МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Стеганография в графических файлах

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Латанова Кирилла Вячеславовича

Научный руководитель
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Саратов 2024

ВВЕДЕНИЕ

Современное общество не может существовать без регулярного обмена информацией. Чаще всего люди делятся ею открыто, но в некоторых случаях возникает необходимость передать информацию так, чтобы потенциальный злоумышленник не смог распознать ее наполнение. Иногда требуемый уровень секретности вынуждает нас делиться данными так, что посторонние даже не осознают, что какая-то информация была передана. В этом случае целесообразно прибегнуть к использованию стеганографии. При ее применении информация передается в нетипичном виде – например, текст зашифрован в графическом файле, в аудиодорожки внедряются водяные знаки и так далее.

Целью данной работы является изучение цифровой стеганографии, детальное рассмотрение одного из ее методов – метода замены самого незначащего бита, – изучение структуры PNG-файла, рассмотрение дополнительных методов криптографии и программная реализация Telegram-бота на языке Python, позволяющего организовать внедрение текста в изображение.

Дипломная работа состоит из введения, 4 разделов, заключения, списка использованных источников и 8 приложений. Общий объем работы – 50 страниц, из них 23 страниц – основное содержание, включая 11 рисунков и 0 таблиц, список использованных источников из 14 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Стеганография

В этом разделе вводится понятие стеганографии, рассматривается общепринятая терминология и примеры применения стеганографических методов. Также вводится понятие стеганографии в графических файлах и методика, используемая для сокрытия информации в изображениях.

1.1 История стеганографии

В данном разделе рассматривается определение стеганографии, приводятся исторические примеры ее применения.

Стеганография – это наука о скрытой передаче информации путем сохранения в тайне самого факта передачи. В отличие от криптографии, в которой скрывается само содержимое отправляемого сообщения, используя стеганографию, можно скрывать сам факт его существования.

1.2 Цифровая стеганография

Рассматривается понятие цифровой стеганографии, причины возможности стеганографии в оцифрованных файлах. Изучены принципы, на которых базируется цифровая стеганография, вводится основная терминология и правила, которым нужно следовать при шифровке информации. Рассматриваются понятия сообщения и контейнера, также изучается сфера применения стеганографических технологий.

Стеганографическая система (далее – стегосистема) – это средства и методы, использующиеся для образования тайного способа пересылки информации.

Сообщение – это информация, которую необходимо тайно передать с помощью стеганографических методов. Сообщением может служить информация любого типа: аудиодорожка, видеоряд, изображение, текст и так далее.

Контейнер – это файл, в который возможно спрятать сообщение. При выборе контейнера важно знать, что он может сильно повлиять на

возможность идентификации спрятанной в нём информации, ее извлечения и расшифровки.

1.3 Стеганография в графических файлах

Рассматривается структура графических файлов, из которой вытекает возможность сокрытия в них секретных сообщений. Далее в данном разделе описывается структура распространенных графических форматов, причины, по которым возможно применение стеганографии в них, а также конкретизируются сферы ее применения.

1.4 Методы сокрытия информации в графических файлах

В данном подразделе приводятся примеры методов стеганографии в изображениях, а именно:

1. Метод замены наименее значащего бита, который заключается в изменении структуры байт картинки-контейнера с целью помещения в них (а именно в крайние правые биты) бит сообщения. Уточняется, что этот метод применим только к изображениям, которые используют сжатие без потерь.
2. Метод использования разности значений пикселей, основанный на работе с пикселями, яркость которых незначительно отличается друг от друга.
3. Метод изменения уровня серого, применяемый исключительно к черно-белым изображениям.
4. Метод дискретного косинусного преобразования, являющийся модификацией метода наименее значащего бита, применяющегося в данном методе к коэффициентам DCT-преобразования.

Также более подробно описывается метод замены наименее значащего бита, измененный и адаптированный под цели и задачи, поставленные в данной работе. Описан прямой и обратный ход разработанного алгоритма.

2 Формат PNG и его структура

В данном разделе изучается формат изображений PNG, история его возникновения, цели его разработки и области его применения. Также описывается структура данного формата, а именно описывается наличие альфа-канала, поддерживающего прозрачность.

В работе описаны следующие особенности формата:

1) Поддержка прозрачности;
2) Сжатие без потерь;
3) Поддержка различных глубин цвета;
4) Поддержка метаданных;
5) Поддержка анимации.

