**СОДЕРЖАНИЕ**

[3 ПРИМЕНЕНИЕ КРИВЫХ БЕЗЬЕ В СТЕГАНОГРАФИИ 2](#_Toc136183247)

[3.1 Кривая Безье 2](#_Toc136183248)

[3.2 Анализ существующих методов, базирующихся на кривой Безье 4](#_Toc136183249)

[3.3 Описание предлагаемого метода встраивания LIBC5 8](#_Toc136183250)

[3.4 Выводы по разделу 16](#_Toc136183251)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17](#_Toc136183252)

# 3 ПРИМЕНЕНИЕ КРИВЫХ БЕЗЬЕ В СТЕГАНОГРАФИИ

# 3.1 Кривая Безье

Кривая Безье - это математическая кривая, которая определяется точками начала и конца отрезка и контрольными точками, которые определяют форму кривой. Кривая Безье [152] названа в честь французского инженера Пьера Безье, который разработал этот метод в 1962 году во время работы над проектированием автомобильных кузовов. Кривые были названы именем Безье, а именем де Кастельжо назван разработанный им рекурсивный способ определения кривых (De Casteljau's algorithm). Кривая Безье может быть использована для создания плавных и сложных кривых, которые могут использоваться в различных областях, таких как графический дизайн, анимация и компьютерное моделирование.

Основная идея кривой Безье заключается в том, чтобы определить форму кривой, используя контрольные точки. На сегодняшний день формулы кривой Безье используются во многих отраслях науки, также многие ученые ведут исследования в этой области [153-158].

Кривая Безье является частным случаем многочленов Бернштейна, представляет собой параметрическую кривую и определяется следующим выражением (Eq.3.1):

  (3.1)

где

– количество опорных точек;

– номер опорной точки;

 – шаг;

 – координата опорной точки;

 – базисная функция кривой Безье.

Координаты кривой описываются в зависимости от параметра :

Добавить графикиДля двух точек (3.2):

 (3.2)

Для трёх точек (3.3):

 (3.3)

Для четырёх точек (3.4):

 (3.4)

Чем больше контрольных точек используется при определении кривой Безье, тем сложнее и точнее форма кривой может быть определена. Кривая Безье имеет ряд преимуществ перед другими способами определения кривых, такими как возможность создавать плавные и сложные кривые, которые могут быть легко изменены при необходимости. Кривые Безье также могут быть просто аппроксимированы с помощью компьютерных алгоритмов, что делает их удобными для использования в компьютерной графике.

Кривые Безье могут быть использованы в сокрытии данных для создания стегоконтейнера, который может содержать секретные данные. Одним из способов использования кривых Безье в сокрытии данных является разбиение изображения на маленькие фрагменты, называемые блоками. Каждый блок может быть аппроксимирован кривой Безье, которая может быть использована для кодирования секретных данных. Для сокрытия данных можно изменять параметры кривой, такие как контрольные точки, чтобы создать изменения в блоках, которые невидимы невооруженным глазом, но могут быть использованы для хранения секретных данных.

Сокрытие данных с использованием кривых Безье может быть выполнено с помощью различных методов. Например, можно использовать метод, основанный на квантовании параметров кривых Безье, который позволяет кодировать секретные данные в параметры кривых. Другой метод, основанный на использовании множественных кривых Безье, позволяет скрыть секретные данные в нескольких кривых. Кривые Безье также могут использоваться для сокрытия данных в звуковых и видеофайлах. Например, можно использовать кривые Безье для создания стегоконтейнера, который может хранить секретные данные в формате звуковых или видеоданных [159].

Однако, как и в любой другой технике сокрытия данных, использование кривых Безье для сокрытия данных имеет свои ограничения и уязвимости, и может быть обнаружено с помощью методов стегоанализа. Исследования использования кривой Безье в сокрытии данных являются довольно распространенным направлением в области криптографии. Ниже приведен обзор нескольких статей на эту тему:

1. В статье [160] исследователи представили новый метод сокрытия информации в изображениях с использованием кривой Безье. Их подход основан на замене пикселей изображения на точки на кривой Безье, которая строится в соответствии с ключом шифрования. Результаты экспериментов показали эффективность метода в сокрытии информации и низкую вероятность обнаружения.
2. В работе [161] авторы использовали кривую Безье первого ранга. Они разработали уравнение кривой Безье, подходящее для рассеяния значений цвета в процессе обработки изображения. Этот алгоритм (имитирующий кривую Безье первого порядка при кодировании изображений) полностью отличается от предыдущих алгоритмов, использовавших то же самое (кривая Безье), с точки зрения метода, используемого при шифровании. Результаты экспериментов показали эффективность метода в сокрытии информации
3. В статье [162] исследователи использовали кривую Безье в сочетании с усовершенствованным алгоритмом криволинейного преобразования для сокрытия данных в медицинских изображениях. Они предложили новый метод, который основан на замене частей изображения на соответствующие значения на кривой Безье, построенной с использованием криволинейного преобразования. Результаты экспериментов показали эффективность метода и его способность сохранять качество изображения.
4. В работе [163] статье предлагается эффективная и настраиваемая схема шифрования визуальных изображений путем объединения гиперхаотической системы 6D, измерения сжатия и встраивания кривой Безье. Экспериментальные результаты моделирования и всесторонний анализ производительности показывают, что схема, предложенная авторами, обладает высоким качеством дешифрования, визуальной безопасностью, надежностью и операционной эффективностью.

