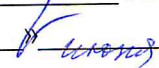


Министерство образования и науки Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра программирования

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП

д-р техн. наук, профессор

 А.Ю. Матросова

«5»  2018 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

ЗАЩИТА ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ СТЕГАНОГРАФИИ В МЕДИАФАЙЛАХ

по основной образовательной программе подготовки магистров
«Математическое и программное обеспечение вычислительных машин и компьютерных
сетей»

направления подготовки 01.04.02 Прикладная математика и информатика

Афанасьев Василий Иванович

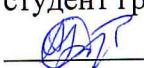
Научный руководитель ВКР
канд. техн. наук, доцент

 С.А. Останин

«4»  2018 г.

Автор работы

студент группы № 1162М

 В.И. Афанасьев

Министерство образования и науки Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Факультет прикладной математики и кибернетики
Кафедра программирования

УТВЕРЖДАЮ
Руководитель ООП
д-р техн. наук, профессор
А.Ю. Матросова
« 5 » февраля 2018 г.

ЗАДАНИЕ

по подготовке ВКР _____ магистра _____
специалиста, бакалавра, магистра нужное вписать (напечатать)

студенту Афанасьеву Василию Ивановичу группы № 1162М
фамилия, имя, отчество

1. Тема ВКР Защита информации с использованием стеганографии в медиафайлах

2. Срок сдачи студентом выполненной ВКР:

а) на кафедре 4 июня 2018
б) в ГЭК 7 июня 2018

3. Исходные данные к работе:

Цель работы: разработать программный продукт для внедрения скрываемой информации в медиафайлы (видео, аудио и изображения).

Задачи исследования:

- 1) Изучить процесс оцифровки аудио, видео и изображений;
- 2) Изучить принципы хранения оцифрованных медиаданных;
- 3) Разработать методы внедрения информации в медиафайлы;
- 4) Разработать методы модифицирования внедряемых данных, для затруднения их обнаружения;
- 5) Реализовать разработанные методы в виде программного продукта, выполняющий внедрение и извлечение скрываемых данных.

Объект исследования – применение стеганографии для защиты информации, путем внедрения скрываемой информации в медиафайлы.

Методы исследования: анализ данных, полученных экспериментальным путем.

4. Краткое содержание работы:

- 1) Постановка задачи
- 2) Описание предметной области
- 3) Описание методов внедрения информации
- 4) Описание разработанного программного продукта

5. Указать предприятие, организацию, по заданию которого выполняется работа
Кафедра программирования ТГУ

6. Перечень графического материала (с точным указанием обязательных чертежей)

Количество рисунков: 23

Количество таблиц: 4

7. Дата выдачи задания « 5 » февраля 2018 г.

Руководитель ВКР (Научный руководитель ВКР)

Доцент. канд. техн. наук.
должность, место работы


подпись

С. А. Остапенко
инициалы, фамилия

Задание принял к исполнению

5 февраля 2018
дата, подпись студента



АННОТАЦИЯ

К магистерской на тему «Защита информации с использованием стеганографии в медиафайлах».

Выполнена студентом факультета прикладной математики и кибернетики, Томского Государственного университета Афанасьевым Василием Ивановичем.

Магистерская выполнена на ---- страницах, состоит из введения, четырех глав, заключения и списка литературы.

В первой главе описываются задачи, которые необходимо выполнить в рамках данной магистерской диссертации, также описан функционал программного продукта, который необходимо было реализовать.

Во второй главе описываются принципы оцифровки и хранения медиаданных.

В третьей главе описаны методы внедрения скрываемой информации в медиафайлы, а также метод модификации данных для затруднения их обнаружения скрытых данных в файлах-контейнерах.

В четвертой главе описываются средства использованные при реализации методов внедрения скрываемой информации, а также требования предъявляемые к разработанному программному продукту.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
1 Постановка задачи	8
2 Описание предметной области.....	10
2.1 Структура файла растрового изображения	10
2.2 Структура аудиопотока.....	11
2.3 Структура видеопотока.....	13
3 Описание методов внедрения информации	15
3.1 Предлагаемый метод внедрения скрываемой информации в растровую графику	15
3.2 Предлагаемый метод внедрения скрываемой информации в аудиопоток.....	18
3.2.1 Структура WAV файла	18
3.2.2 Описание метода внедрения.....	20
3.4 Предлагаемый метод внедрения скрываемой информации в видеопоток.....	22
3.3 Методы обнаружения внедренной информации	27
3.3.1 Обнаружение скрытой информации в аудиопотоке	27
3.3.2 Обнаружение скрытой информации в видеопотоке.....	28
3.4 Разработанный метод затруднения обнаружения внедренной информации.....	31
4 Описание разработанного программного продукта.....	34
4.1 Анализ существующих решений	34
4.2 Используемые технологии.....	35
4.3 Почему именно эти технологии	36
4.4 Требования, предъявляемые к программному продукту.....	36
4.5 Описание формата входных и выходных данных	37
4.6 Руководство пользователя	38
ЗАКЛЮЧЕНИЕ.....	41
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	42

ВВЕДЕНИЕ

Во все времена методы защиты информации отличались высокой актуальностью. С появлением компьютерных технологий, защита информации стала развиваться как отдельная ветвь научных исследований. Современные цифровые устройства позволяют с легкостью создавать, хранить и передавать информацию в виде видео, аудио и изображений. Текстовая информация обычно переводится в цифровой вид. В наше время имеется несколько методов защиты информации. Самым широко используемым является криптография. В данной статье речь идет о другом методе - стеганографии, который, в отличие от криптографии, скрывает сам факт наличия информации, другими словами, позволяет внедрить информацию в какой-либо файл таким образом, что файл контейнер будет выглядеть как обычный файл без внедренной информацией.

Термин "стеганография" произошёл от греческого языка, буквальный перевод которого - "тайнопись". В древности общей чертой стеганографии был способ записать информацию на какой-либо неприметный объект, после чего он отправлялся адресату. Защита информации достигалась тем, что о наличии сообщения знали лишь отправитель и предполагаемый получатель.

С давних времен известны различные способы и приемы использования стеганографии. К ним относятся: сообщения, написанные на дощечке, которые затем покрываются воском; информация, записанная на голову человека, скрытая, последующим отращиванием волос; "промаргивание" азбуки Морзе во время видеообращения; симпатические чернила, которые проявляются при определенных условиях (например, при ультрафиолетовом излучении); и т.д.[1].

Новый стимул в развитии стеганография получила в связи с появлением электронно-вычислительных систем. Так появились два новых направления: компьютерная и цифровая стеганографии. Компьютерная стеганография основана на способах скрытия информации, используя особенности компьютерной платформы (например, внедрение информации в свободные

кластеры файловой системы). Цифровая стеганография использует методы внедрения информации в цифровые объекты, вызывая некоторые искажения этих объектов.

С ростом объема различной информации, растет доля сведений, которые необходимо держать в тайне. Применение компьютеров позволило усовершенствовать известные идеи скрытия информации, с тем чтобы затруднить обнаружение факта её внедрения[2].

1 Постановка задачи

Необходимо, разработать методы внедрения и извлечения информации в медиафайлы (изображения, видео, звук), а также их модификацию для затруднения их обнаружения. Используя объектно-ориентированное программирование реализовать их в виде программного продукта.

Ключевыми функциями разработанного программного продукта являются:

1. Внедрение и извлечение скрываемой информации в изображения;
2. Внедрение и извлечение скрываемой информации в аудиопоток;
3. Внедрение и извлечение скрываемой информации в видеопоток;
4. Модификация скрываемой информации для затруднения их обнаружения.
5. Изменение разрешения выходного видеопотока;
6. Изменение частоты кадров (FPS) выходного видеопотока.
7. Выбор количества изменяемых бит в изображениях и видеопотоке;

В рамках магистерской диссертации требуется:

1. Изучить принципы хранения мультимедийной информации;
2. Изучить структуры медиафайлов;
3. Разработать методы внедрения скрываемой информации в мультимедийные данные;
4. Изучить методы и средства обнаружения скрытой информации в файлах-контейнерах;
5. Разработать методы модификации внедряемых данных, для затруднения их обнаружения;

6. Создать алгоритмы внедрения скрытой информации на основе разработанных методах;
7. Реализовать разработанные алгоритмы в виде программного продукта, позволяющего внедрять скрываемую информацию в видео, аудио и изображения определенных типов.

2 Описание предметной области

Стеганография – это наука о скрытой передаче информации, в которой скрывается сам факт её существования.

Стегосистема – это совокупность средств и методов для скрытой передачи информации.

Файл-контейнер – это компьютерный файл, в который осуществляется внедрение или извлечение скрываемой информации.

Медиафайл – это компьютерный файл, который содержит аудиопоток, видеопоток или изображения, в соответствии со спецификой мультимедийного контейнера (avi, mp4, wav, bmp и др.).

LSB (Least Significant Bit) – метод внедрения информации, заключающийся на изменении наименее значимых бит.

Кодек – это программа, предназначенная для кодирования и декодирования мультимедийных данных. Существуют множество алгоритмов для сжатия мультимедийных данных, но все они делятся на две категории: сжатие с потерями и сжатие без потерь. Алгоритмы сжатия без потерь позволяют получить исходные данные после декодирования без потери информации. Алгоритмы сжатия с потерями преобразуют мультимедийные данные таким образом, что исходные данные теряют часть информации[3].

2.1 Структура файла растрового изображения

Растровое изображение – это совокупность точек(пикселей), расположенных в определенной последовательности и имеющих определенный цвет. Каждый пиксель кадра имеет свои значения, определяющие цвет. Существуют несколько видов цветовых моделей. Самые распространённые из них это YUV и RGB. Рассмотрим последний из них, так как цветовая модель YUV в большинстве случаев используется в алгоритмах сжатия с потерями[4].

RGB(Red Green Blue) – цветовая модель, представление цвета которой задается совокупностью трех цветовых каналов: красного, зеленого и синего. Каждый из каналов имеет размер в один байт, из чего следует, что цвет одного пикселя представляется в виде трех байт (Рисунок 1).

