestein algorithm was used, embedding is recommended in the low or medium frequency spectrum. The simulation of the algorithm using the data obtained on the simulator is carried out. The simulation shows the stages of the algorithm's operation and demonstrates the features of changing container and data center data during embedding and extraction. Examples of algorithms for embedding and extracting a digital watermark are presented. A study of the dependence of changes in the original container on the size of the digital watermark after its embedding and a comparison of the new algorithm with popular steganographic algorithms has been conducted. The algorithm's resistance to various types of effects on the steganogram is shown.*

Scientific novelty: *the goal is to develop a steganographic algorithm suitable for image copyright protection based on embedding a digital image into the brightness component of the color of a graphic container file using the fast Fourier transform*

.

References

1. Sysoev V. V., Bykov A. Yu. Zashchita avtorskogo prava s ispol'zovaniem cifrovoj [Copyright Protection using Digital Holography]. Sbornik trudov XIII vserossijskoj nauchno-tekhnicheskoj konferencii «Bezopasnye informacionnye tekhnologii». M.: MGTU im. N. E. Baumana. 2024. Pp. 207–213
2. Makarenko, S. I. Etalonnaya model' vzaimodeystviya steganograficheskikh sistem i obosnovanie na ee osnove novykh napravleniy razvitiya teorii steganografii [The Reference Model of the Interaction of Steganographic Systems and the Justification based on it of New Directions in the Development of the Theory of Steganography]. Voprosy kiberbezopasnosti. 2014. No. 2(3). Pp. 24–32.
3. Li D., Yang Z., Jin X. Zero watermarking scheme for 3D triangle mesh model based on global and local geometric features. Multimed. Tools Appl. 2023. Vol. 82. P. 43635–43648.
4. Chen L. et al. IW-NeRF: Using Implicit Watermarks to Protect the Copyright of Neural Radiation Fields. Applied Sciences. 2024. Vol. 14. No. 6184. DOI:10.3390/app14146184.

6 Valentin V. Sysoev, post-graduate student, Department of Information Security, BMSTU, Moscow, Russia. E-mail: valsus88@mail.ru
7 Alexander Yu. Bykov, Ph.D. of technical Sciences, associate professor, Department of Information Security, BMSTU, Moscow, Russia. E-mail: abykov@bmstu.ru

5. Sun W. et al. RWNeRF: Robust Watermarking Scheme for Neural Radiance Fields Based on Invertible Neural Networks. Computers. Materials & Continua. 2024. Vol. 80. Pp. 4065–4083. DOI:10.32604/cmc.2024.053115.
6. Luo Z., Guo Q., Cheung K.C., See S., Wan R. CopyRNeRF: Protecting the CopyRight of Neural Radiance Fields. 2023. DOI:10.48550/arXiv.2307.11526.
7. Li C., Feng B.Y., Fan Z., Pan P., Wang Z. StegaNeRF: Embedding Invisible Information within Neural Radiance Fields. 2022. DOI:10.48550/arXiv.2212.01602.
8. Sivachev A. V., Prokhozhev N. N., Mikhailichenko O. V., Bashmakov D. A. Effektivnost' steganoanaliza na osnove metodov mashinnogo obucheniya [Effectiveness of Steganalysis based on Machine Learning Methods]. Voprosy kiberbezopasnosti. 2017. No. 2(20). Pp. 53–-60. DOI 10.21581/2311-3456-2017-2-53-60.
9. Glinskaya, E. V., Chichvarin N. V. Informatsionnaya bezopasnost' otkrytykh kanalov peredachi proektnoy dokumentatsii, produciruemoy v SAPR [Information Security of open Transmission Channels of Project Documentation produced in CAD]. Voprosy kiberbezopasnosti. 2014. No. 4(7). Pp. 11–22.
10. Abasova, A. M., Babenko L. K. Zashchita informatsionnogo soderzhaniya izobrazheniy v usloviyakh nalichiya destruktivnogo vozdeystviya [Protection of Information Content of Images in the Presence of Destructive Influence]. Voprosy kiberbezopasnosti. 2019. No. 2(30). Pp. 50–57. DOI 10.21681/2311-3456-2019-2-50-57.
11. Kozachok, A. V., Kopylov S. A., Bochkov M. V. Otsenka parametrov neobnaruzhaemosti razrabotannogo podkhoda k markirovaniyu tekstovykh elektronnykh dokumentov [Evaluation of the Undetectability Parameters of the Developed Approach to Labeling Textual Electronic Documents]. Voprosy kiberbezopasnosti. 2020. No. 1(35). Pp. 62–73. DOI 10.21681/2311-3456-2020-01-62-73.
12. Morkovin, S. V. Algoritmy i programmnye sredstva cheloveko-mashinnoy obrabotki tsifrovykh vodyanykh znakov v videoposledovatel'nosti [Algorithms and software tools for human-computer processing of digital watermarks in video sequences]. Modeling, Optimization and Information Technology. 2022. Vol. 10. No. 3(38). Pp. 30–31. DOI: 10.26102/2310-6018/2022.38.3.024.
13. Kryvoshaev I. A., Linnik M. A. Staticheskij sposob steganograficheskogo vstraivaniya informacii na osnove LSB [Static Steganographic Information Embedding Method Based on LSB]. Sistemy i sredstva informatiki [Systems and Means of Informatics]. 2020. Vol. 30. No. 3. P. 56–66. DOI 10.14357/08696527200306.
14. Brūzgienė Rasa et al. Enhancing Steganography Through Optimized Quantization Tables. Electronics. 2024. Vol. 13. No. 2415. DOI: 10.3390/electronics13122415.
15. Binmin P., Qiao T. et al. Novel Hidden Bit Location Method Towards JPEG Steganography. Secur. Commun. Netw. 2022. Vol. 2022. No. 8230263. P. 13. DOI: 10.1155/2022/8230263.
16. Kramarenko S. M. Sposob vneseniya tsifrovykh metok v tsifrovoje izobrazheniye i ustroystvo dlya osushchestvleniya sposoba [Method for embedding digital marks into digital images and device for implementing this method]. 2020. Patent RU2739936C1.
17. Zhigalov I. E., Ozerova M. I., Evstigneev A. V. Application of Cutter-Jordan-Bossen method for data hiding in the image spatial domain. Bulletin of South Ural State University. Series: Computer Technologies, Control, Radioelectronics. 2022. Vol. 23. Pp. 16–23. DOI: 10.14529/ctcr230302.
18. Kushnerevich, P. M. Analiz effektivnosti algoritmov Patchwork i LSB dlya zashchity graficheskikh obrazov s pomoshch'yu vodyanykh znakov [Efficiency Analysis of Patchwork and LSB Algorithms for Graphical Image Protection by Watermarking]. Proceedings of the International Scientific Practical Conference «Integration of Science, Society, Production and Industry: Problems and Prospects». Volgograd. Russia. 2021. Pp. 121–124.