# 3.2 Анализ существующих методов, базирующихся на кривой Безье

Кривая Безье - это математическая кривая, которая часто используется в графическом дизайне и компьютерной графике. Она имеет различные приложения, включая использование в сокрытии данных. В этом обзоре я представлю несколько статей, которые рассматривают использование кривой Безье в качестве средства для сокрытия данных.

В статье [164] авторы исследуют применение кривой Безье в качестве инструмента для сокрытия данных в цифровой водяной метке. Они предлагают новый метод, который использует кривую Безье для создания водяных меток с повышенной устойчивостью к атакам. Параметрическое уравнение кубической кривой Безье выглядит следующим образом: . Чтобы однозначно определить эту кривую, требуются четыре точки: P1, P2, P3 и P4. На рисунке 3.1 показана кубическая кривая Безье. Кривая состоит из трех сегментов, первый из которых отмечен дополнительными красными линиями, второй сегмент отмечен оранжевыми линиями, а третий отмечен зелеными линиями. Линии нанесены на контрольные точки для наглядности, чтобы продемонстрировать, как расположение контрольных точек влияет на внешний вид кривой.

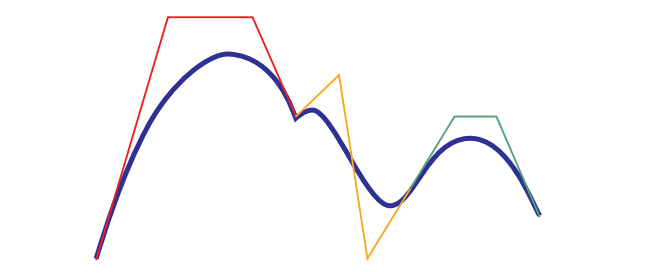


Рисунок 3.2 - Кубическая кривая Безье, состоящая из трех сегментов

Авторы на рисунках 3.2 -3.3 предложили структурную схем алгоритма встраивания и извлечения сообщения из стегоконтейнера.