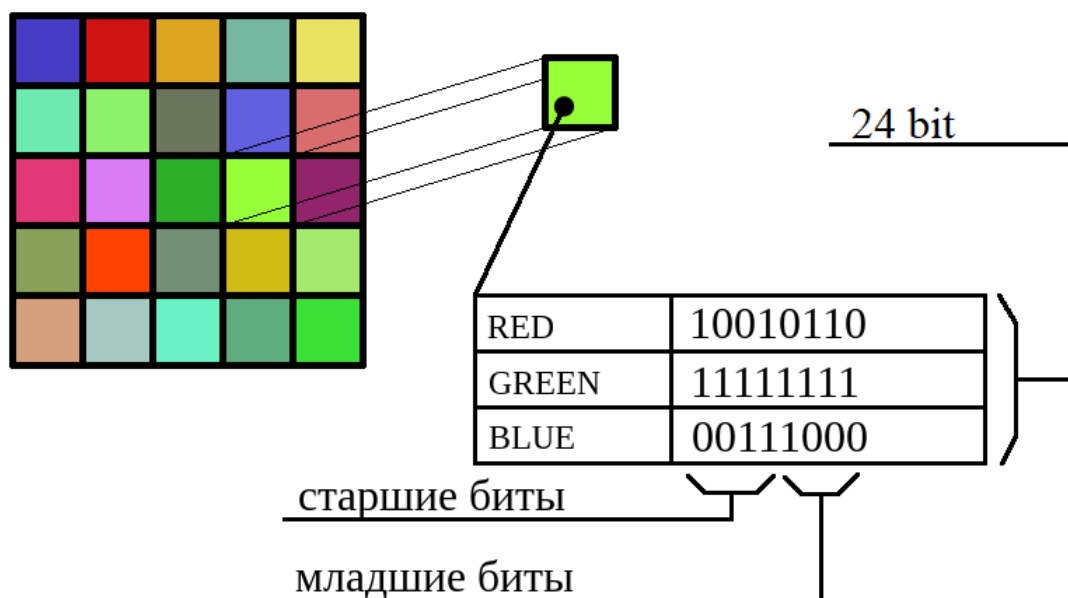


Рисунок 1 – Структура одного пикселя растрового изображения

Основные характеристики растрового изображения:

1. Размерность – совокупность длины и ширины изображения;
2. Цветовая модель – модель описания цвета в виде кортежей чисел;
3. Глубина цвета – размер выделяемый для отображения цвета одного пикселя.

2.2 Структура аудиопотока

Подробно рассмотрим, как хранится звук в цифровом виде.

Как известно из физики: звук – это волна, состоящая из совокупности колебаний определенных частот. Продолжительность одного колебания определяется частотой дискретизации. Например, 22050 Гц – частота

дискретизации радио, означающая, что в одну секунду произойдет 22050 колебаний определенной частоты. Величина обозначающая громкость одного колебания называется амплитудой. Массив данных, хранящих значения амплитуд, вычисляется с помощью импульсно-кодовой модуляции (ИКМ). Который переводит аналоговый сигнал в цифровой, путем разбиения аналогового сигнала на единицу времени и производя квантование и кодирование (Рисунок 2).

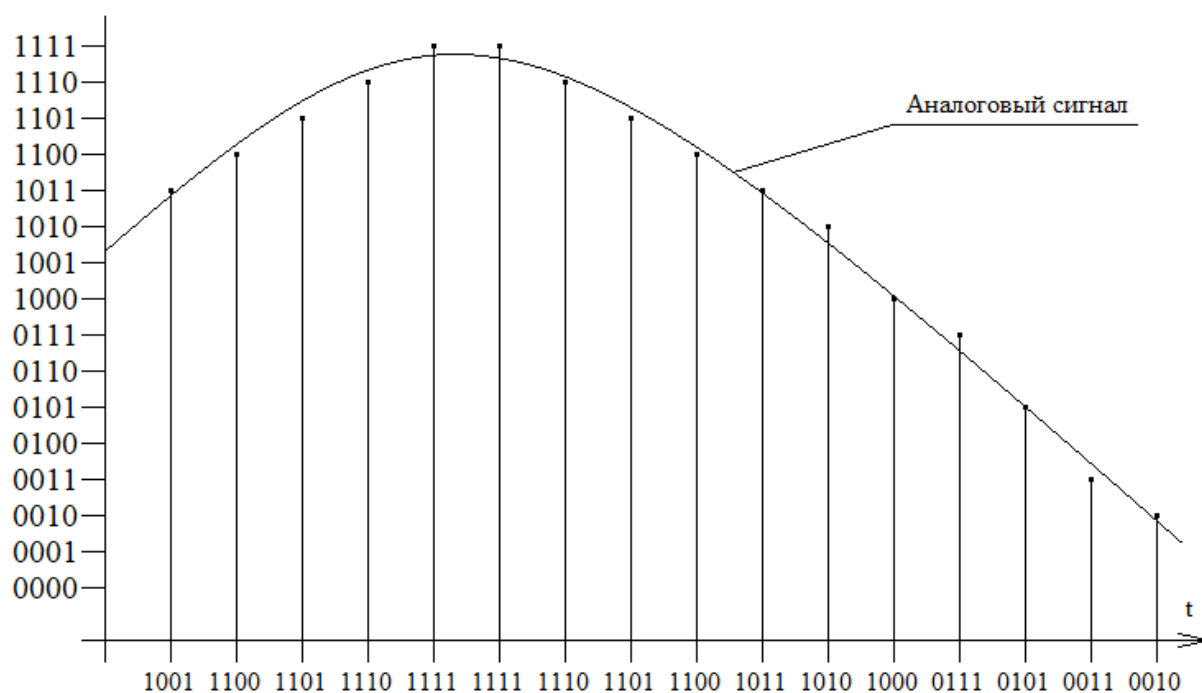


Рисунок 2 - Пример ИКМ размерностью в 4 бита

После импульсно-кодовой модуляции оцифрованный аудиопоток обрабатывается кодеком и записывается в файл.

Основные характеристики аудиопотока:

1. Продолжительность – промежуток времени за который проигрывается аудиопоток;
2. Частота дискретизации – значение определяющие сколько произойдёт колебаний за одну секунду;

3. Количество каналов – параметр, указывающий на количество выходных аудиосигналов.

2.3 Структура видеопотока

Видеопоток по сути является совокупностью быстро меняющихся кадров(изображений), обработанный кодеком (Рисунок 3)[5][6].

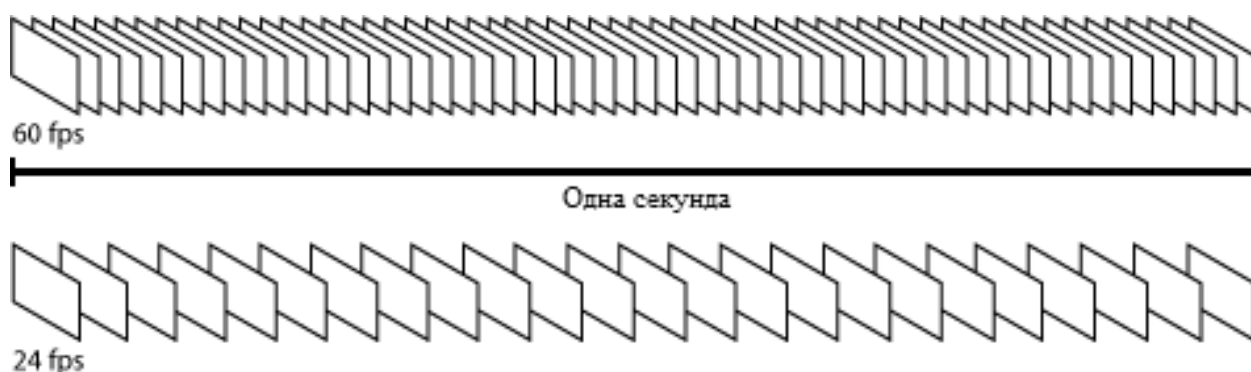


Рисунок 3 - Расположение кадров на единицу времени

Основные характеристика видеопотока:

1. Размерность – совокупность длины и ширины кадров;
2. Продолжительность – промежуток времени за который проигрывается видеопоток;
3. Цветовая модель – модель описания цвета в виде кортежей чисел;
4. Кадровая частота (FPS) – значение указывающие на количество кадров, показываемых за одну секунду.

Существуют разные алгоритмы кодирования, но все они делятся на две основные категории: алгоритмы сжатия с потерями и на алгоритмы сжатия без потерь. Структура видеопотока, обработанного вторым алгоритмом представляет собой чередование кадров, независимые друг от друга. Структура видеопотока, обработанного алгоритмом сжатия с потерями имеет

более сложную структуру. Кадры в этом видеопотоке разделяются на опорные(I), разностные(P) и двунаправленные(B) (Рисунок 4).

Опорные кадры являются независимыми изображениями, начало видеопотока начинается именно с них.

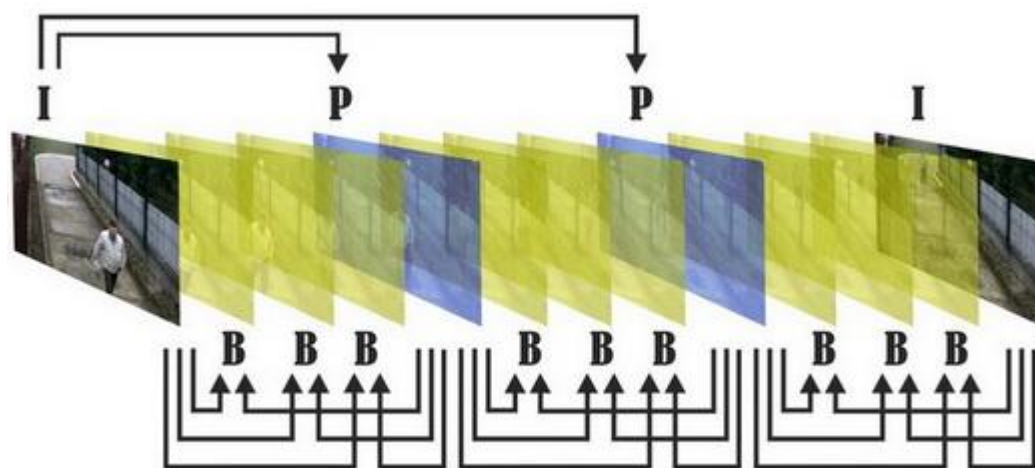


Рисунок 4 – Структура видеопотока, обработанного алгоритмом сжатия с потерями

Между опорными кадрами находятся разностные и двунаправленные кадры, которые могут содержать независимые блоки, а также ссылки на блоки других кадров. Таким образом алгоритм сжатия удаляет избыточную информацию из видеопотока, чем значительно снижает его вес.

3 Описание методов внедрения информации

Так как видеопоток является совокупностью кадров, расположенных в определенной последовательности друг за другом. Кадр видеопотока можно рассматривать как растровое изображение. Поэтому описанный ниже метод распространяется на видеопоток и изображения.

3.1 Предлагаемый метод внедрения скрываемой информации в растровую графику

Перед внедрением информации необходимо декодировать видеопоток на кадры. Предлагаемый нами метод заключается во внедрении информации в эти кадры за счет изменения младших бит информации, отвечающих за цвет. От выбора количества изменяемых бит данных будет зависеть степень визуального обнаружения внедряемой информации. Так как видеофайл, кодированный алгоритмами сжатия без потерь, имеет больший размер, по сравнению с файлом, полученным алгоритмом сжатия с потерями, то мы можем внедрить в этот файл больший объем информации.

