Министерство образования Российской Федерации

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н. Э. БАУМАНА

Факультет: Информатика и системы управления
Кафедра: Информационная безопасность

Курсовой проект на тему:

Обмен сообщениями с использованием алгоритма Куттера-Джордана-Боссена

Преподаватель:
Бородин А. А.

Студент:
Горбунов П. А.

Группа:
ИУ8-31

Москва
2017

Содержание

Цель

Главной целью курсового проекта является предоставление бота, реализуемого в одном из наиболее используемых приложений для общения – социальной сети Telegram – для обеспечения конфиденциальности информации в процессе передачи информации от одного пользователя к другому.

Основные определения

Telegram — бесплатный кроссплатформенный мессенджер для смартфонов и других устройств, позволяющий обмениваться текстовыми сообщениями и медиафайлами различных форматов.

Бот — специальная программа, выполняющая автоматически и/или по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей.

Telegram-bot-api — http-интерфейс для работы с ботом в Telegram.

Стеганография - способ передачи или хранения информации с учетом сохранения в тайне самого факта такой передачи (хранения).

Шифрование - обратимое преобразование информации в целях скрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней.

Контейнер — так называется любая информация, используемая для сокрытия тайного сообщения.

Алгоритм Куттера-Джордана-Боссена — алгоритм встраивания информации в контейнер.

PaaS-платформа (place as a service) - модель предоставления облачных вычислений, при которой потребитель получает доступ к

использованию информационно-технологических платформ: операционных систем, систем управления базами данных, связующему программному обеспечению, средствам разработки и тестирования, размещенным у облачного провайдера.

SQL - язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных. Понятие “реляционный” связано с словами “отношение”, “зависимость”.

Растровое изображение - изображение, представляющее собой сетку пикселей.

Unit-тесты - процесс в программировании, который позволяет проверить исходный код на корректность, а также дает возможность разработчику удостовериться в том, что внесенные изменения не привели к появлению ошибок в уже протестированных местах программы.

Введение

XXI век – век информационных технологий, в котором ежеминутно обрабатывается колоссальный поток постоянно изменяющейся и обновляющейся информации. В связи с этим на сегодняшний день одним из наиболее востребованных направлений деятельности по работе с информацией является информационная безопасность. Так, возрастает актуальность вопроса обеспечения безопасности данных при их передаче, обработке и хранении с помощью компьютерных технологий.

Особое место среди разнообразных средств защиты информации занимает стеганография. Стеганография — наука о скрытой передаче информации путем сохранения в тайне самого факта передачи информации. В настоящее время под стеганографией чаще всего понимают скрытие

информации в текстовых, графических, аудио и видео файлах путем использования специального программного обеспечения.

Данный проект, использующий методы стеганографии, предоставит обычному пользователю возможность передать свою конфиденциальную информацию другому пользователю без явного отображения контекста.

Требования к проекту

- автономная работа на отдельном сервере
- простой интерфейс
- минимальные потери информации при декодировании
- наличие документации

Проектирование системы

Элементы и функционал проекта были объединены в системы. Так, структура проекта представляет собой систему “Телеграмм-бот”, которая взаимодействует с системой “Шифрование текста".

“Телеграмм-бот” включает в себя реализованный на языке программирования “Python” бот с простым интерфейсом, понятным для пользователя принципом работы и действиями для взаимодействия. “Шифрование текста” подразумевает подсистемы, включающие в себя работу с контейнером, а точнее png-файлом, и скрипт, написанный на языке программирования и реализующий алгоритм Куттера-Джордана-Боссена.

Шифрование текста

Данная система состоит из двух классов: Container и Picture.

class Picture - класс, описывающий работу с png-файлом. В классе описаны методы:

- SaveLen - сохраняет информацию о длине последовательности нулей и единиц в контейнере;
- GetLen - достает информацию о длине спрятанной битовой последовательности в контейнере;
- ChangePix - “внедряет” переданную в метод битовую последовательность в контейнер. Реализует внедрение информации по алгоритму Куттера-Джордана-Боссена;
- GetBix - достает из контейнера биты спрятанной битовой последовательности;
- SaveChange - сохраняет все изменения в png-файл.

class Container - класс, предоставляющий простой интерфейс системе “телеграмм-бот” для внедрения/извлечения информации в/из контейнера. В полях данных класса присутствует объект класса Picture. Методы, используемые в данном классе:

- SetModule - задает путь до файла, в который будет производиться внедрение сообщения;
- Encrypt - передача битовой строки в метод ChangePix класса Picture для внедрения в png-файл;
- Decrypt - получение битовой цепочки из метода GetBix класса Picture для последующего декодирования и отправления телеграмму-боту;
- ParseStr - парсер строки байтов, полученных из контейнера. Будет выполняться метод в случае, если функция языка программирования decode() выдаст ошибку.