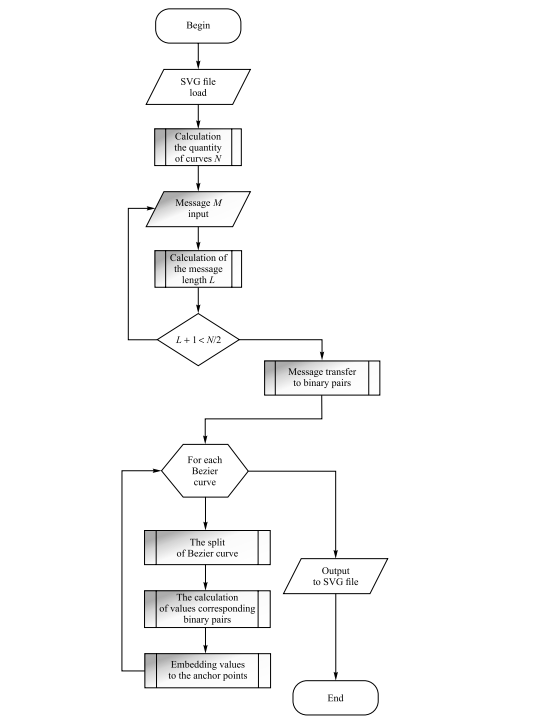


Рисунок 3.2 - Структурная схема алгоритма встраивания скрытого сообщения

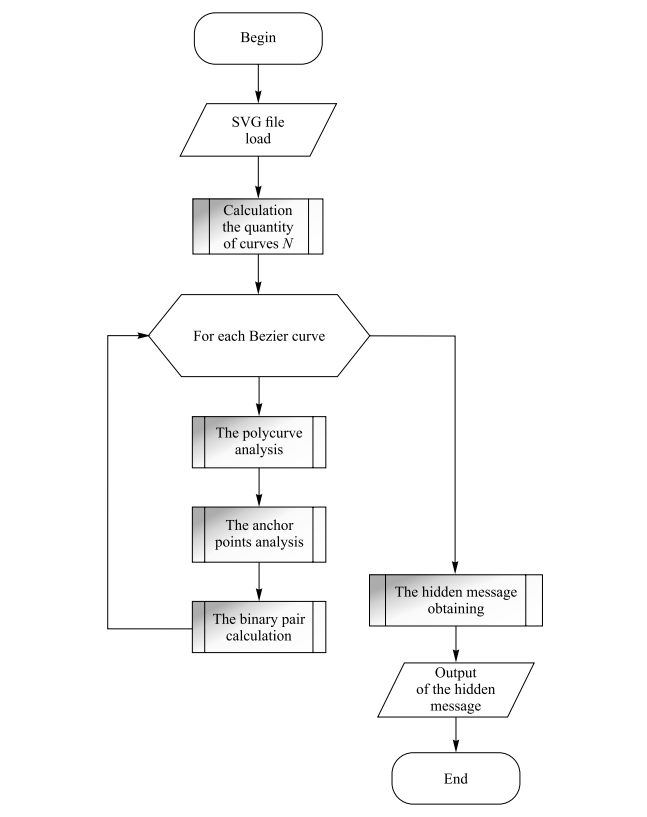


Рисунок 3.3 – Структурная схема алгоритма извлечения сообщения из контейнера

В результате был рассмотрен новый стеганографический метод встраивания и извлечения скрытых сообщений при использовании SVG-файлы в качестве контейнеров для хранения. Метод основан на изменении параметров кубических кривых Безье. Для реализации метода была разработана библиотека StegoSVG, которая использовалась при разработке авторского настольного приложения. Экспериментальные результаты показали эффективность метода в сокрытии информации и низкую вероятность обнаружения.

В разработке [165] авторы предлагают новый метод сокрытия текста на основе кривой Безье и описывают новый метод встраивания сообщения в структурированные формы. Алгоритм встраивания сообщений представляет контуры фигуры в виде набора кубических кривых Безье и отрезков прямых линий. Результаты обширного субъективного теста подтверждают, что изменения формы действительно незаметны. Далее, чтобы протестировать восстановление битов сообщения в зашумленных физических средах, текстовый документ проходит процедуру печати-ксерокопирования-сканирования. Обнаружено, что восстановление сообщений происходит стабильно даже после нескольких циклов копирования. Они показывают, что этот метод имеет высокую устойчивость к атакам и может быть эффективно использован для защиты конфиденциальной информации.

В статье [166] авторами представлен подход к генерации защищенного зашифрованного текста. Идея предлагаемого метода заключается в шифровании ключа, который был сгенерирован с помощью открытого ключа в криптосистеме ElGamal (ElGamal elliptic curve) с эллиптической кривой, с использованием квадратичного уравнения кривой Безье, основанного на поведении PGP.



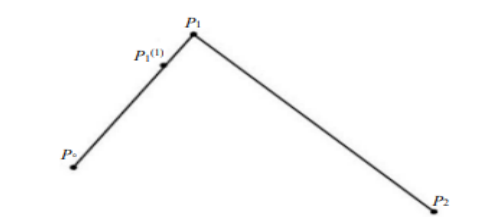


Рисунок 3.4 – Представление интересов  точки



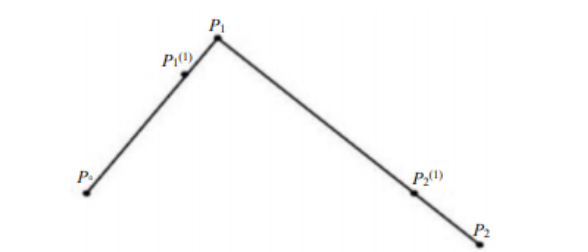


Рисунок 3.5- Представление интересов 

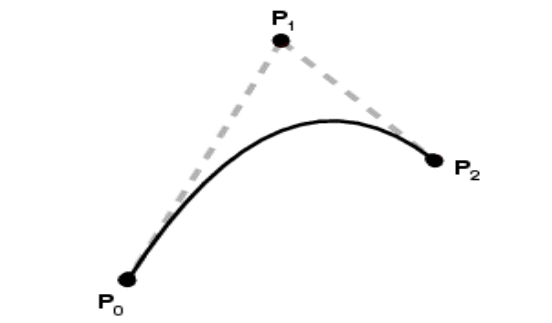


Рисунок 3.6 - Квадратичная кривая Безье

Предложенный способ оказался более эффективным и сложным.