Представим внедряемую информацию в виде последовательности бит. Степень внедрения информации обозначим числом от 1 до 24, означающим, сколько бит информации изменяется: 1-незначительное изменение, 24-полностью разрушается исходная информация.

Пронумеруем отвечающие за цвет биты способом, представленным в таблице 1. Число внедряемых бит определяет их расположение в соответствии с этой таблицей. К примеру, для числа бит, равного 5-ти, заменяемые биты отмечены звездочками в таблице 1.

Таблица 1 - Порядок изменений бит информации одного пикселя

RED	24	21	18	15	12	9	6	3*
GREEN	23	20	17	14	11	8	5*	2*
BLUE	22	19	16	13	10	7	4*	1*

Визуальное отображение работы алгоритма показана на рисунке 5.

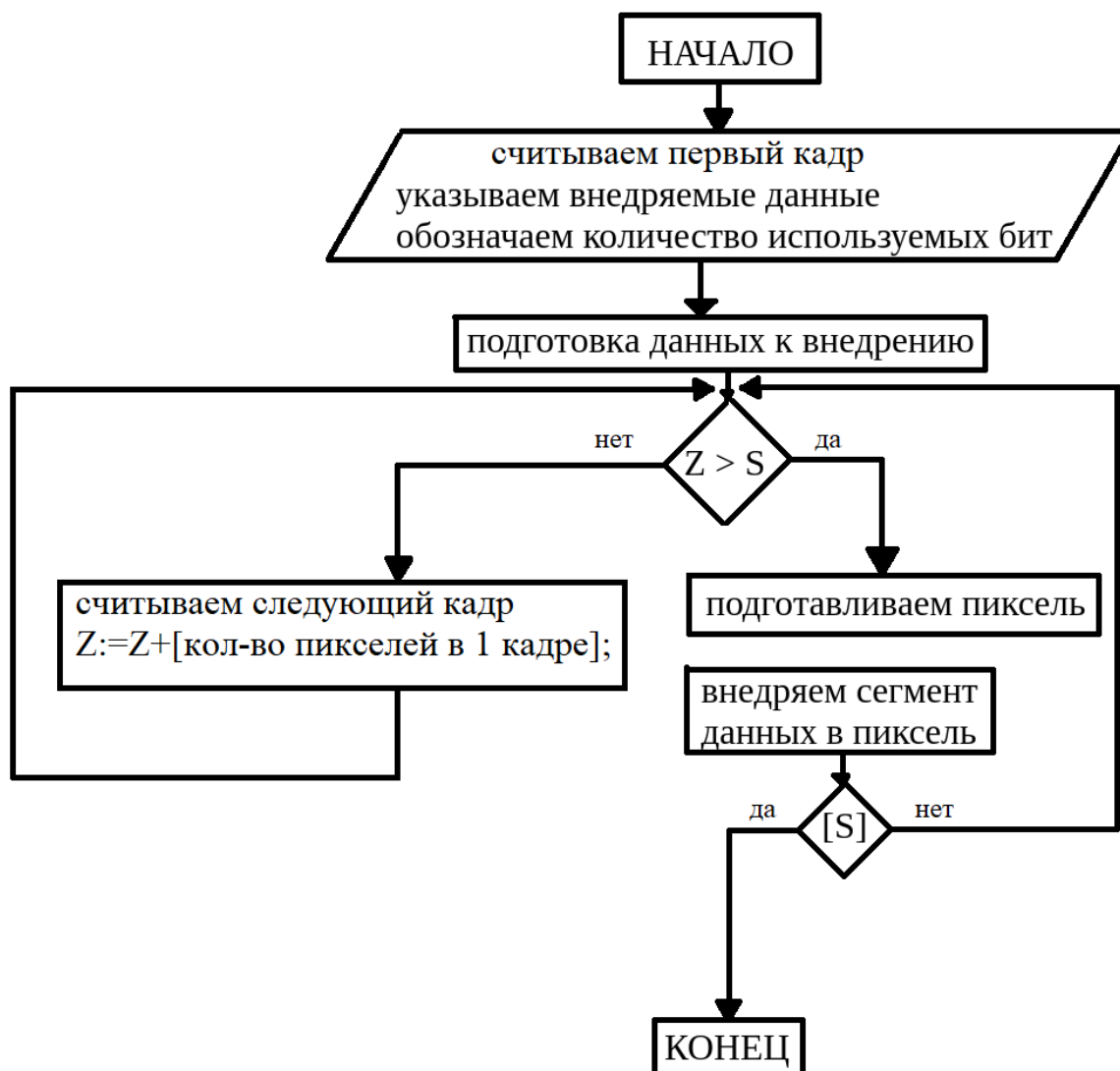


Рисунок 5 – Упрощенная схема алгоритма внедрения

Чем больше бит информации используется в исходном файле, тем больше информации можно внедрить. Не стоит выбирать большие значения, так как это может привести к тому, что видеофайл будет воспроизводиться с очень заметными помехами (Рисунок 6).

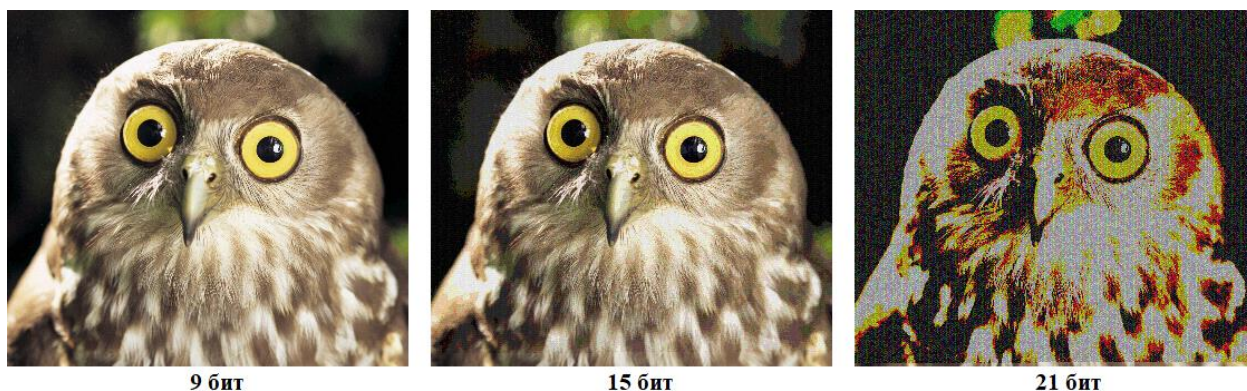


Рисунок 6 - Визуальное изменение кадра, при разном количестве использованных бит информации

Зависимость количества изменяемых бит к надежности скрытия отображена на рисунке 7.

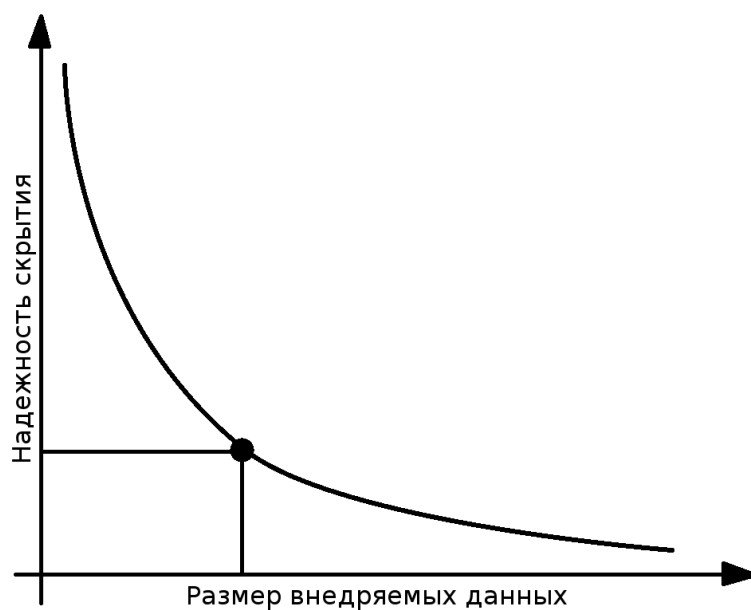


Рисунок 7 - Отношение размера внедряемых данных к надежности скрытия

Разработанный нами метод позволяет внедрять скрытую информацию в растровое изображение, обработанное алгоритмами сжатия без потерь. Количество внедряемой информации вычисляется по формуле(1) для видео и (2) для изображения:

$$S = \frac{x*y*z*t*k}{8} - 20; \quad (1)$$

$$S = \frac{x*y*k}{8} - 20; \quad (2)$$

, где x – ширина кадра/изображения, y – высота кадра/изображения, z – количество кадров в секунду (FPS), t – продолжительность видео в секундах, k – количество используемых бит в одном пикселе, S – количество информации которую можно внедрить в байтах.

3.2 Предлагаемый метод внедрения скрываемой информации в аудиопоток

3.2.1 Структура WAV файла

Практически любой аудиопоток можно конвертировать в wav файл.

Выбор wav файла обуславливается тем, что информация, хранящаяся в нем не обработана алгоритмами сжатия, что позволяет нам взаимодействовать с этой информацией напрямую, без декодирования. Для того чтобы узнать в какие блоки данных мы можем внедрять скрываемую информацию, нам нужно разобрать строение wav файла (Таблица 2)[7].

Таблица 2 – Структура WAV файла

Смещение (размер)	Название	Описание
0 (4 байт)	Chunk Id	Является началом блока RIFF. Содержит "RIFF" (0x52494646).
4 (4 байт)	Chunk Data Size	Размер файла начиная с этой позиции.
8 (4 байт)	RIFF Type	Содержит "WAVE" (0x57415645)
12 (4 байт)	Subchunk Id	Содержит "fmt " (0x666D7420)

Продолжение таблицы 2 – Структура WAV файла

16 (4 байт)	Subchunk Data Size	Размер блока fmt, начиная с этой позиции.
20 (2 байт)	Compression code	Код используемого алгоритма сжатия. В нашем случае должен содержать 1 - PCM/uncompressed.
22 (2 байт)	Number of channels	Число каналов: 1 – моно, 2 – стерео.
24 (4 байт)	Sample rate	Частота дискретизации (22050 Гц, 44100 Гц и т.д.).
28 (4 байт)	Average bytes per second	Количество байт содержащих секунду воспроизведения.
32 (2 байт)	Block align	Количество байт для одного колебания на всех каналах.
34 (2 байт)	Significant bits per sample	Количество используемых бит одного канала для одного колебания. Также называют «точностью звучания».
36 (4 байт)	Chunk Data Id	Содержит "data" (0x64617461).
40 (4 байт)	Chunk Data Size	Размер блока data, начиная с этой позиции.
44 (...)	Data	Блок данных.

