Scientific and Practical Cyber Security Journal (SPCSJ) 4(3): 48 - 52 ISSN 2587-4667
Scientific Cyber Security Association (SCSA)

СТЕГАНОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ В ФАЙЛАХ ФОРМАТА OFFICE OPEN XML С ПОМОЩЬЮ ЦВЕТОВОЙ МОДЕЛИ RGB

STEGANOGRAPHIC INFORMATION PROTECTION IN OFFICE OPEN XML FORMAT FILES USING RGB COLOR MODEL

Babenko Yuriy, PhD student, Taras Shevchenko National University of Kyiv, Kiev, Ukraine.
Babenko Mykhailo, PhD in Engineering Science, Associate Professor, Dniprovsk State Technical University, Kamyanske, Ukraine.

АННОТАЦИЯ. В данной статье рассмотрены алгоритмические особенности реализации методов текстовой стеганографии для скрытой передачи данных. Представлен алгоритм, основанный на модификации цветовых параметров символов текста, а именно применении метода наименьшего значащего бита к цветовым RGB каналам текстовых символов файла формата Microsoft Office Word с расширением DOCX. Разработано программное обеспечение, реализующее данный алгоритм, с возможностью вложения / извлечения скрытой информации.

КЛЮЧЕВЫЕ СЛОВА: стеганография, формат OFFICE OPEN XML, метод наименьшего значащего бита, цветовая модель RGB

ABSTRACT. This article describes the algorithmic features of the implementation of text steganography methods for hidden data transmission. An algorithm based on a modification of the color parameters of the text characters is presented. The least significant bit method is applied to the RGB color channels of text characters in a Microsoft Office Word file format with the .docx extension. Software implementation of algorithm has been developed. Software can embed / extract hidden information.

KEYWORDS: steganography, OFFICE OPEN XML format, The least significant bit method, RGB color model

Введение. Рассматривая способы защиты информации, наряду с криптографическими методами уместно обратить внимание на стеганографические методы. Сам термин «стеганография» означает скрытое сообщение, которое полностью исключает возможность узнать о его существовании третьему лицу. И если, образно говоря, криптография делает понятное непонятным, то стеганография делает видимое невидимым (иногда и в прямом смысле слова). Достигается это «растворением» скрываемой информации среди других данных значительно большего объема.

Компьютерная стеганография основывается на двух основных принципах [1]. Во-первых, файлы с оцифрованными изображениями, а также аудио- и видеофайлы можно определенной мерой изменить без потери их функциональности. Во-вторых, возможности человека различать незначительные изменения звука или цвета довольно ограниченные. Стеганографические методы дают возможность заменить несущественные части данных нужной информацией. Это означает, что семейное фото может

содержать информацию коммерческого характера, а файл с любимой мелодией – секретное сообщение.

Чаще всего стеганография применяется для создания цифровых водяных знаков, которые, в отличие от обычных, можно проявить, лишь используя необходимое программное обеспечение. Цифровые водяные знаки записываются в виде псевдослучайных последовательностей сигналов шума, которые сгенерированы на базе секретных ключей. Такие знаки обеспечивают аутентичность или неприкосновенность документа, дают возможность идентифицировать владельца или автора, проверить права пользователя или дистрибьютора, даже в том случае, когда файл был обработан.

Существующие алгоритмы встраивания секретной информации разделяют на несколько групп:

1. Те, которые работают с самым цифровым сигналом.
2. «Впаивание» секретной информации. В этом случае происходит наложение изображения, звука или текста, которые необходимо спрятать, сверх оригинала. Довольно часто применяется для встраивания ЦВЗ (цифровой водяной знак).
3. Использование возможностей файловых форматов. Сюда относится вложение информации в метаданные или в другие зарезервированные поля файла, которые обычно не используются.

Контейнером могут служить любые данные (файлы) достаточно большого объема, например графические или звуковые. Их структура проста и, как правило, обладает большой избыточностью, позволяющей вместить значительный объем дополнительной информации. Однако текстовые файлы все же более распространены, и их структура широко известна.

Поскольку мы будем встраивать скрытые данные в текстовый файл, рассмотрим методы, с помощью которых это можно реализовать. Для скрытия конфиденциальных сообщений в тексте (так называемая текстовая стеганография – *text steganography*) используется или обычная избыточность письменной речи, или же форматы представления текста.