В статье [167] авторы предлагают новый метод сокрытия данных в изображениях на основе кривой Безье и вейвлет-преобразования. Авторы использовали PSNR в качестве показателя качества изображения. Они показывают, что этот метод имеет высокую устойчивость к атакам и может быть эффективно использован для защиты из

# 3.3 Описание предлагаемого метода встраивания LIBC5

(Разбить на Описание, стегоанализ)

Назовем предлагаемый алгоритм LIBC5 (Lagrange interpolation Bezier curve for 5 points). Алгоритм устроен следующим образом. Рассмотрим группу из пяти значений яркости пикселей интерполированного растрового изображения:. Значения , , являются значениями пикселей оригинального изображения по выбранной составляющей R, G либо B, а значения и являются добавленными путем интерполяции по соответствующей составляющей и взяты последовательно. Встраивание бита информации происходит в пикселях и путем взятия округленного до целого значения ближайшей точки с кривой Безье, такой, что младший бит ее значения совпадает с битом информации, которую мы хотим встроить. Чтобы подобрать нужное значение точки кривой Безье, необходимо задать такой шаг t, который обеспечит достаточный выбор значений точек кривой.

Рассмотрим график усредненных яркостей пикселей и построенной кривой Безье по их значениямc шагом t=0.1 (Рис.2):

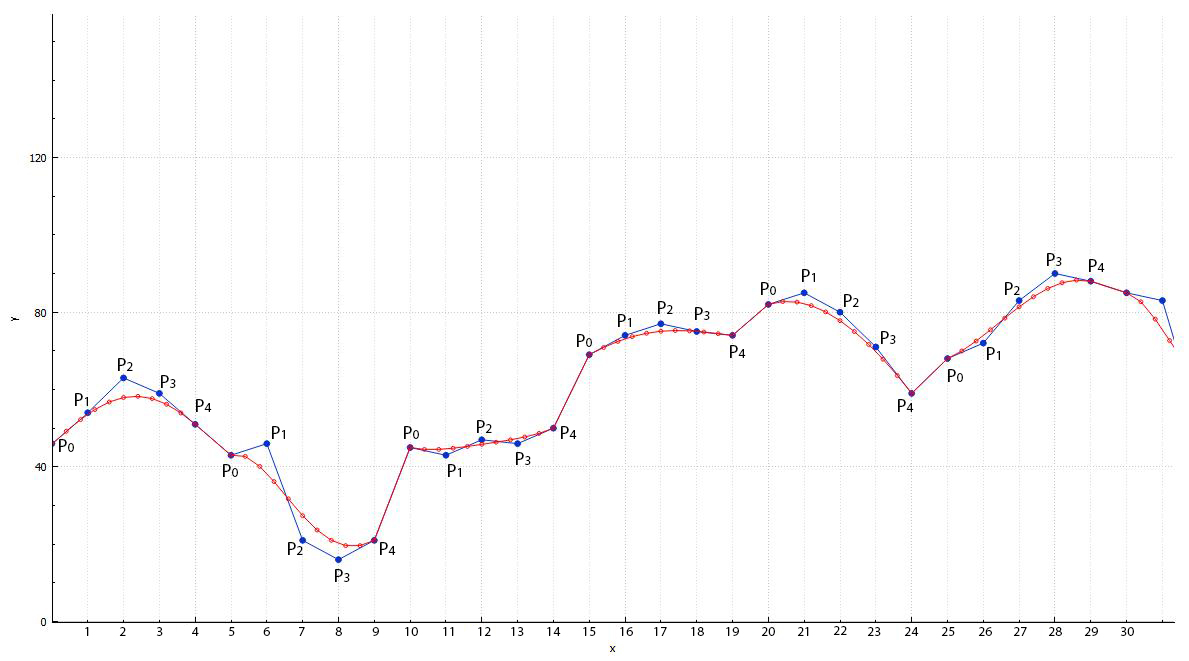


Рисунок 2. Построение кривой Безье для файла 029.bmp с шагом t=0.1

По оси Y находятся значения яркостей изображения одной из составляющих пикселей, взятых из первой строки матрицы пикселей по порядку. Ось Х демонстрирует линейный порядок пикселей и показывает их порядковый номер. Для наглядности в данном примере был взят шаг t=0.1. По графику видно, что на каждые 5 точек яркостей интерполированной картинки (синяя линия), мы имеем 11 точек кривой Безье (красная линия).

Биты информации записываются в и в путем взятия такого значения с кривой Безье, у которого младший бит будет равен встраиваемому биту сообщения. На рисунке 3 показано как определяются соответствующие значения для замены пикселей и :

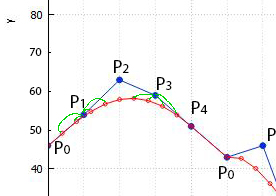


Рисунок 3. Точки кривой Безье для замены пикселей и для растрового файла с шагом t=0.1

По графику на рисунке2 видно, что для точки имеется выбор из 4 значений точек кривой Безье, которые можно использовать, округлив их до целого значения и используя их младший бит. Необходимо взять с кривой Безье наиболее близкое значение с точкой такое, что младший бит этого значения будет совпадать с битом встраиваемого сообщения. Затем для точки также нужно подобрать подходящее значение с кривой Безье из четырех возможных, после чего строится отрезок кривой Безье по следующим пяти точкам пикселей и действия повторяются.