Следует отметить, что перед блоком «data» также может находиться служебная информация, например, название используемого декодера (Рисунок 8).

00000046	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	49	46	46	46	10	8f	02	57	41	56	45	66	6d	74	20	RIFFE.Ц.WAVEfmt
00000010	10	00	00	00	01	00	02	00	44	ac	00	00	10	b1	02	00	D.....±..
00000020	04	00	10	00	4c	49	53	54	1a	00	00	00	49	4e	46	4f	LIST....INFO
00000030	49	53	46	54	0e	00	00	00	4c	61	76	66	35	37	2e	37	ISFT....Lavf57.7
00000046	32	2e	31	30	30	00	64	61	74	61	00	10	8f	02	00	00	2.100.3data..Ц...
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Рисунок 8 - Заголовок WAV файла в HEX редакторе

3.2.2 Описание метода внедрения

Для внедрения информации мы будем использовать блок «data» в котором находится информация о звуковых амплитудах, находящихся в определенном порядке. Для примера приведем аудиопоток содержащий 3 канала и с 16-ти битной точностью звучания (Рисунок 9).

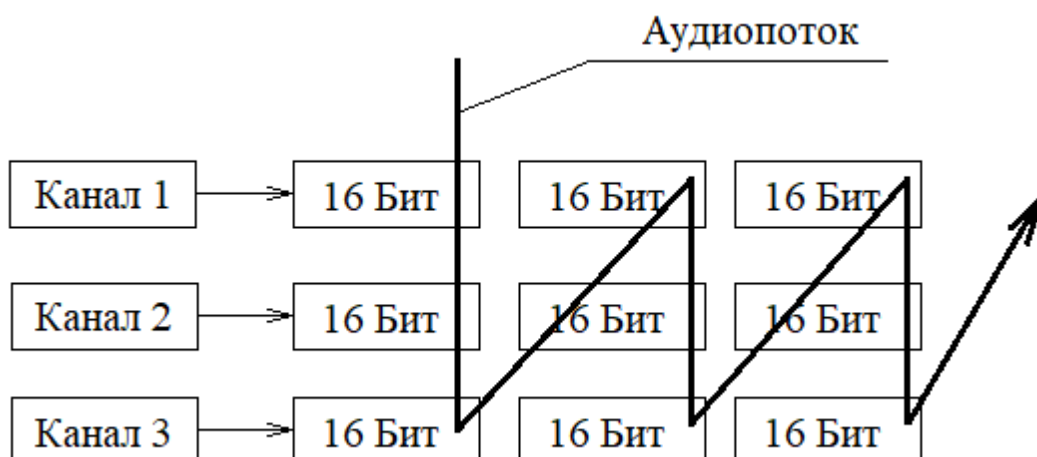


Рисунок 9 – Порядок расположения блоков, содержащих амплитуду

Для незаметного внедрения информации нами предлагается изменять младшие биты блоков, содержащих амплитуду. Иными словами, мы разбиваем внедряемые данные на блоки по 2 бита и записываем их в младшие биты 16-ти битного блока.

Разработанный нами метод позволяет внедрять скрытую информацию в аудиопоток, обработанный алгоритмами сжатия без потерь. Количество внедряемой информации вычисляется по формуле(3):

$$S = (8 * y * z/x)/8 - 20; \quad (3)$$

, где x – Significant bits per sample, y – Chunk Data Size, z – количество используемых бит в одном колебании одного канала, S – количество информации которую можно внедрить в байтах.

После внедрения скрываемой информации в аудиопоток, мы получаем такое его звучание, которое схоже с оригиналом, без посторонних шумов и помех. Пример использования алгоритма показан на рисунке 10.

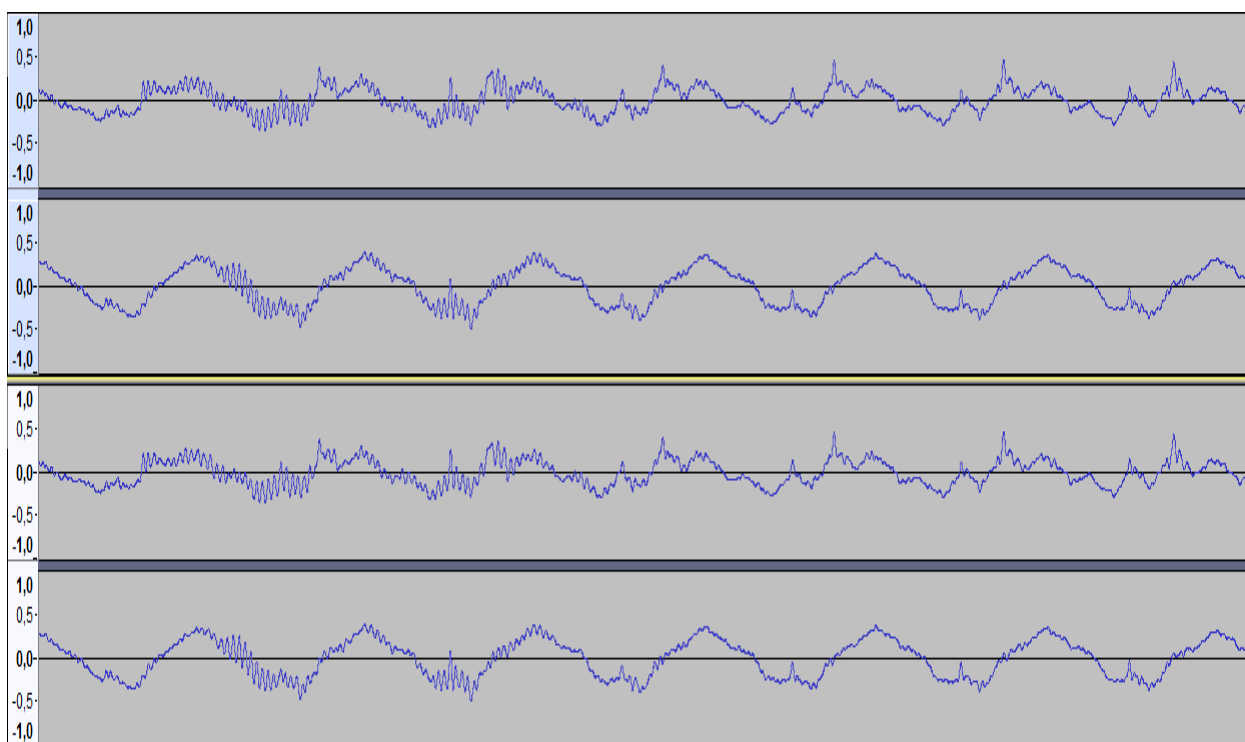


Рисунок 10 – Сравнение исходного и измененного аудиопотока

На рисунке выше показан двухканальный аудиопоток, сверху – исходный, внизу – с измененными битами. Как можно увидеть на рисунке, потоки практически одинаковы. Учитывая, что нам нет необходимости кодировать и декодировать аудиопоток, мы внедряем информацию непосредственно в файл, что происходит намного быстрее, как при внедрении, так и при извлечении скрываемой информации.

3.4 Предлагаемый метод внедрения скрываемой информации в видеопоток

Внедрение информации в видеопоток, обработанный алгоритмом сжатия без потерь аналогичен внедрению информации в растовое изображение. Процесс состоит из трех этапов:

1. Декодирование видео;
2. Внедрение или извлечение скрываемой информации;
3. Кодирование видео.

Такой метод не подойдет для внедрения информации в видеопоток, обработанный алгоритмом сжатия с потерями.

Проблема состоит в том, что при кодировании информации теряется часть данных, а другая часть изменяется и оптимизируется для достижения наибольшего сжатия с минимальными потерями качества. Если использовать алгоритм внедрения скрываемой информации, описанные выше, то при обработке его кодеком, мы потеряем эту информацию.

Для решения данной задачи были изучены принципы кодирования видео. На основе чего нами был разработан следующий алгоритм внедрения информации в видеопоток:

1. Декодирование видео;
2. В полученных кадрах, пиксели разбиваем на матрицы размером 8x8;
3. Производим считывание параметра яркости первой матрицы;
4. Вычисляем усредненное значение яркости всей матрицы;
5. Вычисляем разность необходимых изменений в яркости для внедрения скрываемой информации;

6. Производим корректировку весов яркости в массиве на основе значения, полученного на предыдущем шаге;
7. Переходим к шагу 2 и обрабатываем следующую матрицу кадра, если таких матриц не осталось переходим к следующему кадру;
8. После внедрения всей информации, кодируем видео кодеком.

В предложенном алгоритме стоит учитывать то, что одним из этапов кодирования видео является квантование. Именно на этом этапе происходит потеря и преобразование информации. Уровень сжатия контролируется параметром квантования, чем выше этот уровень, тем сильнее искажается исходная информация в пользу её размера, что затрудняет внедрение скрываемой информации.

Внедрение скрываемой информации происходит в младшие биты усредненного значения яркости, обрабатываемой матрицы. Так как это значение является вещественным числом, то мы будем использовать округление до целого числа. Данные мы будем внедрять в самый младший бит, а остаток мы максимально приблизим к целому числу. При декодировании видеопотока на кадры мы переводим изображения цветовой модели YUV в цветовую модель RGB. На основе чего мы корректируем данные цветов, опираясь на формулу (4).

$$Y = R * 0.299 + B * 0.114 + G * 0.587; \quad (4)$$

, где Y – яркость, R – значение красного цвета, B – значение синего цвета, G – значение зеленого цвета.

Таким образом мы можем максимально подготовить наш внедренный бит информации к изменениям, которые произведет кодировщик при обработке видео. Пример работы изображен на рисунке 11.