В целом, формат PNG является универсальным и гибким форматом изображений, который подходит для различных целей, включая веб-графику, дизайн, иконки и многое другое. Он сочетает в себе высокое качество изображений, поддержку прозрачности и возможность сжатия без потерь, что делает его предпочтительным выбором для многих профессионалов и любителей компьютерной графики.

3 Дополнительные меры защиты информации

В данном разделе рассматривается понятие криптографии и необходимость ее дополнительного использования при защите информации. Также изучаются некоторые из популярных алгоритмов шифрования.

3.1 Шифр Виженера

Шифр Виженера – это метод шифрования алфавитного текста с использованием серии различных шифров Цезаря, основанных на буквах ключевого слова. Алгоритм заключается в следующем: задан алфавит, считывается сообщение и ключ. Затем последовательно добавляем ключ в конец самого себя до тех пор, пока его длина не будет равна длине исходного сообщения. Если длина сообщения не кратна длине ключа, то прерываем процедуру как только закончится текст. После этого будем последовательно складывать порядковые номера в алфавите у символов исходного текста и полученного ключа и записывать полученный результат по модулю 26 (размерности латинского алфавита). Расшифровка, очевидно, будет проводиться в обратном порядке.

3.2 Алгоритм шифрования «Кузнечик»

В данном подразделе рассмотрен алгоритм шифрования Кузнечик, представляющий собой симметричный блочный шифр с длиной блока равной 128 бит и длиной ключа равной 256 бит.

Процесс шифрования состоит из нескольких раундов, каждый из которых включает в себя несколько преобразований: сложение по модулю 2 с раундовым ключом, замена с помощью блоков подстановок и линейное преобразование.

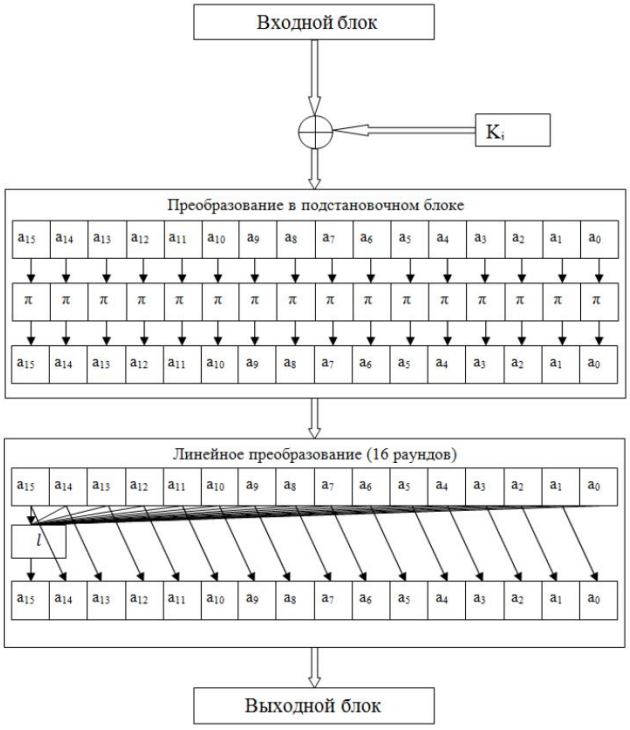

Рисунок 1 – Раунд шифрования алгоритма «Кузнечик»

Один раунд шифрования можно представить так, как показано на рис. 2. Алгоритм шифрования Кузнечик содержит в себе девять аналогичных раундов шифрования. Для каждого раунда из заданного 256-битного мастер-ключа вырабатывается соответствующий раундовый 128-битный ключ с помощью сети Фейстеля.

На вход сети Фейстеля сначала подаются половинки мастер-ключа, а затем выработанные ключи. В качестве левой части подается ключ с индексом

2i, а в качестве правой части ключ с индексом i. Правая часть проходит функцию F (рис. 2) и складывается по модулю 2 с левой частью, затем половинки меняются местами. Подобные преобразования повторяются 8 раз, и мы получаем новую пару ключей. Так вырабатываются ключи с 3-его по 10-ый.