Обозначим округленные значения кривой Безье, которые можно использовать для замены текущей точки или множествами и соответственно, где k – это количество значений при данном шаге t. Значение k фактически представляет собой количество отрезков, на которые делится кривая, проходящая от точки до точки за исключением первого и последнего отрезков, а также за исключением тех двух отрезков, которые окружают значение кривой, соответствующей по порядку. Таким образом, чтобы вычислить k, нужно разделить единичный отрезок на длину интервала t, отнять 4 отрезка и разделить полученное значение пополам, для замены двух значений и . Затем нужно прибавить единицу, так как нам необходимо получить в итоге не количество отрезков, а количество полученных точек(Eq.6):

(9)

Упростив данное выражение, получаем формулу для вычисления k(Eq.10):

(10)

Таким образом, мы видим, что при t=0.1 создается выбор из четырех возможных значений. Краевые значения на кривой, соответствующие , и не используются для замены значений и , так как они соотносятся с теми пикселями изображения, которые изначально составляли матрицу пикселей до процесса интерполяции.

Так как значения точек кривой Безье должны быть округлены до целого, то мы можем получать одинаковые значения, которые сократят выбор для замены и . Поэтому далее рассмотрим случай при t=0.05. На Рисунке 4 красной линией показана кривая Безье, а синей линией построен график яркостей текущего растрового изображения:

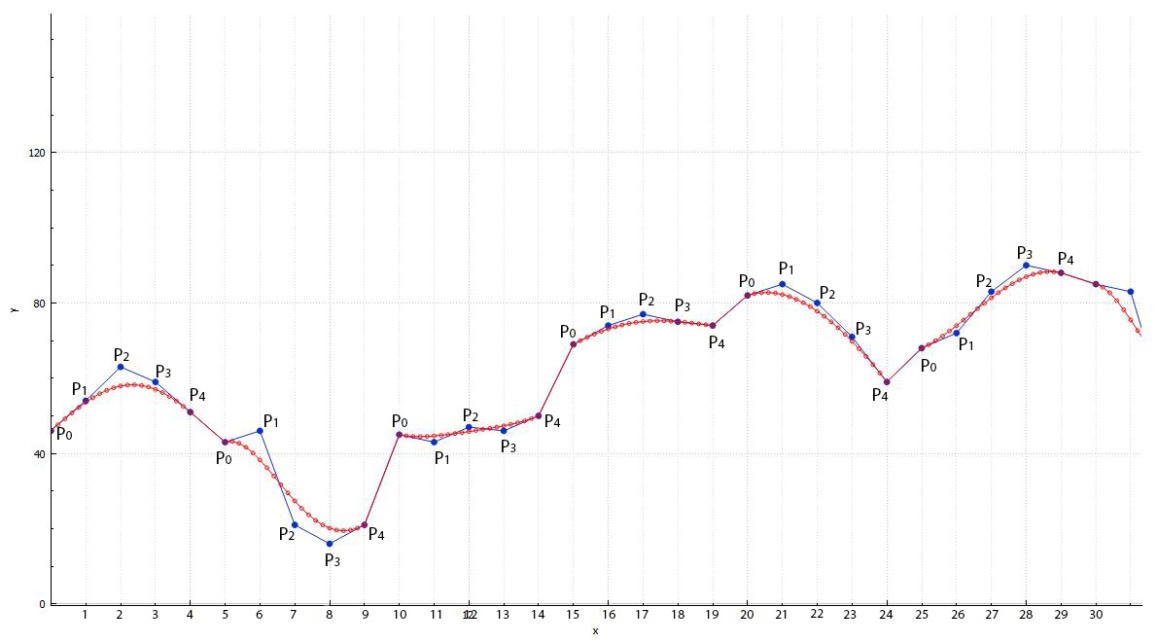


Рисунок 4. Построение кривой Безье для файла 029.bmp с шагом t=0.05

При построении кривой Безье c шагом t=0.05 мы имеет выбор из k=9 значений для замены каждого из и . На рисунке 5 показаны возможные точки для замены и .

