Математические структуры и моделирование
2006, вып. 16, с. 88–101

УДК 681.3

ИСПОЛЬЗОВАНИЕ MIDI-ФОРМАТОВ ДЛЯ СТЕГАНОГРАФИЧЕСКИХ КОНТЕЙНЕРОВ

И.В. Русинович

На основе проведенного исследования и анализа современных стеганографических методов и сфер их применения предлагается новый способ скрытия секретной информации в файлах формата MIDI. Приводится программная реализация разработанного метода и краткий анализ полученных результатов.

1. Введение

Задача надежной защиты информации от несанкционированного доступа является одной из древнейших и не решенных до настоящего времени проблем. Способы и методы скрытия секретных сообщений известны с давних времен, причем данная сфера человеческой деятельности получила название стеганография. Это слово происходит от греческих слов steganos (секрет, тайна) и graphy (запись) и таким образом означает буквально «тайнопись», хотя методы стеганографии появились, вероятно, раньше, чем появилась сама письменность. В дальнейшем для защиты информации стали использоваться более эффективные на время создания методы кодирования и криптографии.

Криптография и стеганография представляют собой два метода скрытия сообщений, которые, хотя и взаимно дополняют друг друга, тем не менее идентичными не являются.

- Криптография. При помощи шифрования происходит изменение содержимого послания или файла так, что оно становится нечитаемым для всякого, кто не является авторизованным получателем. Последний, однако, имеет специальный ключ, используя который можно «открыть» файл и прочитать в том виде, как он создавался изначально отправителем. Зашифрованные сообщения не скрываются, так что их отправку и получение можно легко обнаружить и отследить. Если в дальнейшем будет выявлено использованное при шифровании средство, то взломщику кодов останется только подобрать ключ для расшифровки сообщения.

Омский государственный университет.
E-mail: i_rusinovich@mail.ru

- Стеганография. Можно рассматривать стеганографию как мощный метод шифрования. С ее помощью те, кто обменивается информацией, пытаются скрыть сам факт переписки от глаз стороннего наблюдателя. В отличие от криптографии, стеганографию невозможно обнаружить. Часто стеганография используется наряду с шифрованием. Благодаря такой комбинации, закодированное и невидимое сообщение становится полностью защищенным от перехвата.

Хорошо известны различные способы скрытого письма между строк обычного незащищаемого письма: от применения молока до использования сложных химических реакций с последующей обработкой при чтении.

Другие методы стеганографии включают использование микрофотоснимков, незначительные различия в написании рукописных символов, маленькие проколы определенных напечатанных символов и множество других способов по скрытию истинного смысла тайного сообщения в открытой переписке.

Компьютерные технологии придали новый импульс развитию и совершенствованию стеганографии, появилось новое направление в области защиты информации - компьютерная стеганография [1].

Цифровые средства коммуникаций стимулируют появление новых возможностей, в том числе для решения уже известных задач. В качестве примеров можно привести задачи скрытой коммуникации, новые средства защиты авторских прав в области цифровой аудио- и видеоиндустрии, встраивание серийных номеров в программный продукт для отслеживания пиратского его распространения и т.д. Существует много весьма специфических приложений стеганографии в различных областях. Например: военная разведка, где нужны средства скрытых коммуникаций в цифровых средах; органы правосудия и контрразведки, которые нуждаются как в скрытых коммуникациях, так и в доступе к трафику для обнаружения фактов скрытых коммуникаций и для их извлечения, хотя это в настоящее время практически невозможно; технологии, используемые в электронной коммерции и в электронных платежах, где методы стеганографии необходимы как средство аутентификации источника сообщения с помощью электронной подписи; обеспечение защиты прав личности, поскольку стеганография дает возможность исключить или затруднить контроль личной переписки как криминальными элементами, так и государственными органами.

Современный прогресс в области глобальных компьютерных сетей и средств мультимедиа привел к разработке новых методов, предназначенных для обеспечения безопасности передачи данных по каналам телекоммуникаций и использования их в необъявленных целях. Эти методы, учитывая естественные неточности устройств оцифровки и избыточность аналогового видео- или аудиосигнала, позволяют скрывать сообщения в компьютерных файлах (контейнерах). Причем, в отличие от криптографии, данные методы скрывают сам факт передачи информации. В качестве носителя секретной информации может быть использован файл любого типа. Например, секретная информация может быть встроена в текстовый файл, в файлы аудио или видео, в файл изображения, в текст программы и даже в заголовочную часть IP-пакетов. Другими словами,

скрытая (дополнительная) информация может быть встроена в любой файл, если он занимает больше места, чем это минимально требуется.

Как и любые инструменты, стеганографические методы требуют к себе внимания и осторожного обращения, так как могут быть использованы как для целей защиты, так и для целей нападения.