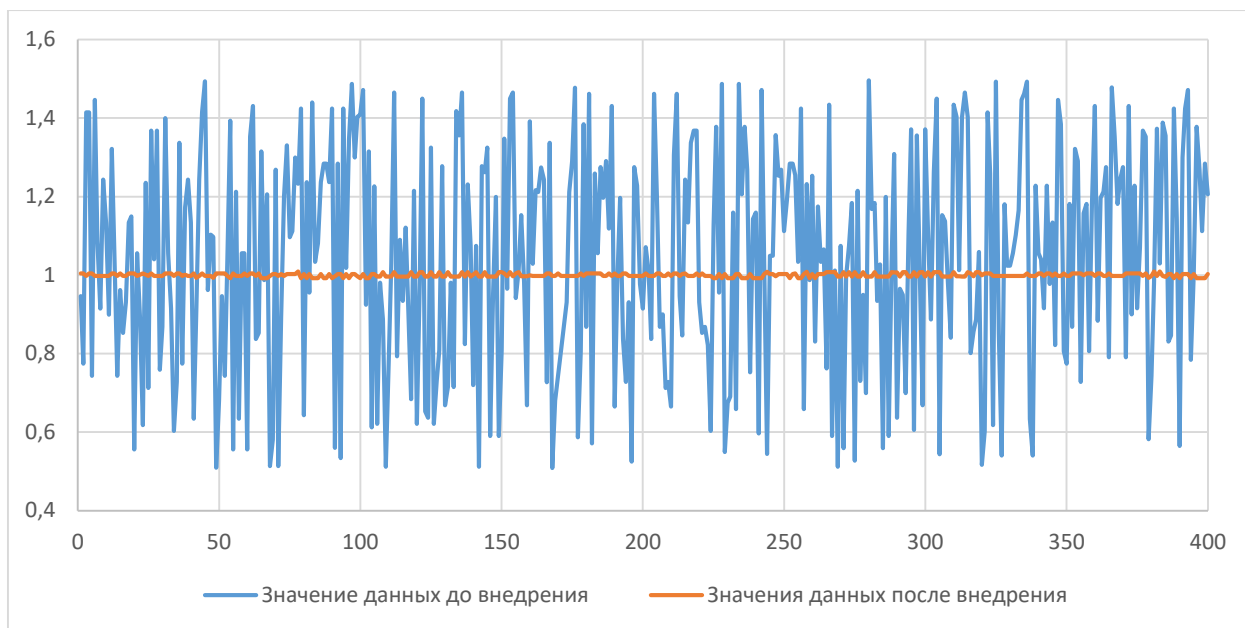


Рисунок 11 – Значения остатка яркости до и после внедрения скрываемой информации

На рисунке выше синем цветом отображены значения остатка усредненной яркости, после работы алгоритма, эти значения имеют вид отображенные оранжевым цветом.

После внедрения информации происходит кодирование в видеофайл и внедренные данные также подвергаются обработки и оптимизации. В результате мы получаем видеофайл, при декодировании которого мы можем извлечь внедренную информацию. Результат работы алгоритма показан на рисунке 12, здесь оранжевым отображен график усредненных значений яркости до обработки кодеком, синем отображен график, значений, которые мы получили после кодирования.

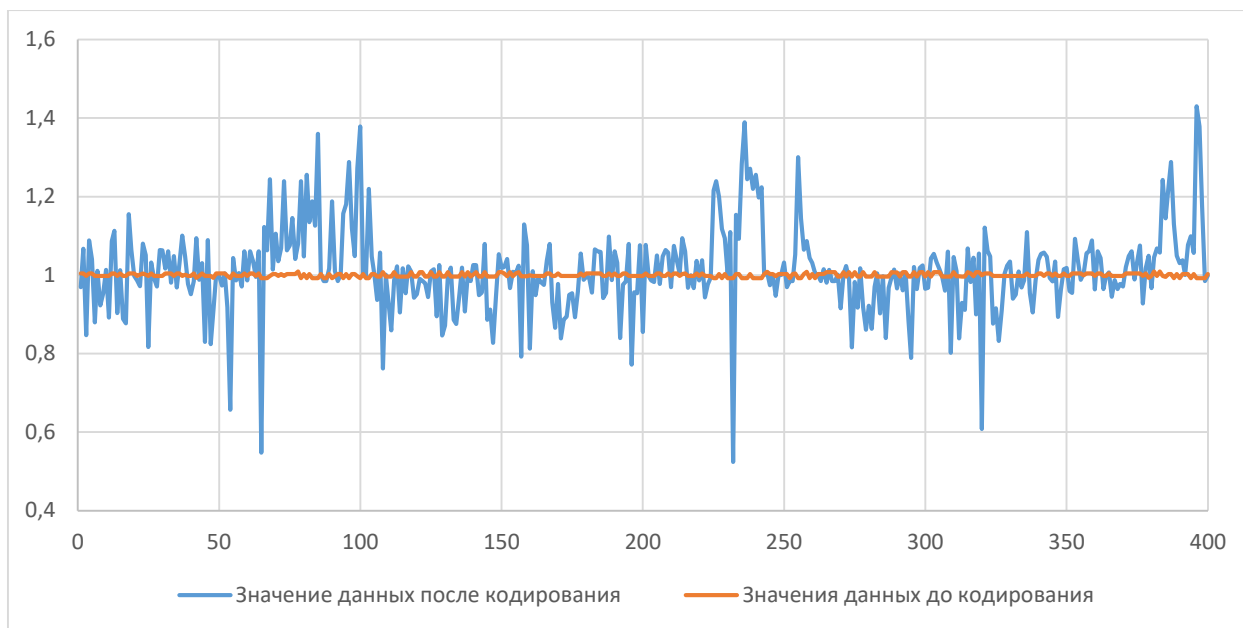


Рисунок 12 – Значения данных до и после обработки видео кодеком

Так как мы внедряли информацию в самый младший бит, то мы знаем, диапазон остатка, при котором значение младшего бита измениться на противоположный: $0.5 < x < 1.499$, где x – это значение остатка усредненной яркости. На графике, отображенном на рисунке выше, можно заметить скачки очень приближенных к этой границе, в случае если хотя бы одно значение превысит допустимый диапазон мы потеряем данные которые внедрили. Для того, чтобы этого не произошло были произведены эксперименты, на предмет выявления оптимальные значений квантования и использование определенных бит в яркости.

В виду того, что мы не можем точно предугадать преобразованные данные после квантования, точность сохранности внедренной информации была получена экспериментальным путем.

Целью экспериментов являлось получить оптимальный показатель квантования при изменении определенного числа бит во входных данных, при которых можем эти данные извлечь после кодирования и декодирования видео.

На рисунке 12 было произведено внедрение информации в самый младший бит и с указанием значения квантования равного 1, даже при таких параметрах, после кодирования видео, значение яркости вышло за пределы допустимых значений, и мы потеряли несколько бит. В дальнейшем решено использовать второй бит в качестве носителя скрываемой информации, таким образом диапазон допустимых значений увеличился. На рисунке 13 отображена работа алгоритма с внедрением скрываемой информации во второй бит и значение квантования равного 8.



Рисунок 13 – Пример внедрения информации

На рисунке выше диапазон допустимых значений составляет: $-0.5 < x < 1.49$, при внедрении информации корректировка значений идет в 0.5, отображенная оранжевой линией.

Нахождение значения квантования, необходимого для целостного извлечения скрываемой информации, мы получили, анализируя данные, полученные после работы протокола с разными значениями квантования. Было произведено несколько циклов внедрения файла размером 7282 байта в один и тот же видеофайл. В таблице 3 показаны количество найденных

несоответствий исходных данных и данных полученных после внедрения и последующего извлечения из видеопотока.

Таблица 3 – Количество ошибок при разных значениях квантования

Значение квантования при кодировании	1-11	12	13	14	15
Количество найденных несоответствий	0 байт	15 байт	66 байт	102 байт	207 байт

На основе полученных данных был сделан вывод: для получения целостных внедренных данных, стоит использовать значение квантования не больше 11.

3.3 Методы обнаружения внедренной информации

3.3.1 Обнаружение скрытой информации в аудиопотоке

Существуют несколько методов обнаружить информацию в файле-контейнере:

- 1) Сравнить исходный файл (если такой имеется) и файл с внедренными данными, можно сравнивать спектограммы, но изменения между двумя файлами видны только большим увеличением (Рисунок 14), гораздо эффективнее использовать HEX редактор;
- 2) Воспринимаемая ухом акустическая информация, с выявлением шума в аудиопотоке;
- 3) Считывание нескольких бит информации из областей предположительного внедрения, для последующего сравнения их с

файловыми сигнатурами. Например, рассмотрим часть данных из блока «data», значение Significant bits per sample=8:

FC FE FD FD FD FD FC FC FD FC FD FC FD FC FD FE FC FE FF FD FC
FF FC FD FC FE FF FE FC FF FD FD, считав по два бита младших
бита с каждого байта, мы получим массив:

00100101 01010000 01000100 01000110 00101101 00110001 00101110
00110101, переводя в шестнадцатеричную систему и произведя
соответствие с таблицей ASCII получим следующее:

25 50 44 46 2B 31 35 => %PDF-1.5, что означает начало внедренного
файла PDF.

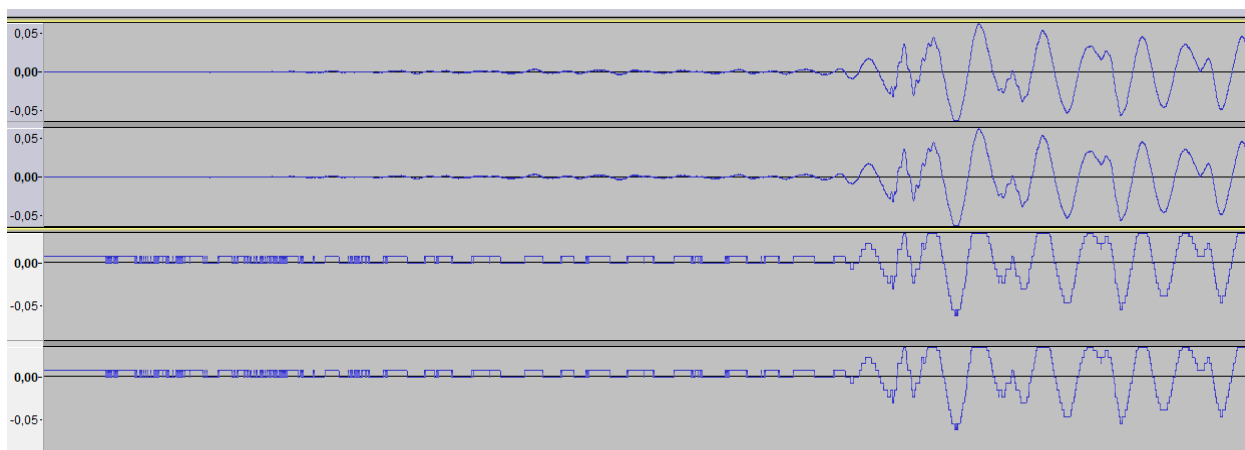


Рисунок 14 - Сравнение спектрограмм с внедрёнными данными и без них

3.3.2 Обнаружение скрытой информации в видеопотоке

При малейшем подозрении на то, что в видеофайле находится скрытая информация, применяются методы её обнаружения. Самым распространённым из них является анализ срезов кадров.