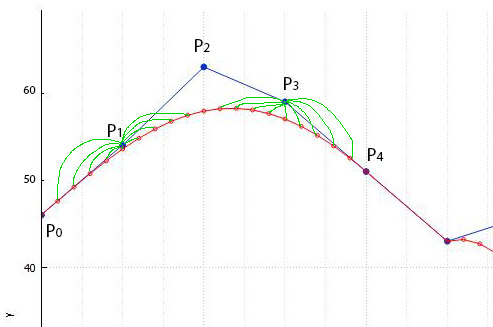


Рисунок 5. Возможные точки для замены и на кривой Безье для файла 029.bmp с шагом t=0.05.

Рассмотрим пример встраивания сообщения с авторской информацией в растровый файл 029.bmp с помощью рассмотренного метода LIBC5. В таблице 1 показаны округленные значенияR при соответствующих значениях t для файла 029.bmp, а также соответствие этих значений точкам, , , и :

Таблица 1. Пошаговые значения  для группы из пяти пикселей

|  |  |  |
| --- | --- | --- |
|  |  | P |
| 0 | 46 |  |
| 0.05 | 48 |  |
| 0.1 | 49 |  |
| 0.15 | 51 |  |
| 0.2 | 52 |  |
| 0.25 | 54 |  |
| 0.3 | 55 |  |
| 0.35 | 56 |  |
| 0.4 | 57 |  |
| 0.45 | 57 |  |
| 0.5 | 58 |  |
| 0.55 | 58 |  |
| 0.6 | 58 |  |
| 0.65 | 58 |  |
| 0.7 | 58 |  |
| 0.75 | 57 |  |
| 0.8 | 56 |  |
| 0.85 | 55 |  |
| 0.9 | 54 |  |
| 0.95 | 53 |  |
| 1 | 51 |  |

По таблице 1 видно, что для замены значения имеются следующие варианты: 48, 49, 51, 52, 54, 55, 56, 57. А для замены : 58, 57, 56, 55, 54, 53. Также следует учитывать, что из возможных значений для замены и мы должны выбрать одно подходящее значение, младший бит которого совпадет с битом сообщения для встраивания. При этом, если младший бит либо уже совпал с битом сообщения, то замены не потребуется.

Предположим, что последовательность битов сообщения:m={1,0}. Переведем в двоичный вид значения множества R1, соответствующего:

=4810=1100002

=4910=1100012

=5110=1100112

=5210=1101002

=5410=1101102

=5510=1101112

=5610=1110002

=5710=1110012

Для встраивания первого бита по порядку из последовательностиm, нам необходимы такие значения изR1, у которых самый младший бит равен 1, то есть , , и . Заменив значения пикселя на любое из них, мы таким образом запишем бит 1 в матрицу пикселей растрового файла и перейдем к .

Для переведем в двоичный вид значения соответствующего множества R3, исключая повторяющиеся значения:

=5810=1110102

=5710=1110012

=5610=1110002

=5510=1101112

=5410=1101102

=5310=1101012

Для встраивания второго бита по порядку из последовательности m, нам необходимы такие значения изR3, у которых самый младший бит равен 0, то есть , , и . Заменив значения пикселя на любое из них, запишем бит 0 в матрицу пикселей растрового файла. Таким образом, блок из пяти пикселей пройден, в него записано 2 бита сообщения m.

Аналогично происходит встраивание в последующие блоки изображения, взятые линейно из матрицы пикселей. Изображение 029.bmp размером 450 по ширине и 450 по высоте позволяет использовать по 90 блоков в каждой строке для каждой из составляющих RGB. Таким образом, общее количество блоков для данного изображения составит 40500 при использовании одной из составляющих пикселей либо 121500 при использовании трех составляющих. При встраивании в каждый блок по 2 бита, мы получаем формулу (11) для вычисления максимального количества встроенных бит в растровое изображение по одной из составляющих размером W по ширине и H по высоте предложенным методом встраивания(Eq.11):

(11)

В рассмотренном примере N=81000 бит или 10125 байт. В случае использования всех составляющих яркость пикселей, значение N увеличится в три раза. Следует отметить, что при необходимости заполнения контейнера на половину его ёмкости, можно использовать блоки пикселей через один, так как кривая Безье по пяти точкам строится в отдельности для каждого блока из пяти пикселей.

Если мы вернемся к рассмотренному примеру, то закономерно возникает вопрос: По какому алгоритму выбирается значение ri для замены пикселя изображения, если мы имеем выборку из нескольких подходящих значений? Данный алгоритм был реализован таким образом, что из подходящей выборки значений ri выбирается первое по порядку и используется для замены пикселя. Таким образом, при замене используется значение точки кривой Безье, наиболее близкое к из набора подходящих значений.