2. Компьютерная стеганографии и стеганографические методы

2.1. Основные принципы компьютерной стеганографии и области ее применения

К. Шеннон дал нам общую теорию тайнописи, которая является базисом стеганографии как науки. В современной компьютерной стеганографии существует два основных типа файлов: сообщение — файл, который предназначен для скрытия, и контейнер — файл, который может быть использован для скрытия в нем сообщения. При этом контейнеры бывают двух типов. Контейнер-оригинал (или «пустой» контейнер) — это контейнер, который не содержит скрытой информации. Контейнер-результат (или «заполненный» контейнер) — это контейнер, который содержит скрытую информацию. Под ключом понимается секретный элемент, который определяет порядок занесения сообщения в контейнер.

Основными положениями современной компьютерной стеганографии являются следующие [3]:

1. Методы скрытия должны обеспечивать аутентичность и целостность файла.

2. Предполагается, что противнику полностью известны возможные стеганографические методы.

3. Безопасность методов основывается на сохранении стеганографическим преобразованием основных свойств открыто передаваемого файла при внесении в него секретного сообщения и некоторой неизвестной противнику информации — ключа.

4. Даже если факт скрытия сообщения стал известен противнику через сообщника, извлечение самого секретного сообщения представляет сложную вычислительную задачу.

2.2. Стеганографические методы

В настоящее время методы компьютерной стеганографии развиваются по двум основным направлениям:

1. Методы, основанные на избыточности аудио- и визуальной информации.

Младшие разряды цифровых отсчетов содержат очень мало полезной информации. Заполнение их дополнительными данными практически не влияет на качество восприятия.

2. Методы, основанные на использовании специальных свойств компьютерных форматов.

   Среди них можно выделить следующие: методы использования зарезервированных для расширения полей компьютерных форматов данных, методы специального форматирования текстовых файлов, методы скрытия в неиспользуемых местах гибких дисков, методы использования имитирующих функций, методы удаления идентифицирующего файл заголовка.

Выбор и применение того или иного стеганографического метода полностью зависит от типа используемого контейнера и требований, предъявленных к процессу записи сообщения в контейнер и последующей передачи этого контейнера от отправителя к получателю.

2.3. Требования к стеганографическим методам

Рассмотрим основные требования к методам цифровой стеганографии [2]:

1. «Прозрачность». Требование «прозрачности», отсутствия видимых различий между оригиналом и файлом со встроенной информацией является общим по отношению ко всем задачам стеганографии. К сожалению, не существует формальных критериев оценки степени прозрачности встроенной информации, поскольку она оценивается человеком, а различные субъекты могут иметь на этот счет различные мнения.

2. Робастность (устойчивость). В общем случае, в особенности в случае встраивания водяных знаков, встроенная информация должна быть робастной по отношению к случайным возмущениям, свойственным процессам передачи информации, к преобразованиям типа код/аналог и аналог/код и к сжатию с относительно небольшими потерями. Конкретные требования всегда уточняются приложением. Стоит отметить, что требование робастности всегда вступает в противоречие с требованием прозрачности встроенной информации.

3. Устойчивость по отношению к попыткам удаления встроенной информации. Встроенная (скрытая) информация (например, водяной знак) может быть объектом целенаправленных попыток ее зашумления или удаления. Такие попытки принято называть атаками. Устойчивость скрытой информации по отношению к атакам является одним из наиболее важных требований к методам стеганографии. Вообще говоря, абсолютной устойчивости встроенной информации по отношению ко всем возможным атакам или их комбинациям добиться невозможно, если при этом не жертвовать разрушением контекста оригинала. Базовый принцип — это обеспечение

определенного уровня устойчивости по отношению к заданному множеству атак в рамках соответствующего приложения.

4. Возможность встраивания заданного относительного объема информации. Объем информации, которая должна быть записана в файл-контейнер, зависит от приложения, и этот объем обычно оценивается относительной величиной по отношению к размеру файла-контейнера. Существует противоречие между возможным объемом встроенной информации и «прозрачностью», а также робастностью. Если требуется встроить большой объем информации, то придется пойти на уменьшение «прозрачности» и снижение робастности. Обычно отыскивается компромисс между этими тремя характеристиками.

5. Секретность маркировки. Для большинства приложений требуется обеспечить секретность встроенной информации, т.е. ее защиту с помощью секретного ключа. Обычно рассматриваются два уровня обеспечения секретности. Первый из них должен предотвратить неавторизованный доступ к декодированию (извлечению) встроенной информации или идентификации факта ее присутствия. Во втором варианте допускается, что неавторизованный пользователь может обнаружить присутствие встроенной информации, однако она должна быть защищена от извлечения. Второй случай обычно используется тогда, когда в изображение встроена пара водяных знаков, один из которых имеет открытый ключ доступа, а другой защищен секретным ключом или ключами.

