Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Методы защиты информации»

**ОТЧЕТ**

к лабораторной работе

на тему:

**«Стеганографические методы»**

|  |  |
| --- | --- |
|  | Выполнил:  Слуцкий Никита Сергеевич,  студент группы 053505 |
|  | Проверил: Лещенко Евгений Александрович, ассистент каф. Информатики |

Минск 2023

СОДЕРЖАНИЕ

[Введение 3](#_Toc150253657)

[1 Краткие теоретические сведения 4](#_Toc150253658)

[2 Ход выполнения работы 5](#_Toc150253659)

[Заключение 6](#_Toc150253660)

[Приложение А 7](#_Toc150253661)

# Введение

Стеганография – это область информационной безопасности, которая изучает методы и техники скрытой передачи информации. Она предоставляет возможность передачи сообщений таким образом, чтобы эта передача оставалась незаметной для сторонних наблюдателей. Стеганографические методы активно применяются для скрытой передачи информации в различных областях, включая кибербезопасность, цифровое искусство, аудио и видео коммуникации и многое другое.

Целью данной лабораторной работы ставится реализовать программное средство, сокрытия (извлечения) текстового сообщения в (из) JPEG изображение(я) на основе метода сокрытия в частотной области изображения методом дискретного косинусного преобразования (ДКП). А также оформить отчёт о проделанной работе в соответствии с положениями стандарта предприятия БГУИР 2017.

# 1 Краткие теоретические сведения

Стеганография – это способ спрятать информацию внутри другой информации или физического объекта так, чтобы ее нельзя было обнаружить. С помощью стеганографии можно спрятать практически любой цифровой контент, включая тексты, изображения, аудио- и видеофайлы. А когда эта спрятанная информация поступает к адресату, ее извлекают.

Стеганографию иногда сравнивают с криптографией, поскольку и то и другое – это способы секретной коммуникации. Однако между ними есть разница, поскольку при стеганографии данные при отправке не шифруются, а при получении не нужен ключ для расшифровки.

Стеганография позволяет скрывать информацию так, чтобы не вызвать подозрений. Одна из самых распространенных техник стеганографии – LSB (least significant bit – «наименее значащий бит»): секретные данные встраиваются в наименее значащие биты медиафайла:

1 Каждый пиксель в изображении состоит из трех байтов данных, соответствующих красному, зеленому и синему цветам. Некоторые форматы изображения содержат дополнительный, четвертый байт для прозрачности, так называемый «альфа-канал».

2 LSB-стеганография позволяет изменять последний бит каждого из этих байтов так, чтобы спрятать один бит информации. Таким образом, чтобы спрятать 1 Мб данных с помощью этой техники, понадобится графический файл объемом 8 Мбайт.

3 Изменение последних битов пикселя не влияет на зрительное восприятие картинки, так что при сравнении оригинального и измененного файлов разница незаметна.

Метод ДКП (дискретное косинусное преобразование) – метод сокрытия данных, который заключается в изменении величин коэффициентов ДКП. При этом ДКП применялось ко всему изображению в целом.

LSB, о котором сказано выше, неустойчив к различным искажениям, в том числе и сжатию. В таких случаях применяются более устойчивые методы, засекречивающие информацию в частотной области, а не во временной, как в методе LSB. Преимуществом преобразований в частотной области является их применение не только ко всему изображению, но и каким-то определенным его зонам.

В данном способе изображение разделяется на блоки 8x8 пикселей, каждый из которых используется для шифрования одного бита сообщения. Шифрование начинается с произвольного подбора блока. Для осуществления секретного канала абоненты должны выбрать два определенных коэффициента ДКП, которые будут применяться для шифрования секретной информации. Данные коэффициенты – косинус-функции, соответствующие средним частотам. Такое соответствие позволит сохранить данные в необходимых областях сигнала при JPEG-сжатии, так как эти области не будут удаляться.

# 2 Ход выполнения работы

В рамках лабораторной работы был создан проект на языке программирования TypeScript. Для работы с разными изображениями использована среда браузера и библиотека React.

Таким образом программный продукт представляет собой простое клиентское веб-приложение, разработанное в экосистеме NodeJS, исполняющееся в браузере.

С помощью средства можно выбрать картинку, ввести сообщение и сохранить изображение. Также можно раскодировать и проверить сообщение.

# Заключение

В заключение, стеганографические методы предоставляют возможность скрытой передачи информации, что является важным инструментом в сфере информационной безопасности и коммуникации. Отчет рассмотрел различные методы стеганографии и их применение в различных областях.

Применение стеганографических методов имеет свои преимущества и вызовы. С одной стороны, они позволяют обеспечить конфиденциальность и защиту данных, а также внедрить скрытые коммуникации. С другой стороны, существует потенциальная угроза злоупотребления и нарушения конфиденциальности.

Успешное использование стеганографических методов требует баланса между безопасностью, эффективностью и этичностью. Важно тщательно взвесить требования и потенциальные риски в каждом конкретном сценарии применения стеганографии.

В рамках лабораторной работы было реализовано программное средство для сокрытия и извлечения текста из изображений на языке программирования TypeScript, составлен отчёт. Цели, поставленные во введении, можно считать достигнутыми.

# Приложение А

(Листинг кода)

function decodeMessage(colors: any) {

let history: any = [];

let messageSize = getNumberFromBits(colors, history);

if ((messageSize + 1) \* 16 > colors.length \* 0.75) {

return '';

}

if (messageSize === 0) {

return '';

}

let message: any[] = [];

for (let i = 0; i < messageSize; i++) {

let code = getNumberFromBits(colors, history);

message.push(String.fromCharCode(code));

}

return message.join('');

}

function getBit(number: any, location: any) {

return ((number >> location) & 1);

};

function setBit(number: any, location: any, bit: any) {

return (number & ~(1 << location)) | (bit << location);

};

function getBitsFromNumber(number: any) {

let bits: any[] = [];

for (let i = 0; i < 16; i++) {

bits.push(getBit(number, i));

}

return bits;

};

function getNumberFromBits(bytes: any, history: any) {

let number = 0,

pos = 0;

while (pos < 16) {

let loc = getNextLocation(history, bytes.length);

let bit = getBit(bytes[loc], 0);

number = setBit(number, pos, bit);

pos++;

}

return number;

};

function getMessageBits(message: any) {

let messageBits: any = [];

for (let i = 0; i < message.length; i++) {

let code = message.charCodeAt(i);

messageBits = messageBits.concat(getBitsFromNumber(code) as any);

}

return messageBits;

};

function getNextLocation(history: any, total: any) {

let loc = 0;

while (true) {

if (history.indexOf(loc) >= 0) {

loc++;

} else if ((loc + 1) % 4 === 0) {

loc++;

} else {

history.push(loc);

return loc;

}

}

};