Телеграмм-бот

Данная система описывает реализацию телеграмм-бота на языке программирования “Python”. class SteganyBot описывает телеграмм-бота, его

методы взаимодействия с пользователем, методы получения/отправления png-файла и методы, связанные с системой “реализация алгоритма”. Отдельным классом предоставлен интерфейс для работы с базой данных - class DataBase. Основными методами класса SteganyBot являются:

- Start - метод бота “/start” - приветствие пользователя и ознакомление с базовыми методами;
- Play - метод бота “/play” - запрос о последующем действии: зашифровать, расшифровать, загрузить картинку, отправить картинку;
- Encrypt, Decrypt, Upload - обработка методов “/encrypt”, “/decrypt”, “/upload” соответственно;
- SendPicture - отправление png-файла пользователю;
- EncryptMessage - получение от пользователя сообщения и внедрение его в png-файл путем создания объекта класса Container;
- DecryptMessage - расшифрование сообщения из переданного пользователем контейнера через взаимодействие с классом Container;
- GetText и GetDoc - обработка сообщений/документов от пользователя и перенаправление на дальнейшие методы.

Класс DataBase - собрал воедино методы, взаимодействующие с подключенной базой данных. В базе данных хранится информация о каждом пользователе:

- username;
- имя;
- фамилия;
- идентификационный номер чата.

Основными методами являются такие методы, как:

- AddUser - добавление информации о пользователе в базу данных;

- CheckUser - проверка на наличие информации о пользователе;
- GetId - возвращение идентификационного номера чата.

На уровне класса SteganyBot описывается некая система флагов для регулирования вызова методов или направление потока строк в методы. Такая структура называется UsersFlags - массив словарей, где ключ является идентификатором чата, а значения являются словарем флагов. Словари используются для простого взаимодействия с флагами.

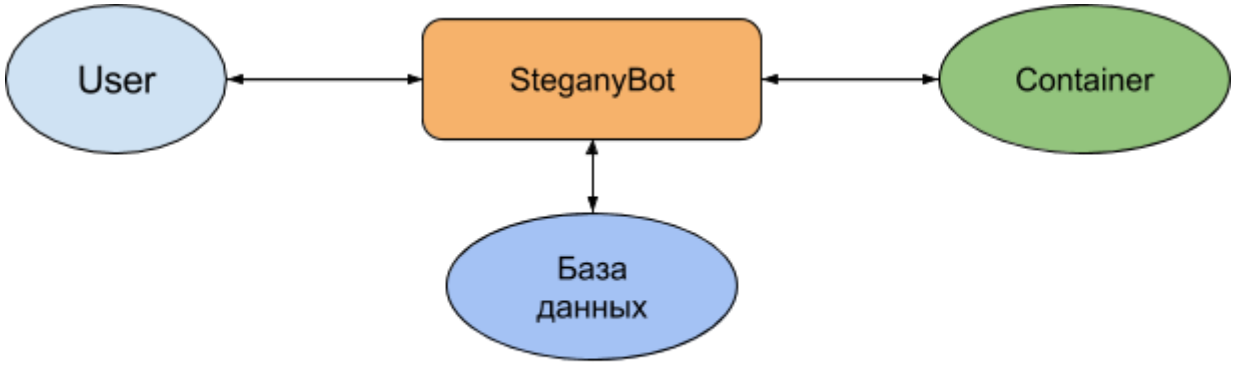


Рисунок 1. Взаимодействие подсистем в телеграмм-боте

“Телеграмм-бот” использует набор сторонних программ и инструментов, которые подключаются для определенной задачи, например, автоматизации процесса сборки или выполнения тестов. В курсовом проекте используются следующие инструменты:

- Github - хостинг, на котором содержатся исходные файлы проекта. Предоставляет возможность подключения других программ/инструментов;
- TravisCI - веб-сервис для автоматизации проверки исходников, находящихся в репозитории на Github’e;

- Heroku - облачная PaaS-платформа, предоставляющая возможность использования проекта как web-приложение.

Алгоритм Куттера-Джордана-Боссена

Для встраивания информации в контейнер используется одно из свойств зрительной системы человека. Данное свойство выражается в том, что человек воспринимает изменения яркости синего цвета намного хуже, чем изменения яркости красного и зелёного цвета.

Именно по этой причине для встраивания информации будет использоваться синий цвет заданного контейнера-изображения.

Изображение будет рассматриваться в цветовой модели RGB.

Обозначения

Для дальнейшего анализа будут необходимы следующие переменные:

$B_{x,y}$ - яркость синего цвета пикселя с координатами (x,y);

$B^*_{x,y}$ - измененная яркость синего цвета пикселя;

$Y_{x,y}$ - яркость пикселя;

m_i - i-ый бит сообщения, которое мы хотим встроить;

λ - коэффициент, задающий энергию встраиваемого бита данных (задаётся исходя из функционального назначения и особенности стеганосистемы);

σ - размер области, по которой будет прогнозироваться яркость.