Наиболее сложным объектом для скрытия данных по многим причинам является электронная (файловая) версия текста. В отличие от текстового файла его "жесткая" копия (например, бумажная) может быть обработана как высокоструктурированное изображение и поэтому является относительно легко поддающейся разнообразным методам скрытия, таким как незначительные изменения формата текстовых шаблонов, регулирование расстояния между определенными парами символов (кернинг), расстояния между строками и т.п. В значительной степени такая ситуация вызвана относительным дефицитом в текстовом файле избыточной информации, особенно в сравнении с графическими или, например, звуковыми файлами. В то время как в большинстве случаев существует возможность внести незаметные глазу и неощутимые на слух модификации в изображение и звук, даже дополнительная буква или знак пунктуации в тексте могут быть легко распознаны случайным читателем.

Скрытие данных в тексте требует поиска таких модификаций, которые были бы незаметными подавляющему большинству читателей. Обычно рассматривают три группы методов, которые получили наибольшее распространение при встраивании скрываемых данных в текст [2]:

- методы произвольного интервала, которые осуществляют встраивание путем манипуляции с пробельными символами (свободным местом на печатной полосе);
- синтаксические методы;

- лексические методы;
- семантические методы, в основу алгоритмов которых положено манипулирование словами, зависимое от скрываемых бит данных.

Наибольшее распространение получили следующие методы текстовой стеганографии:

1. Метод изменения порядка следования маркеров конца строки CR/LF. Использует индифферентность подавляющего числа средств отображения текстовой информации к порядку следования символов перевода строки (CR) и возврата каретки (LF), ограничивающих строку текста. Традиционный порядок следования CR/LF соответствует 0, а инвертированный LF/CR означает 1.
2. Метод хвостовых пробелов. Предполагает дописывание в конце коротких строк (менее 225 символов; значение 225 выбрано достаточно произвольно) от 0 до 15 пробелов, кодирующих значение полубайта.
3. Метод знаков одинакового начертания. Предполагает подмену (бит 1) или отказ от такой подмены (бит 0) русского символа латинским того же начертания.
4. Изменение количества промежутков. Будем считать, что один промежуток отвечает биту «0», а два – «1». Программа получает любой текст в качестве контейнера и вкладывает в него сообщение, заменяя его биты на соответствующее количество промежутков. Важную роль здесь также играет и способ кодирования символов. Нужно получить код символов оптимальной длины, и чтобы при этом двойной промежуток встречался по возможности меньше раз.
5. Метод изменения межстрочного расстояния, или line-shift coding. В его стандартной реализации предлагается скрывать стегосообщение в изменении высоты межстрочных интервалов.
6. Word-shift coding. Изменяется расстояние между словами текста. Суть метода заключается в том, что берется текст с разными расстояниями между словами. Выделяется максимальное и минимальное расстояние, которые обозначаются соответственно 1 и 0, а другие расстояния увеличивают или уменьшают до размеров выделенных.
7. Feature coding. Внесение специфических изменений в очертания отдельных букв.

Рассмотренные выше методы довольно легко встраиваются в любой текст, независимо от его содержания, назначение и языка. Однако они имеют несколько существенных недостатков: обладают малой пропускной способностью и могут быть выявлены для электронного документа путем изменения параметров размера и начертания шрифта.

Поэтому мы будем использовать другой метод, который имеет название LSB (Least Significant Bit, наименьший значащий бит). Суть заключается в замене наименее значащих бит контейнера на биты сообщения, которое необходимо спрятать [3]. Младший значащий бит несет в себе меньше всего информации. Известно, что человек в большинстве случаев не способен заметить изменений в этом бите. Фактически, НЗБ – это шум, поэтому его можно использовать для встраивания информации путем замены менее значащих битов контейнера битами секретного сообщения. Поскольку возможности человеческого глаза различать оттенки одного и того самого цвета довольно ограниченные, такая замена будет незаметной для человека.

Именно на базе метода LSB и будет реализован алгоритм сокрытия тайной информации в цветовых RGB каналах текстовых символов файла формата Microsoft Office Word с расширением DOCX, которому посвящена эта работа.

Постановка задачи. Разработать программное обеспечение, с помощью которого можно будет спрятать тайную информацию таким образом, чтобы о ее существовании не узнал кто-нибудь другой. Также необходимо обеспечить возможность вытягивания секретного сообщения из контейнера, в котором оно уже скрыто. В качестве контейнера (или хранилища) для тайных данных мы будем использовать файл формата Microsoft Office Word с расширением DOCX. Почему именно DOCX, а не, например, DOC или TXT? На это есть несколько важных причин.