Обозначим множество подходящих значений ri для встраивания каждого очередного бита mj как и , то есть содержит значения из , а содержит значения из такие, что их самый младший бит совпадает с текущим битом встраиваемого сообщения. Значения e и f переменны и зависят от пикселей используемого изображения. Чем меньше шаг t, тем меньше вероятность того, что k окажется равным e или f. При возникновении такой ситуации, что либо , ни одно из значений множества R1 либоR3не подойдет для встраивания очередного бита сообщения. При декодировании невозможно будет определить наличие такого случая, что создаст ошибку декодирования в связи с потерей бита сообщения. В программной реализации мы взяли шаг t=0.01, который снизит вероятность данной проблемы и позволит получить достаточную выборку S1 иS3Но каким бы маленьким ни был шаг t, количество значений множеств R1 иR3 всегда будет ограничено, так как они являются округленными значениями, высчитанными по формуле (1) и даже при максимально возможном количестве построенных точек кривой Безье мы будем получать несколько равных значений. Поэтому здесь стоит подумать над качеством исходных контейнеров. Если мы рассмотрим растровое изображение, в котором встречаются блоки только с одинаковой яркостью, то неизбежно столкнемся с потерей встраиваемого бита сообщения. Таким образом, для однозначного декодирования, необходимо предусмотреть случаи заведомо неподходящих блоков пикселей, которые будут пропущены. Алгоритм LIBC5 был запущен для набора из 800 растровых изображений grayscale размером 450 на 450 точек, произведено успешное декодирование встроенных сообщений, представляющих собой псевдослучайную последовательность. Кривая Безье строилась с шагом t=0.01.

Заменяя в матрице пикселей значения, полученные с помощью интерполяции Лагранжа, мы не нарушаем общую статистику пикселей изображения. В то же время, используя кривую Безье, мы пользуемся плавным переходом значений пикселей, минимизируя возможные искажения. Таким образом, можно заполнить встроенной информацией до 50% пикселей картинки. Но так как в исследованных ранее методах [12-13] процент встраивания оставался на уровне 21%, то для сравнительного анализа мы встроили примерно такое же количество информации.

**стегоанализ**

Далее мы провели стегоанализ разработанного метода LIBC5. Для эксперимента был использован набор из 800 изображений размером 450х450 пикселей, заполненных предложенным методом на 21%. Так как сообщение перед внедрением шифруют, то оно выглядит как случайная последовательность. Соотвественно, в нашем эксперименте сообщение имитировалось при помощи генератора случайных чисел.

Анализ полученных стегоконтейнеров проводился методом RS (Regular–Singular) [14]. RS анализ использует чувствительный метод двойной статистики, полученной из пространственных корреляций в изображениях. В RS методе существуют три главных фактора, которые влияют на точность оцененной длины сообщения: начальное отклонение, уровень шума изображения-контейнера и размещение бит сообщения в изображении. Этот метод показывает достаточно точный результат даже на шумных изображениях. Проведенные эксперименты методом RS показали стегоустойчивость алгоритма LIBC5. В таблице 2. приведен результат RS анализа на наборе пустых контейнеров, а в таблице 3 - на заполненных контейнерах с помощью метода LIBC5. Полученные результаты подсчета ошибки первого рода показали, что ошибка составляет 0%. Процент обнаружения встроенной информации методом RS показал абсолютную устойчивость разработанного нами метода LIBC5 в отношении данного вида стагоанализа.

Таблица 2. RS анализ на наборе пустых контейнеров (800 картинок)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L | | 0% | 1-4% | 5% и более |
| Доля файлов | 450x450 | 61,25 | 38 | 0,5 |

Таблица 3. RS анализ на наборе заполненных контейнеров (800 картинок)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L | | 0% | 1-4% | 5% и более |
| Доля файлов | 450x450 | 84,75 | 15,25 | 0 |

Проведем сравнение результатов разработанного метода со стеганографическим методом INMI (Improved Neighbor Mean Interpolation), который мы исследовали в статье [12]. В работе [15] ранее была получена модификация метода INMI, основанная на использовании интерполяционного полинома Лагранжа второй степени для получения изображения-контейнера. Интерполяционная формула Лагранжа имеет следующий вид (Eq.12):

 (12)

где  - многочлен степени , принимающий значение равное единице, в узле . равное нулю – в остальных узлах ,.

В модифицированном алгоритме метода INMI изображение, полученное путём добавления дополнительных строк и столбцов пикселей к исходному изображению, рассматривались фрагментами по 5 пикселей, пронумерованных от 0 до 4. Известные пиксели (0, 2, 4) считаются узлами интерполяции. Поэтому с помощью интерполяционного полинома Лагранжа второй степени были найдены неизвестные значения пикселей (1 и 3). Изображение, полученное путем добавления дополнительных строк и столбцов пикселей к исходному изображению, рассматривали фрагментами по 5 пикселей, пронумерованных от 0 до 4.

