УДК 003.26.7 004.9

Программное средство сокрытия данных в звуковых файлах

Маврина М.В., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
Кафедра «Информационная безопасность»*

*Научный руководитель: Алешин В.А., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru*

На сегодняшний день стеганография – одна из важнейших прикладных наук в информационной безопасности.

Стеганографию обычно используют совместно с методами криптографии, таким образом, дополняя её. Криптография защищает содержание сообщения, а стеганография защищает сам факт наличия каких-либо скрытых посланий.

В качестве носителя скрытой информации (стегоконтейнера) в данной работе используются звуковые файлы (.wav). При анализе существующей практики было выявлено, что без определенных технических средств (т.е. с помощью органов чувств) человек вряд ли обнаружит сам факт изменения исходного файла, что делает стегосистему в большей степени устойчивой к аудиовизуальному обнаружению, чем в случае сокрытия в других типах файлов (видео, изображениях, текстовых и т.д.).

Целью данной работы является программная реализация наиболее распространенного метода встраивания – метода наименьшего значащего бита – LSB (Least Significant Bit) и создание элемента стегосистемы (заполненного стегоконтейнера) с его использованием.

При проведении исследования использовались следующие методы: анализ; сравнение; расчет; оценка.

Назначение разработки: создание заполненных стегоконтейнеров и/или извлечение из них информации. Язык реализации: C#.

Как уже говорилось выше, метод LSB обладает низкой устойчивостью к различным стегоатакам. Однако устойчивость напрямую зависит от размеров сообщения и стегоконтейнера. Поэтому, в случае, рассматриваемом в данной работе – когда размер стегоконтейнера во много раз больше размера сообщения, автор считает его применение

достаточно эффективным, а сам метод – устойчивым к большинству методов стегоанализа.

Следует отметить, что разрабатываемое приложение не претендует на статус профессионального программного обеспечения, каким являются, например, программные продукты Max File Encryption [7], Fox Secret [4] и некоторые другие.

Для обеспечения достаточной эффективности автор предлагает использовать рассеянное (т.е. непоследовательное) встраивание бит скрываемой информации. Шаг встраивания рассчитывается как функция, зависящая от трех параметров – размера скрываемого сообщения (t) – в байтах, размера стегоконтейнера (s) – в байтах и глубины дискретизации (b) – в битах (1):

$$step = f(s, t, b) = \frac{s - c}{\frac{b}{8} * (t * 8)} = \frac{s - c}{b * t}, \quad (1)$$

где константа $c = 174$ байта является суммой: заголовок файла (44 байта) + 128 байт – кодирование размера сообщения (32 бита (в Int32) с шагом 4 байта) + 2 байта («технический» промежуток между кодированием размера сообщения и кодированием самого сообщения).

На значение шага встраивания наложено следующее ограничение: $step \geq 4$. Выбранное значение ограничителя объясняется тем, что большинство стегоатак проходят успешно при заполнении стегоконтейнера более чем на 35-40%. В случае наложения ограничения максимальное использование емкости стегоконтейнера составит 25%, что также позволяет повысить стойкость выбранного метода ко многим видам стегоанализа, в частности, к RS-анализу.

Разработанная программа представляет собой консольное приложение с интерфейсом на английском языке, предназначенное для создания стегоконтейнера или извлечения из него ранее сокрытой информации, при сокрытии которой использовалось данное приложение.

Работа программы сопровождается инструкциями с пояснениями, а также обработкой некорректно введенных данных.

Входными данными приложения являются указания по выполнению (определяются однозначно из запросов приложения) и полные пути к файлам – потенциальному стегоконтейнеру и скрываемым данным (разделителем является одинарный обратный слеш).

Структурно рассматриваемая программа состоит из трех частей: основная часть (класс Program и классы-исключения), обработчик входных файлов (в число функций

входит также создание выходных файлов, поскольку они получаются после некоторого конечного числа этапов обработки исходных) и непосредственно работа со стегосистемой.

Обработчик файлов и работу со стегосистемой также можно разделить по направленности действия на части: первая направлена на создание стегосистемы, вторая – на извлечение из нее сообщения.

Все возникающие в процессе работы приложения исключительные ситуации обрабатываются только в основном методе программы (т.е., на каком бы этапе ее работы они не возникали, они будут переданы по стеку вызовов в основной метод и только там соответствующим образом обработаны). Подобная структура позволяет упростить реализацию.

Основной метод выполняет цикл для многократного использования приложения, не закрывая его; каждая итерация цикла сопровождается запросом к пользователю относительно закрытия приложения. Внутренние конструкции ветвления отвечают за определение последовательности действий в зависимости от цели использования приложения и определение момента завершения приложения. Предусмотрена защита от некорректно введенных данных на всех этапах.