Во-первых, файл с расширением DOCX, в отличие от DOC, представляет собой zip-архив с XML-документами, который можно распаковать и получить всю необходимую информацию: текст, изображения, таблицы и т.п. Благодаря этому довольно легко вкладывать и получать скрытые в нем данные. Формат файла основан на Open XML, подробно описанный в стандарте ECMA-376: Office Open XML File Formats, и использует сжатие по алгоритму ZIP для уменьшения размера файла. Данный архив содержит два типа файлов – файлы формата XML с расширениями xml и rels и медиафайлы, например, изображения. Логически файл состоит из трех видов элементов: типов, частей и связей. Типы – это список сущностей, встречающихся в документе, например, типов медиафайлов или частей документов, части – это отдельные части документа, для каждой части документа создан отдельный файл формата XML. Между частями документа устанавливаются связи. Таким образом, можно сказать, что файл формата docx представляет собой набор сжатых файлов формата XML, причем все текстовое содержимое электронного документа Microsoft Word формата DOCX находится в одном XML файле, а именно в document.xml. Файл document.xml представляет собой XML файл в элементной форме, где каждому элементу обычно соответствует один атрибут.

Во-вторых, DOCX – наиболее популярный и массовый формат, и его частое использование не будет вызвать ни у кого сомнений на предмет вложенных в нем данных, что несомненно является большим плюсом его использования при стеганографической защите.

В-третьих, размер DOCX-файла значительно меньше, чем его аналога с расширением DOC. Особенно это заметно в файлах, которые содержат большое количество изображений или графиков.

Результаты работы. Как было уже сказано раньше, человеческий глаз не в состоянии отличить незначительные оттенки одного и того же цвета. Этим можно удачно воспользоваться при построении алгоритма вложения тайных данных в контейнер. Суть этого алгоритма заключается в следующем. У нас есть сообщения, которое необходимо спрятать в документ с расширением DOCX. При этом сам документ должен уже содержать в себе текстовую информацию. От объема этой информации будет зависеть объем тех данных, которые мы сможем в него вложить. Чем больше текста содержит документ, тем больше данных мы сможем в него спрятать. Тайное сообщение мы будем вкладывать в RGB каналы цвета каждого текстового символа из этого файла. Для этого нам сначала необходимо «разобрать» документ Word, получить из него все необходимые данные: текст и информацию о цветах каждого из символов в формате RGB. Потом полученные составляющие цвета

нужно перевести в двоичную систему счисления и заменить младшие биты составных цвета битами нашего сообщения. Более детально объясним это на примере:

Это 1 байт нашего сообщения:

10 101 010

Это RGB цвета одного символа:

R: 11110**00**

G: 00001**000**

B: 11001**000**

Заменив 2 младших бита в канале R и 3 младших биты в каналах G и B, получим следующий результат:

R: 11110**10**

G: 00001**101**

B: 11001**010**

Данная операция не внесет в цвет заметных человеческому глазу искажений. Вместе с тем она поможет нам вложить ровно 1 байт нашего сообщения в цвет каждого символа входного файла. Т.е. максимальное количество байт (или символов), которые мы можем спрятать, будет равно количеству символов документа с расширением DOCX, включая промежутки, табуляции, символы возврата каретки и абзаца.

Аналогичным образом выполняется и вытягивание данных из контейнера. Для того, чтобы получить сообщение, нужно, как и в первом случае, «разобрать» документ Word, получить цвета текстовых символов в формате RGB и прочитать необходимое количество последних бит каждого канала. Они и будут составлять один байт (или символ) скрытых данных. Проделав эти действия для всех других цветов, мы получим полностью текст секретного сообщения.

Авторами разработано программное обеспечение, которое реализует вышеописанный алгоритм. Также реализован собственный парсер docx-документов, который в отличие от уже существующих, полностью удовлетворяет требованиям задачи и включает в себя лишь необходимые функции, такие как считывание текста с сохранением форматирования, считывание цветов символов, которые используются в файле, и т.п.

Выводы. На базе метода LSB с использованием цветовой модели RGB построен алгоритм вложения скрытых данных в документ Microsoft Word с расширением DOCX.

Разработано программное обеспечение на языке C#, реализующее данный алгоритм, с возможностью вложения/извлечения скрытой информации. Программа функционирует в средах ОС семейства Windows. Для разработки программного обеспечения использовался пакет Microsoft Visual Studio.net 2017.

Результаты, полученные в этой работе, будут полезны в научно-технической сфере и сфере защиты данных.

Список использованной литературы

1. Грибунин В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. - М.: «Солон-Пресс», 2009. - 272 с.

2. А Конахович Г.Ф. Компьютерная стеганография. Теория и практика. / Г.Ф.Конахович, А.Ю.Пузыренко. - К.: "Мк-Пресс", 2006. - 288 с.

3. .Домарев В.А. Безопасность информационных технологий. Системный подход / В.В.Домарев . - К.: ООО "ТИД "ДС", 2004 .- 992 с.