Значения пикселей изображения-контейнера получают по следующей формуле (Eq.13):

 (13)

где  – номер пикселя во фрагменте из пяти пикселей.

На основании рассмотренного алгоритма INMI, в работе [12] мы определили, что максимальная емкость контейнера составляет 21% и зависит от изображения. Мы проводили исследования на наборе из 800 изображений размера 450 на 450, то есть используя тот же набор стегоконтейнеров, что и в данном исследовании метода LIBC5. Приведем результирующие таблицы стегоанализа метода INMI:

Таблица 4. RS анализ на наборе пустых контейнеров размером 450х450 (800 картинок)

|  |  |  |  |
| --- | --- | --- | --- |
| Доля верно обнаруженных стегоконтейнеров | L | | |
| 0% | 1-4% | 5% и более |
| 61 | 39 | - |

Таблица 5. RS анализ на наборе контейнеров размером 450х450, заполненных с помощью метода интерполяции на 12%.

|  |  |  |  |
| --- | --- | --- | --- |
| Доля верно обнаруженных стегоконтейнеров | Длина L | | |
| 0% | 1-4% | 5% и более |
| 51,5 | 48 | 0,5 |

Сравнив полученные результаты по разработанному нами методу LIBC5 в таблицах 2 и 3, и результаты метода INMI в таблицах 4 и 5, можно сделать вывод, что LIBC5 более устойчив к стегоанализу и может успешно применяться на растровых изображениях в задачах встраивания ЦВЗ.

# 3.4 Выводы по разделу

Рассмотрены существующие методы , базирующиеся на Безье. Предложен новый метод, который.... В ходе экспериментального исследования было показано, что .... Рекомендуется использовать новый метод ...

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Bezier, P.E. Numerical Control-Mathematics and applications. (1972) John Wileyand Sons, London.
2. Ntoko, N-M. A formulation for Bézier-type curves. [Computers in Industry](https://www.scopus.com/sourceid/19080)Том 15, Выпуск 4, Страницы 363 - 368December 1990.
3. Kajla, A., Acar, T. Bézier–Bernstein–Durrmeyer type operators. [Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales - Serie A: Matematicas](https://www.scopus.com/sourceid/19400157008). Том 114, Выпуск 11 2020.
4. Wu, Y., Li, Xin. Curve intersection based on cubic hybrid clipping. [Visual Computing for Industry, Biomedicine, and Art](https://www.scopus.com/sourceid/21101037157)*.*Том 5, Выпуск 1, 2022.  # 17. DOI 10.1186/s42492-022-00114-3
5. Блинова Е.А., Голик А.А. Модификация стеганографического метода на основе встраивания дополнительных значений координат в изображениях формата SVG. – 2018. C.130-133
6. Анисимова Э.С. Сжатие изображений с помощью квадратичных кривых Безье //Естественные и математические науки в современном мире. – 2014. – №. 14. – С. 42-46.
7. Ковтун В.Ю., Кинзерявый А.Н. Экспериментальное исследование метода побитового сокрытия информации в векторные изображения //Безпека iнформацii. – 2014. – Т. 20. – №. 1. – С. 66-70.
8. Менщиков А.А., Шниперов А.Н. Метод скрытого встраивания информации в векторные изображения //Доклады Томского государственного университета систем управления и радиоэлектроники. – 2015. – №. 1 (35). – С. 100-106.
9. Karim A.A., Hussein A.M. J. A., Alwan H. M. Image Steganography System Using Bezier Curve //Al-Mansour Journal. – 2019. – Т. 31. – №. 1. – С. 111-133.
10. Al-kufi M. A. H. J., Kadhim O. N., Razaq E. S. Simulate a first-order Bézier curve in image encoding //Journal of Physics: Conference Series. – IOP Publishing, 2020. – Т. 1530. – №. 1. – С. 012080.
11. Dumka A. et al. Advanced digital image processing and its applications in Big Data. – CRC Press, 2020.
12. Wang X., Ren Q., Jiang D. An adjustable visual image cryptosystem based on 6D hyperchaotic system and compressive sensing //Nonlinear Dynamics. – 2021. – Т. 104. – №. 4. – С. 4543-4567.
13. Блинова Е. А., Урбанович П. П. Steganographic method based on hidden messages embedding into Bezier curves of SVG images //Журнал Белорусского государственного университета. Математика. Информатика. – 2021. – №. 3. – С. 68-83.
14. Ghadi D.M. A Novel Encryption System Based on PGP Using Elliptic Curve Cryptosystem and Bézier Curve for Secure Information Exchange //Journal of Physics: Conference Series. – IOP Publishing, 2021. – Т. 1999. – №. 1. – С. 012111.