Срез кадра – это изображение, полученное путем использования только одного из бит цветовой матрицы. Иными словами, изображение строится на основе одного из двадцати четырех бит, отвечающих за цвет пикселя. Если бит равен единице, то окрашиваем его белым цветом, если нулю - черным. В итоге мы получаем одноцветное изображение среза кадра. В примере приведен срез

кадра без внедренной информации (Рисунок 15). Визуально, срезы старших бит выглядят более похожими на силуэты исходного кадра, чем срезы младших бит[8].



Рисунок 15 - Пример среза кадра

При анализе срезов кадров, можно убедиться в том, что в видеофайле имеется скрытая информация, если обнаружатся какие-либо отклонения от обычного представления.

Приведем пример обнаружения данных в срезах (Рисунок 16).

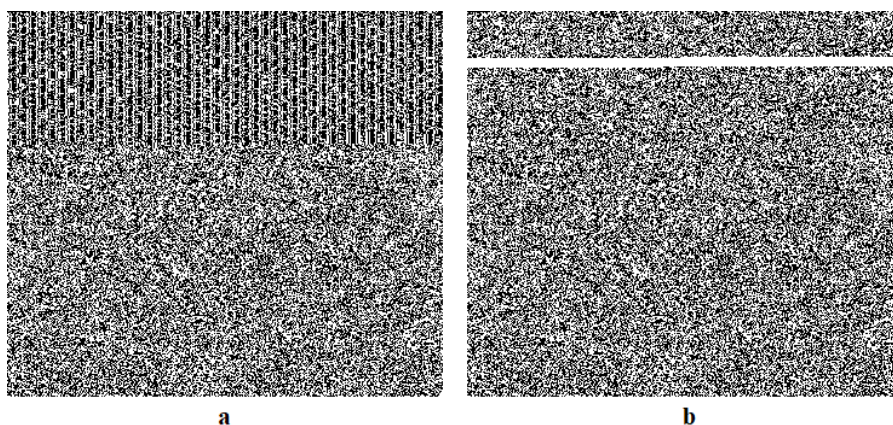


Рисунок 16 - Примеры обнаружения внедренных данных в срезах

На срезе «а» видно, что верхняя часть контрастно отличается. Это означает, что на данном кадре заканчивается внедряемая информация. Данные срезы были сделаны по значению младшего бита синего цвета, и их хаотичность отличается от хаотичности внедренных данных.

Поскольку рассмотренный метод позволяет ограничить требования к скрываемой информации только её размером, то возможна ситуация, при которой внедряемые данные содержат последовательность повторяющихся байтов. В этом случае наблюдается картина, изображенная на срезе «b», представляющая пробел, состоящий из нулей или единиц.

Известен также другой метод обнаружения скрытой информации. Он заключается в считывании небольшого участка кадра, формируя массивы из всей возможной комбинации бит. Одной из программ, позволяющих проводить анализ срезов кадра является StegSolve. В полученных массивах производится поиск, по ключевым словам, если встречается последовательность вида: «7z», «ID3», «PDF» и др., означающая начало внедренного файла (Рисунок 17). В данном примере демонстрируется обнаружение данных, путем считывания одного бита синего цвета. В первой строке находится надпись «PDF -1.4.», что указывает на внедренный PDF-документ. Если встречаются осмысленные сообщения, то предполагается, что это часть текстового файла, либо текстового сообщения.

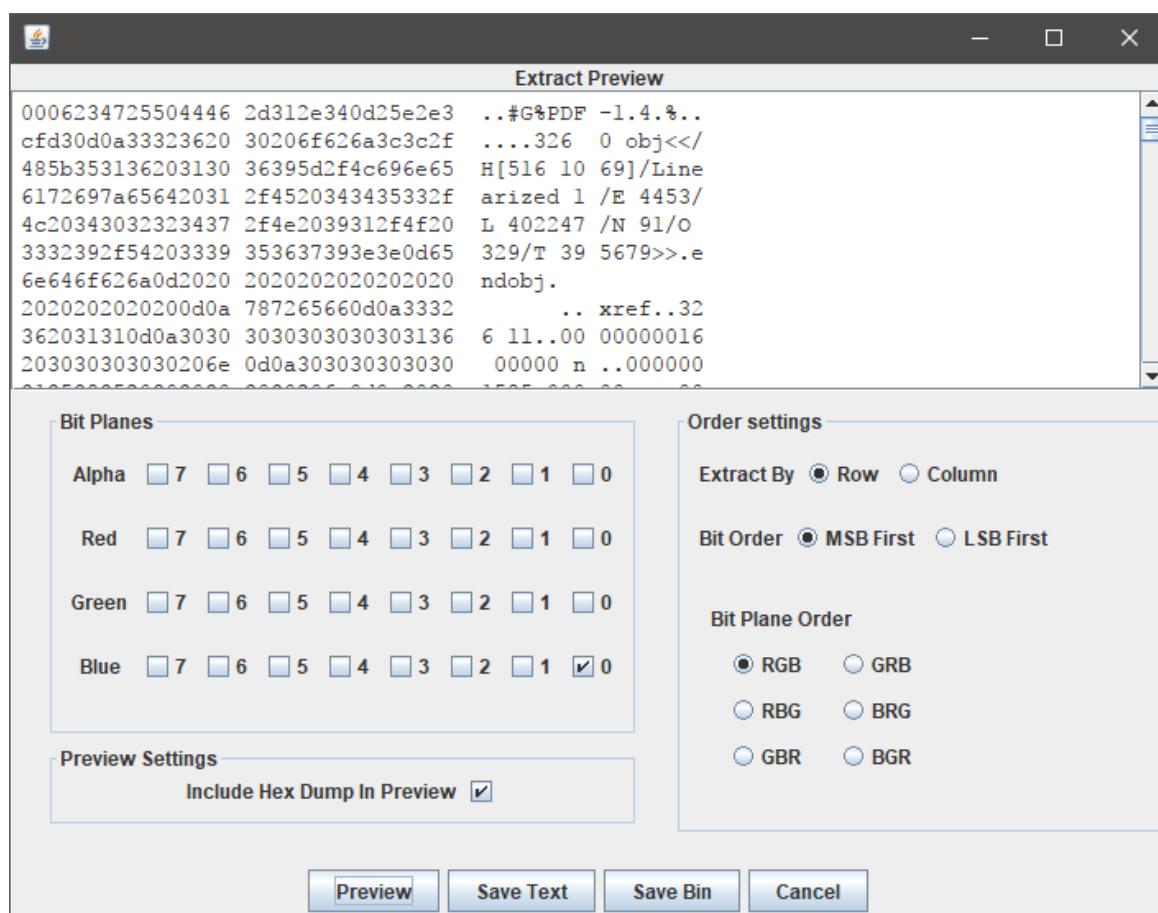


Рисунок 17 - Обнаружение внедренной информации, путем извлечение данных из одного бита синего цвета

3.4 Разработанный метод затруднения обнаружения внедренной информации

Чтобы уменьшить шансы обнаружения скрываемых данных и избежать ситуаций, описанных выше, эти данные нужно определенным образом подготовить. Нами предлагается следующий подход.

Размер внедряемого файла записываем в первые 4 байта.

Выполняем вычисление MD5 хеша внедряемых данных и записывается в 16 байт после блока размера данных (Рисунок 18).

Производим вычисление хеша из полученных данных в результате конкатенации вычисленного в начале хеша и размера.

Полученный результат записываем в массив размером в один мегабайт и повторяем операцию, используя хеш полученный на предыдущем этапе, пока массив не будет заполнен.

Производим операцию исключающего или (XOR) между внедряемыми данными и полученным ранее массивом (Рисунок 19).

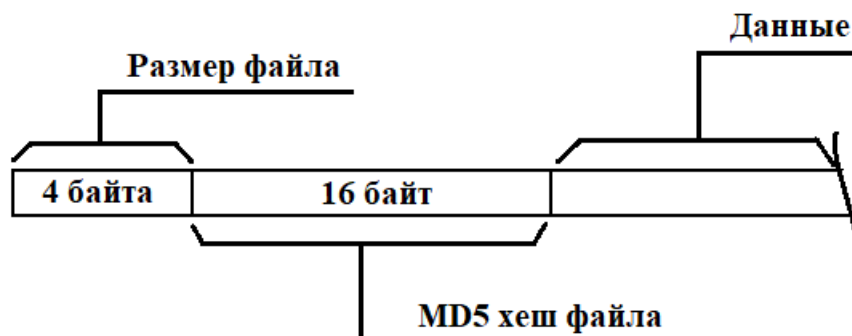


Рисунок 18 - Структура внедряемых данных

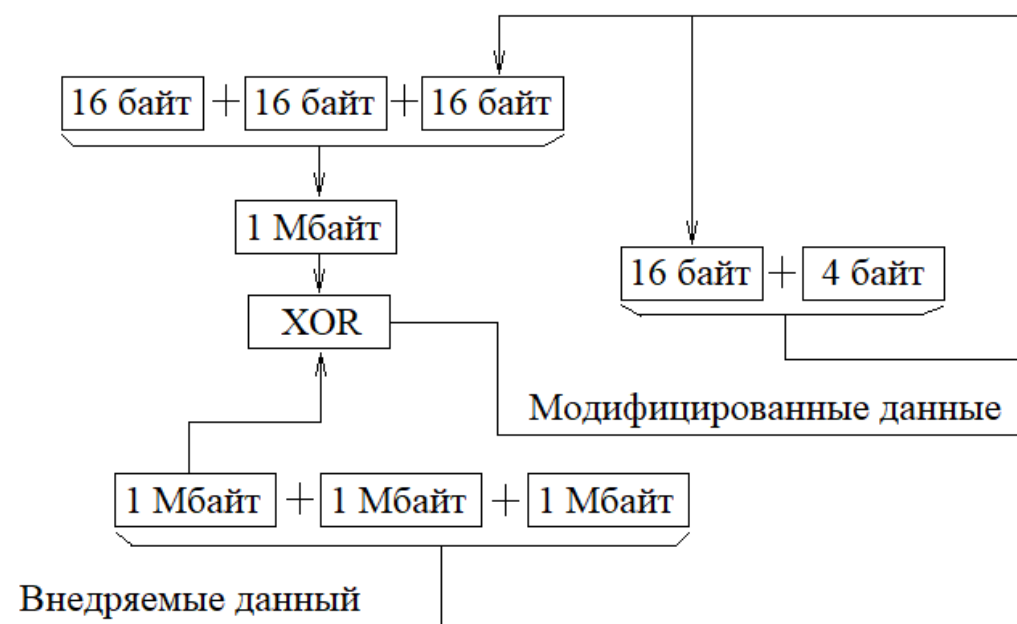


Рисунок 19 - Схема отображающая процесс модификации внедряемых данных

Эти манипуляции необходимы для того, чтобы во внедряемых данных отсутствовали повторяющиеся байты и заголовочные слова файлов (Рисунок 19). Предложенный подход делает практически невозможным обнаружение

внедренной информации. Результат предложенного метода демонстрируется на рисунке 21, отображающем те же внедренные данные, что и на срезах рисунка 16.