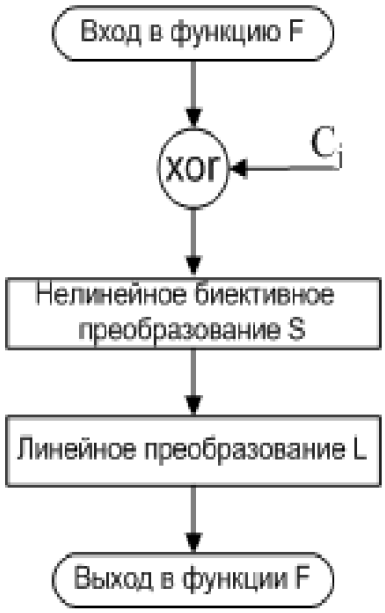

Рисунок 2 – Функция F, используемая в сети Фейстеля

Следующим этапом осуществляется преобразование при помощи блока подстановок S. 128-битный блок данных поступает на вход преобразования S, и разбивается на шестнадцать байтов. Далее этот блок подается на вход линейного преобразования L, в котором блок разбивается на 16 байт. После операции умножения этих байт выполняется операция сложения всех 16-ти элементов между собой. Подобные расчеты проводятся 16 раз, в результате получаем новое 16-ти байтное значение. Так выполняется один раунд зашифрования. Остальные 8 раундов выполняются абсолютно аналогично. После сложения по модулю два с последним, десятым, раундовым ключом мы получаем искомый зашифрованный текст.

Операция расшифрования выполняется по такому же принципу, но в обратном порядке и с использованием раундовых преобразований, инверсных к тем, что использовались при зашифровании.

3.3 DES-алгоритм

DES – это симметричный блочный шифр, который использует один и тот же ключ для шифрования и расшифрования данных. В DES происходит разделение данных на блоки. Для шифрования данных DES разделяет их на блоки фиксированного размера, который составляет 64 бита.

DES использует 56-битный ключ для шифрования и расшифрования данных. Ключ должен быть предварительно сгенерирован и известен только отправителю и получателю. Ключ используется для генерации раундовых ключей, которые используются в каждом раунде шифрования.

Процесс шифрования DES состоит из 16 раундов. Каждый раунд включает в себя несколько шагов:

- Начальная перестановка;
- Раундовая функция, включающая в себя следующие шаги:
    1) Расширение;
    2) Ключевое смешивание
    3) Замена;
    4) Перестановка;
- Применение раундового ключа;
- Функция сети Фейстеля;
- Обратная перестановка.

Процесс расшифрования DES аналогичен процессу шифрования, но с использованием обратных операций.

DES обладает высокой криптографической стойкостью, что означает, что он представляет собой надежную защиту от несанкционированного доступа к

данным. Алгоритм использует сложные математические операции, которые делают его очень трудным для взлома без знания правильного ключа.

В целом, стандарт DES обладает некоторыми ключевыми свойствами, которые делают его надежным и эффективным алгоритмом шифрования. Вот ряд его преимуществ:

1. Криптографическая надежность;

2. Эффективность;

3. Широкое применение.

4 Техническая реализация

Результатом раздела 4 является разработанный Telegram-бот, написанный на языке Python и осуществляющий цифровую стеганографию методом замены LSB в файлах формата PNG. Также в программе создана возможность дополнительной защиты информации с помощью одного из описанных выше шифров на выбор. Данный способ взаимодействия с пользователем выбран по причине растущей популярности мессенджера Telegram, доступности интерфейса пользователя и удобства инструментов разработчика.

В присланное пользователем изображение шифруется служебная информация о сообщении и само сообщение длиной не более 4096 символов, отправляется сообщение об ошибке при попытке отослать слишком длинное сообщение.

Также в разделе 4 произведен анализ разработанного алгоритма: отправлялось сообщение допустимой длины, слишком длинное сообщение, проводилось тестирование на цветных, черно-белых изображениях и на картинках с прозрачными областями.

ЗАКЛЮЧЕНИЕ

В ходе написания данной работы было изучено понятие стеганографии и ее история. Были приведены примеры использования каждого вида. Кроме того, были описаны основные понятия, связанные с каждым видом стеганографии. Более подробно рассмотрены методы цифровой стеганографии в графических файлах, изучен и применен метод LSB и разобрана структура формата PNG. Также применены и реализованы некоторые криптографические алгоритмы для дополнительной защиты передаваемой текстовой информации.

В практической части работы на языке Python разработан Telegram-бот для осуществления внедрения текста в графический контейнер и извлечения из него зашифрованной информации. Бот поддерживает выбор криптографического алгоритма для шифрования поступившего от пользователя текста и ключа к нему и приём контейнера для кодирования и декодирования информации.

Разработанный бот может использоваться для создания секретных контейнеров и отправки его другим пользователям Telegram с возможностью извлечения из него информации путем взаимодействия с тем же мессенджером.