Встраивание

Встраивание информации будет производиться следующим образом: 1 бит сообщения будет внедряться в 1 пиксель контейнера. Секретный ключ задаёт координаты пикселей, в которые будет производиться встраивание.

При встраивании яркости красного и зелёного цветов остаются без изменений, а яркость синего – изменяется по следующей формуле:

$$B^*_{x,y} = B_{x,y} + \lambda Y_{x,y} \text{ при } m_i = 1$$

или

$$B^*_{x,y} = B_{x,y} + \lambda Y_{x,y}, \text{ при } m_i = 0$$

где $\lambda = 1.3$,

$$Y_{x,y} = 0.3 * R_{x,y} + 0.59 * G_{x,y} + 0.11 * B_{x,y}$$

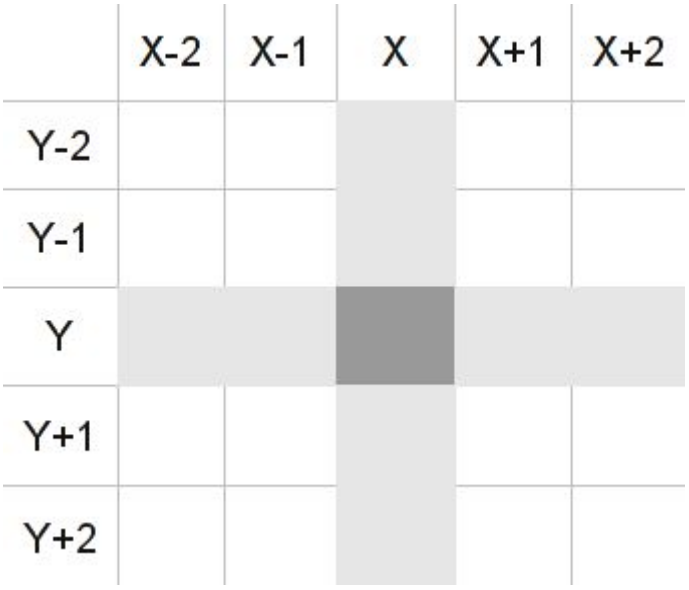
Извлечение

В связи с тем, что принимающая сторона не обладает оригинальным изображением, пересылаемым от пользователя, однозначно понять увеличилась или уменьшилась яркость синего цвета не представляется возможным. Поэтому по формуле производится подсчет значений яркости синего цвета в ячейках для извлечения встроенной информации:

$$\overline{B_{x,y}} = \frac{\sum_{i=1}^{\sigma}(B_{x,y+i}+B_{x,y-i}+B_{x+i,y}+B_{x-i,y})}{4\sigma},$$

где $\sigma = 1/3$.

Проиллюстрируем на примере ($\sigma = 2$):



Пиксель в центре – это пиксель, чью яркость синего цвета необходимо спрогнозировать, опираясь на пиксели, обозначенные светло-серым цветом. И наконец, для извлечения скрытого сообщения необходимо использовать формулу:

$$m_i = 1 \text{ при } B^*_{x,y} > \overline{B_{x,y}} \text{ или } m_i = 0 \text{ при } B^*_{x,y} < \overline{B_{x,y}}$$

Плюсы и минусы

Рассмотренный стеганографический алгоритм внедрения информации в контейнер обладает следующими преимуществами:

- Высокая пропускная способность;
- Высокая устойчивость к несанкционированному ознакомлению;
- Высокая устойчивость к частотному детектированию;
- Высокая устойчивость к разрушению младших бит контейнера;
- Устойчивость к атаке сжатия.

Однако при его использовании выявляются некоторые недостатки:

- Извлечение носит вероятностный характер.

Для уменьшения вероятности ошибки используют помехоустойчивое кодирование.

Выбор технологий

Выбор языка программирования

Для курсового проекта при выборе языка программирования было произведено сравнение между двумя языками программирования: “Python” и “C++”.

“Python” - высокоуровневый интерпретируемый язык программирования, преимущества которого заключаются в следующих характеристиках:

- динамическая типизация
- автоматическое управление памятью
- механизм обработки исключений
- простой синтаксис кода
- большое количество сторонних библиотек

Отрицательной чертой данного языка программирования является его производительность.

“C++” - компилируемый, сильно типизированный язык программирования, плюсы которого проявляются в нижеперечисленных характеристиках:

- высокая производительность
- возможность работы на низком уровне с памятью, адресами, портами.
- кросс-платформенность

Из минусов стоит отметить:

- строгую типизацию
- сложность привязывания библиотек.

