

АЛГОРИТМИЧЕСКИЕ ОСОБЕННОСТИ РЕАЛИЗАЦИИ СТЕГАНОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ WEB-ДОКУМЕНТОВ НА ОСНОВЕ РАСТРОВОЙ ГРАФИКИ И МОДЕЛИ RGB

Оригинальный контент может быть преобразован из одного формата графики в другой, без согласования с авторами. При различных изменениях и преобразованиях текстовых документов (являются контейнерами стеганографической системы) одной из важных проблем является растривание текста: контуры букв становятся нечеткими, а цвет по контуру переходит в градиент [1].

Текстовые документы в растровом формате необходимы для сохранения точного визуального представления контента без возможности его редактирования. Это особенно важно для официальных документов, сертификатов и договоров. Многие организации требуют предоставления документов именно в растровом виде, особенно когда речь идет об отсканированных подписях или юридически значимых копиях.

Формат PNG характеризуется (в сравнении, например, с JPG, TIFF, BNP) более высокой устойчивостью к цифровым шумам и хорошим балансом между размером стегоконтейнера и сохранностью внедренных данных.

Ранее был представлен метод для контроля целостности данных, защиты прав собственности на мультимедийную информацию, отслеживания распространения информации [2]. Он основан на изменении пространственной области документа-контейнера. Также были предложены реализующие его алгоритмы стеганографического преобразования, использующие в качестве контейнера элементы web-приложения на основе растровой графики. В качестве базового элемента контейнера, цветовые параметры которого модифицируются в модели RGB при осаждении информации, выступает пиксель изображения. Внедрение (извлечение) информации происходит в пикселях, имеющих одинаковое значение в одном или нескольких цветовых каналах. Особенностью разработанного метода является то, что процессы внедрения (извлечения) информации осуществляются при сравнительном анализе значений одного или двух цветовых координат базового пикселя и пикселя для внедрения. Псевдокод алгоритма внедрения, в соответствии с которым будут происходить преобразования, представлен в листинге 1.

Входные: сообщение m , web-документ c , опорный пиксель z_0

Выходные: стегоконтейнер S

- (1) Преобразовать web-документ c в файл растровой графики;
- (2) Расширить сообщение m ;
- (3) Подсчитать количество знаков n в расширенном сообщении m ;
- (4) Представить m в двоичном виде M ;
- (5) Определить начальные координаты встраивания;
- (6) Определить цветовой канал c_{RGB} ;
- (7) Определить цветовой канал c_{RGB}' , отличный от c_{RGB} ;
- (8) Определить ключевое значение канала Q ;
- (9) Определить значение цветового канала c_{RGB}' для пикселя z_0 ;
- (10) Сформировать массив пикселей $Z[]$ из изображения c ;
- (11) Определить размер l массива $Z[]$;
- (12) если $l \geq n + 1$ то
- (13) для i от 1 до $n + 1$:
- (14) Определить значение цветового канала c_{RGB}' для пикселя $z[i]$;
- (15) если $c_{RGB}'(z_0) > c_{RGB}'(z[i])$ то
- (16) если $M[i] = 1$ то
- (17) $c_{RGB}'(z[i]) = c_{RGB}'(z[i]) - Q$;
- (18) иначе если $M[i] = 0$ то
- (19) $c_{RGB}'(z[i]) = c_{RGB}'(z[i]) + Q$;
- (20) $i = i + 1$;
- (21) Сохранить преобразованное изображение c как S ;

Листинг 1 – Псевдокод алгоритма прямого преобразования

Ключевым шагом алгоритма внедрения является выбор массива пикселей, Z ($Z = \{z_i\}$, здесь $i \in [0, \text{length}(Z)]$). Алгоритм формирования массива Z представлен в листинге 2.

Входные: изображение-контейнер C , ключевой параметр Q , опорный пиксель z_0 , c_{RGB} – цветовой канал с совпадающими цветовыми параметрами пикселя, c_{RGB}' – цветовой канал для внедрения сообщения

Выходные: массив $Z[]$;

- (1) Определить t – высота контейнера C , r – ширина контейнера C ;
- (2) Определить значение пикселя z_0 для канала c_{RGB}' ;
- (3) для j от 1 до t :
- (4) для n от 1 до r :
- (5) Определить значение пикселя $c[j][n]$ для канала c_{RGB}
- (6) если $c_{RGB}(c[j][n]) == Q$ то
- (7) Определить значение пикселя $c[j][n]$ для канала c_{RGB}'

- (8) если $c_{RGB}'(z_0) - 2 \cdot Q \leq c_{RGB}'(c[j][n]) \leq c_{RGB}'(z_0) + 2 \cdot Q$ то
- (9) если $c_{RGB}'(z_0) - Q \leq c_{RGB}'(c[j][n]) \leq c_{RGB}'(z_0) + Q$ то
- (10) добавить пиксель $s[j][n]$ в массив $Z[]$;
- (11) иначе если $c_{RGB}'(c[j][n]) \leq c_{RGB}'(z_0) - Q$ то
- (12) $c_{RGB}'(c[j][n]) = c_{RGB}'(c[j][n]) - Q$;
- (13) иначе $c_{RGB}'(c[j][n]) = c_{RGB}'(c[j][n]) + Q$;
- (14) $n = n + 1$;
- (15) $j = j + 1$;
- (16) Вернуть $Z[]$;

Листинг 2 – Формирование массива пикселей Z

Псевдокод алгоритм обратного преобразования представлен на листинге 3.