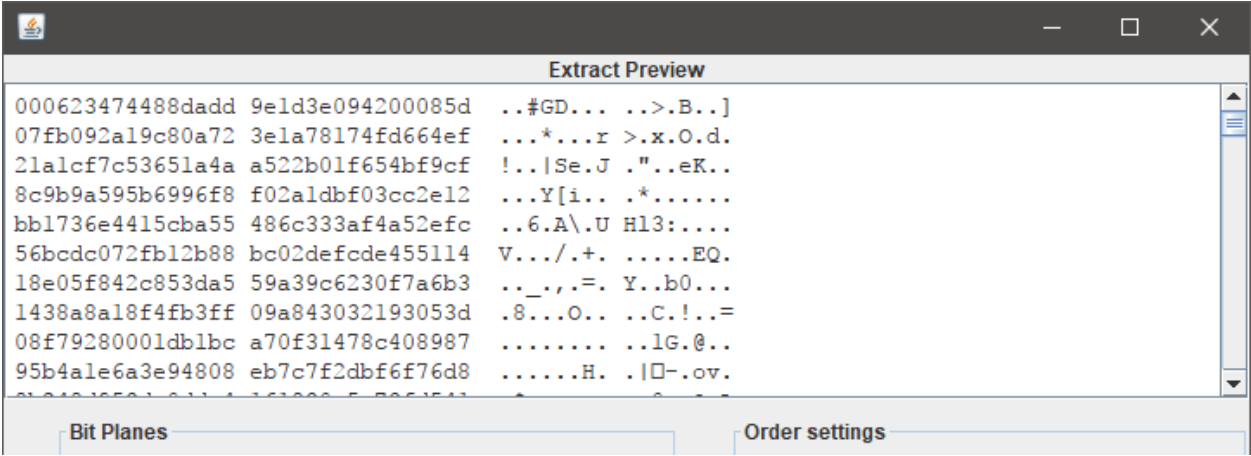


Рисунок 20 - Отображение модифицированного PDF файла, используя один бит синего цвета

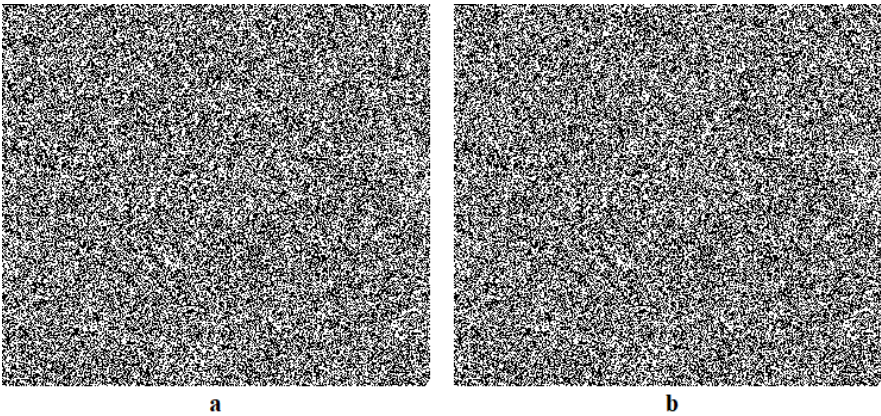


Рисунок 21 - Пример среза с обработанными алгоритмом, данными

4 Описание разработанного программного продукта

4.1 Анализ существующих решений

При изучении предметной области, был произведен обзор аналогичных программных продуктов, которые не раскрывают своего методы внедрения информации. Самые распространённые из тех, что предоставляют возможность внедрять данные в видео представлены в таблице 4, с описанием отличий от разработанной нами программы.

Таблица 4 – Список аналогичных программ

Название программы	Выявленные недостатки
DeEgger Embedder v1.21	Внедряет информацию методом конкатенации, иными словами выходной файл будет размером суммы размера исходного файла и размера внедряемых данных.
OpenPuff v4.00	Самая многофункциональная из рассмотренных программа. Несмотря на то, что при внедрении информации указывается на mp4 файл, информация внедряется вовсе не в видео поток, а в звуковую дорожку. При попытке внедрить информацию в mp4 файл без звуковой дорожки выдает сообщение о недостаточном объеме для внедрения информации.

Продолжение Таблица 4 – Список аналогичных программ

StegoStick v1.0	Очень медленная программа, так и не дождался её завершения. Судя по тому, что при внедрении информации не было ограничений в размере, выдвигаю предположение: что данная программа использует меток конкатенации файлов.
-----------------	--

Это не полный список рассмотренных программных продуктов. При выборе стеганосистемы надо обращать внимание на то, что метод сокрытия данных не может быть надежным, если при выборе файла контейнера не происходит его анализ на предмет, объема, который можно внедрить в файл-контейнер.

Также были рассмотрены программные продукты, которые позволяли внедрять информацию в WAV файлы и изображения. У всех рассмотренных программ не было возможности выбрать количество изменяемых бит.

4.2 Используемые технологии

В ходе реализации программного продукты были использованы следующие инструменты и технологии:

1. ObjectPascal[9] – объектно-ориентированный язык программирования, разработанные в фирме Apple Computer в 1986 г.;
2. Free Pascal – компилятор языка ObjectPascal, является свободно распространяемым и кроссплатформенным инструментом;
3. Lazarus[10] – открытая, кроссплатформенная среда разработки приложений, используя язык ObjectPascal;
4. FFmpeg[11] – набор свободных библиотек, позволяющих кодировать и декодировать аудио и видео.

4.3 Почему именно эти технологии

Компилятор Free Pascal был выбран в связи с тем, что он позволял создавать кроссплатформенное программное обеспечение, является свободно распространяемым продуктом с открытым исходным кодом. Благодаря чему данный компилятор получил широкую популярность и быстрым развитием благодаря сообществу разработчиков.

Выбор интерактивной среды разработки Lazarus обусловлен тем, что он является кроссплатформенным программным продуктом с открытым исходным кодом. Также в него уже интегрирован компилятор FreePascal. Данная интерактивная среда разработки предоставляет набор инструментов, упрощающих этапы разработки программных продуктов, одним из которых является упрощенный редактор форм разрабатываемого приложения.

FFmpeg набор библиотек кроссплатформенных библиотек с открытым исходным кодом. Используется для кодирования и декодирования аудио и видео, а также для извлечения кадров из видеопотока.

Все используемые программные продукты являются кроссплатформенными и позволяют создать программу для разных операционных систем.

4.4 Требования, предъявляемые к программному продукту

При создании программного продукта должны быть выполнены следующие требования:

1. Принимать на вход любой видеофайл;
2. Внедрять скрываемую информацию в изображения формата BMP;
3. Внедрять скрываемую информацию в аудиофайл формата WAV;
4. Возможность обрезать фрагмент видеофайла, а также указать разрешение выходного видеофайла;

5. Модифицировать внедряемые данные для затруднения их обнаружения и увеличить стойкость к большинству атак на стегосистемы;
6. Возможность указать количество изменяемых бит в исходном файле;
7. Возможность указать кадровую частоту (FPS) в выходном видеофайле;
8. Производить вычисления объема информации, который доступно внедрить в файл-контейнер;
9. Возможность указать кодек выходного файла для аудио и видео;
10. Надежное внедрение скрываемой информации и гарантированное извлечение с учетом того, что файл-контейнер не был поврежден.

4.5 Описание формата входных и выходных данных

В качестве внедряемых входных данных может служить файл любого типа, ограничение имеет только допустимый размер, вычисленный при анализе файла-контейнера.

Входные данные для аудио, видео и изображении, в качестве файла-контейнера могут использоваться любые соответствующие форматы. Данные отличенные от выходного формата файла-контейнера будет автоматически переконвертированы.

Выходным файлом-контейнером для изображения используется формат BMP.

Для звукового файла-контейнера используется: WAV.

Выходной файл-контейнер для видео служат форматы AVI и MP4.

Список кодеков, которые обрабатывают выходной файл-контейнер:

HuffYUV – видеокодек, сжатие происходит путем предсказания следующего пикселя и кодирования алгоритмом Хаффмана;

OpenJPEG[12] – видеокодек, широко используется в медицине, где необходима высокая точность изображения, в основе лежит дискретное вейвлет-преобразование;

WAV – формат в основном используются для хранения, записи и воспроизведения аудиоданных высокого качества.

4.6 Руководство пользователя

При запуске программы появляется главное окно (Рисунок 22).