Для данного курсового проекта необходимо, чтобы для языка программирования было предоставлено Telegram API для написания телеграмм-бота и его функционала. Помимо Telegram API, должна присутствовать библиотека для работы с базой данных. А также следует отметить, что к проекту нет требований к производительности. Именно поэтому для данного курсового проекта был выбран язык программирования – «Python».

Список используемых библиотек

Далее приведен список используемых библиотек, необходимых для реализации телеграмм-бота:

- PostgreSQL - объектно-реляционная база данных;
- Psycopg2 - библиотека для «Python», предоставляющая интерфейс для работы с базой данных PostgreSQL, выполнения SQL-запросов к базе данных;
- Python-telegram-bot - библиотека для «Python» предоставляемая API для работы с Telegramm. С помощью этой библиотеки реализован телеграмм-бот. У библиотеки имеется подробная документация.
- PIL - (Python Image Library) - библиотека для языка «Python» необходимая для работы с растровой графикой

Описание технических решений

В ходе разработки проекта возникли некоторые проблемы:

Оценочный характер извлечения сообщения из контейнера по алгоритму Куттера-Джордана-Боссена приводит к возможному появлению ошибочного определения бита: вместо истинного значения «0» может быть установлено значение «1» и наоборот.

В приведенном случае необходимо модифицировать алгоритм с целью уменьшения этой вероятности следующими возможными способами:

1. Необходимо дополнить процедуру извлечения следующими условиями:

$$if\ (\delta = 0\ and\ B^{*}_{(x,\,y)} = 0)\ then\ \delta = -0.5$$

$$if\ (\delta = 0\ and\ B^{*}_{(x,\,y)} = 255)\ then\ \delta = 0.5$$

2. Необходимо дополнить процедуру встраивания следующим правилом:

$$if\ \lambda = 0\ then\ \lambda = \frac{\alpha}{\upsilon}$$

3. Для уменьшения вероятности ошибки извлечения при реализации алгоритма Куттера-Джордана-Боссена необходимо встраивать каждый бит информации несколько раз.

При декодировании битовой последовательности методом над строкой decode() библиотеки языка «Python» существует вероятность получения ошибки «UnicodeError». В большинстве случаев данная ошибка связана с декодированием в русские символы.

Суть ошибки заключается в том, что при извлечении бита из контейнера может возникать его ошибочное определение. Данный сбой влечет за собой установление совсем другого байта. Это может стать причиной дальнейшего неверного определения пары unicode-байтов. В

кодировке Unicode представлением символа в «utf-8» является пара из двух байт в hex-формате (в формате 0xd0 - шестнадцатеричное число). Ошибка влечет за собой возможность изменения формата первого байта и дальнейшей неполадке «UnicodeError» метода decode().

Если методом decode() не получается декодировать последовательность байт, полученную после извлечения информации из контейнера, необходимо вызвать функцию ParseStr(text), которая сможет декодировать информацию посимвольно. Данная функция сначала формирует байтовую пару, затем проверяет формат первого байта, который должен совпадать для русских символов с форматом байта 0xd0 или 0xd1. Если выявляется несовпадение, функция автоматически добавляет требуемый байт и впоследствии декодирует его. Таким образом происходит просмотр всех байтов, после чего получается текстовый формат информации, которая изначально была зашифрована. Однако, с точки зрения русского языка, полученный текст может оказаться всего лишь набором символов.

При одновременной использовании телеграмм-бота несколькими людьми команда, вызванная одним пользователем может принять сообщение от другого пользователя. Так, например, результат шифрования, вызванный одним человеком, может отправиться другому человеку.

В сложившейся ситуации необходимо создать структуру, которая будет хранить флаги для корректной работы команд. Это важно для того, чтобы команды не пересекались между собой. Такая структура флагов будет связана с идентификатором чата.

Заключение

В процессе выполнения курсового проекта был реализован Телеграмм-бот, внедряющий сообщение пользователя в png-контейнер с помощью стеганографического алгоритма Куттера-Джордана-Боссена. Телеграмм-бот был успешно запущен на сервисе Heroku. Работа проектного кода была успешно протестирована unit-тестами.

Список используемой литературы

1. Защелкин К. В., Иващенко А. И., Иванова Е. Н. Усовершенствование метода стеганографического скрытия данных Куттера-Джордана-Боссена / Радіоелектронні і комп'ютерні системи. – 2013. - № 5. - С. 151–155. (Дата обращения: 12.12.2017)
2. Lundh Fredrik, Ellis Matthew. Python Imaging Library Overview. – 2002 [Электронный ресурс] Режим доступа: https://habrahabr.ru/sandbox/59849/, свободный (Дата обращения: 12.12.2017)
3. Python Telegram Bot's documentation [Электронный ресурс] Режим доступа: https://python-telegram-bot.readthedocs.io/en/stable/, свободный (Дата обращения: 12.12.2017)