Следует отметить, что классы-обработчики файлов статические, поэтому для них отсутствуют конструкторы. В данном приложении это позволило значительно сэкономить ограниченный ресурс памяти.

Данная программа представляет собой консольное приложение, предназначенное для создания стегоконтейнера или извлечения из него ранее сокрытой информации, при сокрытии которой использовалось данное приложение.

Работа программы сопровождается инструкциями с пояснениями, а также обработкой некорректно введенных данных.

Входными данными приложения являются указания по выполнению (определяются однозначно из запросов приложения) и полные пути к файлам – потенциальному стегоконтейнеру и скрываемым данным (разделителем является одинарный обратный слеш).

Интерфейс приложения: английский.

В связи с тем, что формат звуковых данных .wav теряет популярность, предполагается реализация дополнения, которое дает возможность работать с другими форматами звуковых данных.

Также, необходима реализация удобного пользовательского интерфейса, в частности, возможность выбора входных файлов из разворачивающегося каталога и

возможность использования строковых данных в качестве сообщения, а не только считанных из входного текстового файла.

Автор предполагает, что в некоторых случаях может быть актуальным сокрытие не только текстовой информации, но и любой другой, например, бинарной, объем которой не превышает $\frac{1}{4}$ объема звукового файла, поэтому целесообразно реализовать поддержку подобных ситуаций.

Совершенствованию поддается основной метод приложения – LSB – для повышения стойкости к стегоатакам на основе анализа распределения бит можно не использовать для сокрытия полностью нулевые или полностью единичные байты, хотя это усложнит предварительный анализ потенциального стегоконтейнера, уменьшит его пропускную способность и количество таких файлов. В связи с этим может стать актуальной проблема использования не единичного стегоконтейнера, а кортежа, т.е. скрывать сообщение последовательно в нескольких файлах.

Так как оптимизация проекта на высоком уровне не проводилась (исключая оптимизацию, производимую компилятором), автор полагает, что проведение оптимизации позволит снизить затраты времени на выполнение программы. Однако оптимизация не должна повлиять на читаемость и/или расширяемость программного кода – это может затруднить (в худшем случае – даже сделать невозможной) дальнейшую разработку и/или сопровождение программы.

В процессе выполнения работы были досконально изучены проблемы и методы цифровой стеганографии, в частности, стеганографии в звуке. На основании изученного теоретического материала было разработано и реализовано консольное приложение на языке программирования высокого уровня C#, использующее в качестве метода сокрытия информации метод наименьшего значащего бита (LSB).

В результате анализа данного стеганографического метода были выявлены слабые стороны алгоритма и выявлены условия, при которых можно повысить устойчивость стегосистемы, реализованной с его использованием. На основании условий выведена формула шага встраивания – функция, зависящая от трех переменных (объема стегоконтейнера, объема сообщения и частоты дискретизации звукового файла-контейнера). Также сформулированы математические ограничения для шага встраивания.

В процессе реализации приложения использовались библиотечные классы и методы, предназначенные для чтения и записи файлов, обработки данных на низком уровне (на уровне байтов и битов), чтения заголовка файла.

В связи с ограничениями, наложенными на время разработки основной части

приложения, автором изложены предложения по совершенствованию приложения, которые позволят сделать его более функциональным.

По окончании основного процесса программирования был проведен тщательный анализ программы и затем – рефакторинг (улучшение существующего программного кода) с целью исправления основных ошибок проектирования и обеспечения читабельности и расширяемости кода.

Полученное приложение соответствует сформулированному техническому заданию и прошло все стадии разработки, заявленные в данном техническом задании.

Список литературы

1. Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. М.: МК-Пресс, 2006. 288 с.
2. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. М.: Солон-Пресс, 2002. 272 с.
3. Структура WAV файла Режим доступа: <http://audiocoding.ru/статьи/структура-wav-файла.html> (дата обращения 07.09.2014).
4. Программы для стеганографии. Fox Secret. Режим доступа: <http://gr1g0ry.blogspot.ru/2011/08/steganographysoftfoxsecret.html> (дата обращения 07.09.2014).
5. Понятие стеганографической стойкости. Режим доступа: <http://crypts.ru/ponyatie-steganograficheskoy-stojkosti.html> (дата обращения 07.09.2014).
6. Скрытие данных в аудиосигналах. Режим доступа: <http://crypts.ru/skrytie-dannyx-v-audiosignalax.html> (дата обращения 07.09.2014).
7. Max File Encryption 1.8. Режим доступа: <http://www.softeza.com/ru/fileencryption/> (дата обращения 07.09.2014).