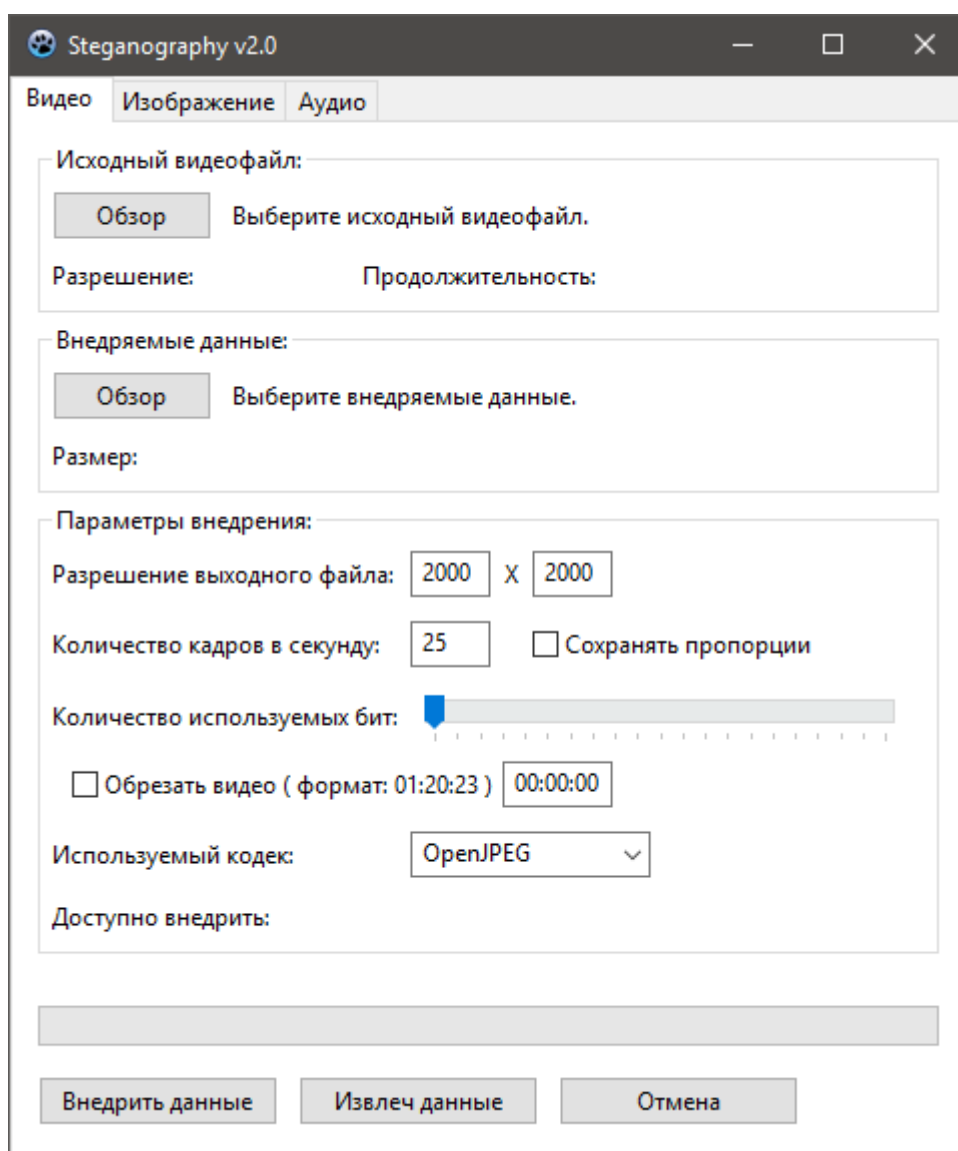


Рисунок 22 – Окно программы

Перед внедрением скрываемой информации необходимо определиться с выбором файла-контейнера. Программа позволяет внедрять информацию в видео, аудио и изображения. Определившись с выбором нажмите на соответствующую вкладку вверху окна программы.

Для каждого метода внедрения имеются свои параметры. Три из них присущие всем методам обязательны к указанию, это:

- 1) Выбрать исходный файл-контейнер;
- 2) Выбрать внедряемый файл, который мы хотим скрыть;
- 3) Указать количество используемых бит.

Для аудио и видео можно указать кодек, которым будет обрабатываться выходной файл.

Для видео можно также указать кадровую частоту (FPS), разрешение видеопотока и обрезать видео до определенного момента.

При указании всех параметров необходимо запустить процесс внедрения нажав на кнопку «Внедрить данные», появится диалоговое окно сохранения файла с внедренной информацией, можно перезаписать исходный файл. Затем запустится процесс внедрения (Рисунок 23), длительность которого зависит от объема файла-контейнера и размера внедряемых данных.

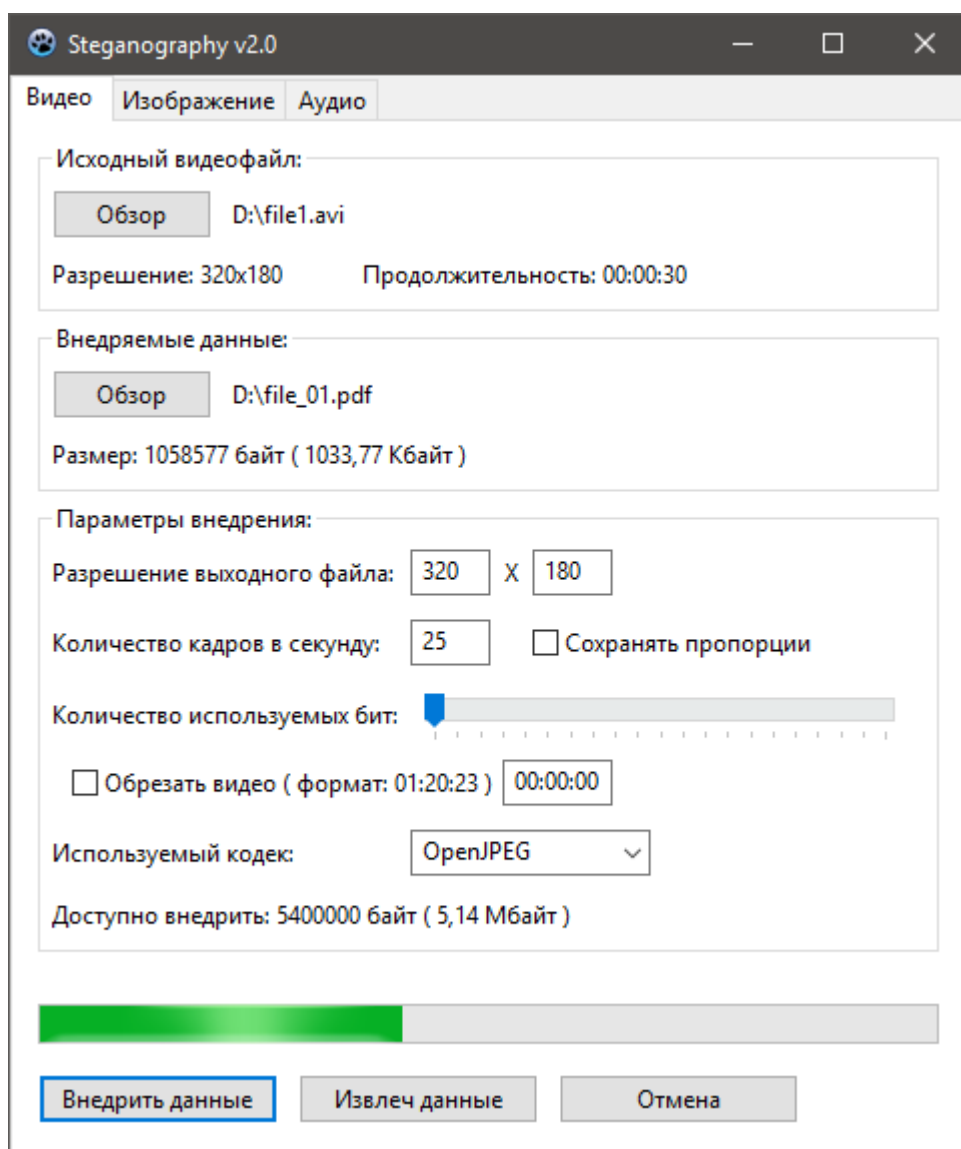


Рисунок 23 – Процесс внедрения данных

Извлечение скрытой информации происходит путем выбора исходным файлом файл-контейнер с внедренными данными и указанием количества бит, указанных при внедрении информации. После чего появится диалоговое окно для сохранения скрытого файла.

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы были изучены принципы оцифровки медиаданных, а также изучена структура хранения оцифрованных данных в файле. Изучены методы кодирования и декодирования мультимедийной информации.

Разработаны методы внедрения скрываемой информации в медиафайл, а также методы модификации внедряемых данных, для затруднения их обнаружения.

Разработана программа, позволяющая внедрять файлы любых типов в видео файл формата AVI и MP4, звуковой файл формата WAV и изображения формата BMP. Извлечение данных происходит путем обработки файла контейнера. При этом необходимо указать лишь количество бит использованных при внедрении информации. Также реализован алгоритм внедрения информации в видеопоток, кодированный алгоритмом сжатия с потерями. Разработаны и реализованы методы преобразования внедряемых данных, для повышения уровня сокрытия. После внедрения информации обнаружить её практически невозможно.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Барабаш А. Стеганография. Древняя тайнопись в цифровую эпоху / А. Барабаш. – М. :Наука, 2009. – 67 с.
2. Грибунин В. Г. Стеганографические системы. Критерии и методическое обеспечение / В. Г. Грибунина, В. Н. Фомченко, А. П. Мартынов. – М. : Мир, 2016. — 324 с.
3. M. Giovanni Handbook of Data Compression / M. Giovanni, D. Salomon, ; под ред. D. Bryant. – 5-ое изд. – М. : Springer, 2009. – 18 с.
4. Гребенников О. Ф. Основы записи и воспроизведения изображения / О. Ф. Гребенников, В. В. Раковский, Н. К. Игнатьев. — М. : Искусство, 1982. — 239 с.
5. Michael W. Image Compression Fundamentals, Standards and Practice / W. Michael, S. David. — М. : Kluwer Academic Publishers, 2001. — 776 с.
6. ISO/IEC 15444-3 – 2007. Information technology - JPEG 2000 image coding system. – М. : International Organization for Standardization, 2007.
7. Smith T. Microsoft readies MP3-killer digital music format / T. Smith. – М. : Science, 2007.
8. Goljan M. Searching for the Stego Key / M. Goljan, D. Soukal // Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI. – 2014. – С. 70-82.
9. Буч Г. Объектно-ориентированное проектирование с примерами применения / Г. Буч. – М. : Конкорд, 1992. — 519 с.
10. Кучер Т. В. Учебник по программированию Free Pascal и Lazarus / Т. В. Кучер, Е. Р. Алексеев, О. В. Чеснокова. – М.: Альт Линукс, ДМК Пресс, 2010. — 440 с.
11. FFmpeg Developer Documentation [Электронный ресурс] //URL: <http://www.ffmpeg.org/developer.html> (дата обращения: 14.04.2017).
12. OpenJPEG Documentation [Электронный ресурс] //URL: <http://www.openjpeg.org/doxygen> (дата обращения: 13.05.2017).

13. Афанасьев В.И. "Стеганография в видеофайлы" // сб. материалов 5-ой Международной молодежной научной конференции "Математическое и программное обеспечение информационных, технических и экономических систем". – Томск, 2017.

Отчет о проверке на заимствования №1

Автор: d7t4@gmx.com / ID: 5790367

Проверяющий: (d7t4@gmx.com / ID: 5790367)

Отчет предоставлен сервисом «Антиплагиат»- <http://www.antiplagiat.ru>

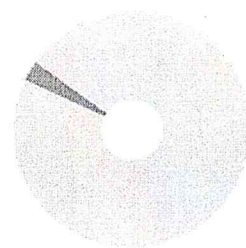
ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 4
Начало загрузки: 07.06.2018 09:21:49
Длительность загрузки: 00:00:01
Имя исходного файла: VKR
Размер текста: 1615 кБ
Символов в тексте: 42739
Слов в тексте: 4830
Число предложений: 278

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 07.06.2018 09:21:50
Длительность проверки: 00:00:02
Комментарии: не указано
Модули поиска:

ЗАИМСТВОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
3,14%	0%	96,86%



Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	1,31%	1,84%	Российской Федерации (ми...	http://lib.exdat.com	18 Апр 2016	Модуль поиска Интернет	11	14
[02]	0,12%	1,18%	ДП СМК НУ ТГУ 05.10.06.2010.	http://geo.tsu.ru	раньше 2011	Модуль поиска Интернет	1	8
[03]	0,59%	0,59%	Шаблоны титульных листов...	http://tsu.ru	22 Сен 2016	Модуль поиска Интернет	2	2

Еще источников: 4

Еще заимствований: 1,11%