Входные: web-документ s , c_{RGB} – цветовой канал с совпадающими цветовыми параметрами пикселя, c_{RGB}' – цветовой канал для внедрения сообщения, начальные координаты извлечения, ключевой параметр Q , длина сообщения n

Выходные: извлеченное сообщение m

- (1) Создать массив пикселей $z[]$ из документа s ;
- (2) Создать пустой массив $M[]$;
- (3) для p от 0 до 7:
- (4) Определить $c_{RGB}'(z[p])$ – цветовой код пикселя $z[p]$;
- (5) если $c_{RGB}'(z_0) > c_{RGB}'(z[i])$ то
- (6) $l = l \| 1$;
- (7) иначе $l = l \| 0$;
- (8) $p = p + 1$;
- (9) для p от 8 до $8l$:
- (10) Определить $c_{RGB}'(z[p])$ – цветовой код пикселя $z[p]$;
- (11) если $c_{RGB}'(z_0) > c_{RGB}'(z[i])$ то
- (12) $L = L \| 1$;
- (13) иначе $L = L \| 0$;
- (14) $p = p + 1$;
- (15) для p от $l(8+1)$ до $8L+7$:
- (16) Определить $c_{RGB}'(z[p])$ – цветовой код пикселя $z[p]$;
- (17) если $c_{RGB}'(z_0) > c_{RGB}'(z[i])$ то
- (18) $M = M \| 1$;
- (19) иначе $M = M \| 0$;
- (20) $p = p + 1$;
- (21) Преобразовать двоичный массив $M[]$ в строку m ;
- (22) Представить сообщение m в десятичном виде;
- (23) Вернуть m ;

Листинг 3 – Псевдокод алгоритма обратного преобразования

Для реализации представленных алгоритмов данного метода было разработано программное средство.

Для реализации стеганографических алгоритмов описанного методов (внедрение/извлечение M и статистическая обработка стегоконтейнера) разработано специализированное приложение (рабочее название – *StegTool*) на основе языка программирования *Python* с использованием фреймворка *Kivy* и библиотеки *Pillow* для обработки изображений. Вид одного из диалоговых окон (в режиме сравнения двух изображений) и основной функционал приложения показаны на рис. 1.

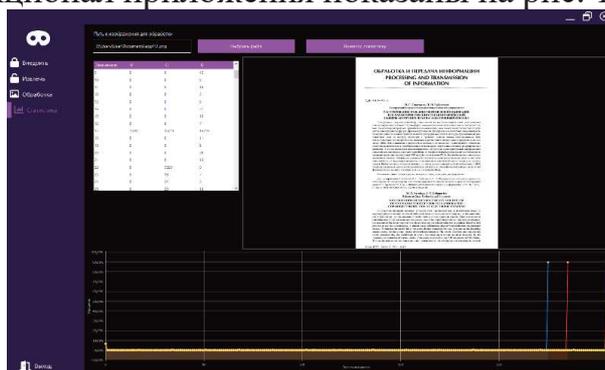


Рисунок 1 – Диалоговое окно приложения StegTool в режиме сравнения двух изображений

Описаны алгоритмические особенности стеганографического метода. Представлены псевдокоды основных алгоритмов (прямое и обратное преобразование, формирование массива Z). Представлено программное средство, реализующее предложенный метод. В программном средстве реализованы следующие функции: внедрение информации, извлечение информации, анализ изображения, выбранного в качестве контейнера. Данный функционал позволит не только провести стеганографическое преобразование, но и исследовать контейнер для улучшения эффективности и увеличения пропускной способности.

ЛИТЕРАТУРА

1. Шутько, Н. П. Моделирование стеганографической системы в задачах по охране авторских прав / Н.П. Шутько, Н.И. Листопад, П.П. Урбанович // Восьмая Междунар. научно-техн. конф. «Информационные технологии в промышленности» (ИТГ2015) : тезисы докладов. - Минск : ОИПИ НАН Беларуси, 2015. – С. 30-31.

2. Савельева М. Г., Урбанович П. П. Метод стеганографического преобразования web-документов на основе растровой графики и модели RGB // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. 2022. № 2 (260). С. 99–107.