2.4. Разделение секрета

После встраивания информации в файл наиболее важной задачей является усложнение процесса получения секретной информации неавторизованными пользователями. Как уже упоминалось, один из способов основывается на секретной маркировке. Другой способ, обеспечивающий устойчивость по отношению к попыткам удаления встроенной информации, называется разделением секрета. Суть его состоит в следующем: при помощи каких-либо алгоритмов исходное сообщение разделяется на m составляющих (долей) так, что каждая отдельная составляющая сама по себе ничего не значит, и, лишь собрав все доли вместе и применив к ним обратный алгоритм, можно получить исходные данные. Получается, что даже если противнику попадет одна или несколько долей, но не все, то он все равно не сможет узнать секретную информацию, правда, при этом потеряется факт секретности существования сообщения. Но предположим такую ситуацию: некто, не являющийся авторизованным получателем, обнаружил несколько долей и уничтожил их. Получается, что не только он не смог получить скрытую информацию, но и тот, кому она предназначалась. Даже необязательно преднамеренное пагубное воздействие, ведь один из стеганографических контейнеров либо его часть могут быть повреждены в процессе передачи от отправителя к получателю, и снова секретная информация не сможет

быть восстановленной. Во избежание подобного рода неблагоприятных ситуаций существует так называемая (m, n)-пороговая схема: сообщение разделяется на n долей так, чтобы по любым m из них можно было восстановить исходные данные. Имея меньше, чем m долей, ничего сделать нельзя, а от $m + 1$ до n долей хватит с избытком. Существует множество алгоритмов для осуществления (m, n)-пороговой схемы, приведение которых в данной работе мы обойдем. Некоторые из них описаны в [4].

3. Разработка технологии встраивания информации в MIDI-контейнер

3.1. Формат файла MIDI

Файлы с оцифрованным звуком содержат значения амплитуды звукового сигнала, измеренные через одинаковые промежутки времени, в то время как файлы с нотной записью (в частности MIDI) содержат последовательность команд, сообщающих, какую ноту и каким инструментом и как долго нужно воспроизводить в тот или иной момент времени. В связи с этим количество информации в файлах с нотной записью в сотни и тысячи раз меньше, чем в файлах с оцифрованным звуком [5]. Это заметно ограничивает возможности записи данных методами, основанными на избыточности информации файла-контейнера. Неудивительно, что в этом направлении проводится гораздо меньше исследований. Однако использование специальных свойств форматов может предоставить большие возможности для скрытия информации от посторонних глаз.

Стандартный MIDI-файл — это специально разработанный формат файлов, предназначенный для хранения данных, записываемых и/или исполняемых секвенсером. Секвенсер может быть как программой для компьютера, так и аппаратно выполненным модулем.

В этом формате хранятся стандартные MIDI-сообщения (т.е. статус-байты и соответствующие им байты данных), а также временные метки или маркеры для каждого сообщения (т.е. последовательности байтов, указывающие, какое количество условных единиц времени необходимо подождать перед тем, как исполнить следующее событие MIDI). Этот формат позволяет сохранять информацию о темпе, временном разрешении, выраженном в количестве тиков на одну четвертную длительность, обозначения размера, информацию о музыкальных ключах, а также хранить названия треков и паттернов.[1] Формат предусматривает возможность сохранения в одном файле нескольких паттернов и треков таким образом, что программы-приложения могут выбирать из всего набора хранимой информации ту, которая будет понятна данному приложению.

Формат разработан так, чтобы любой секвенсер мог читать и записывать MIDI-файл таким образом, чтобы не потерялись его данные, и так, чтобы формат был достаточно гибким, т.е. чтобы приложения могли сохранять в файлах

[1]Треки используются в MIDI-файлах и MIDI-секвенсерах для записи MIDI-сообщений. Назначение аналогично дорожкам ленточного магнитофона. Набор треков, воспроизводимых одновременно, образует паттерн.

свою специфическую информацию, понятную только этим приложениям, но не понятную другим программам, причем при загрузке файлов MIDI непонятная другим программам-приложениям информация не приводит к недоразумениям, а просто игнорируется [6].

Данные всегда хранятся в виде записей. В одном MIDI-файле могут сосуществовать несколько различных записей. Каждая запись может иметь свой собственный размер, т.е. количество байтов в различных записях может быть различно. Каждая запись начинается с указания ее идентификатора, который состоит из четырех букв, т.е. из четырех ASCII байтов. Этот идентификатор указывает, какой тип записи представлен в содержащихся в записи байтах данных. Последующие за идентификатором четыре байта (каждый из которых состоит из 8 бит) образуют 32-битное значение, указывающее длину (или размер) данной записи. Все записи должны начинаться с этих двух полей: идентификатора записи и размера записи. Эти два поля, занимающие всего 8 байт, образуют заголовок записи.

Фактически все MIDI-файлы начинаются с заголовка MThd, и именно этот факт является указанием на то, что мы имеем дело со стандартным MIDI-файлом. Запись MThd имеет длину 6 байт. Первые два байта данных содержат информацию о формате или типе MIDI-файла. Существует три различных типа (формата) MIDI-файлов. Тип 0 означает, что файл содержит MIDI-данные, записанные на одном трэке, файл типа 1 подразумевает, что в нем содержатся несколько (но возможно и всего один) одновременно воспроизводимых трэков, файлы типа 2 содержат один или несколько независимых трэков, каждый такой трэк образует свой собственный паттерн. Следующие два байта определяют количество треков, хранимых в файле, эти два байта обозначаются NumTracks. Естественно, что для файлов типа 0 значение NumTracks всегда равно 1. Для двух оставшихся типов возможны другие значения. Оставшиеся два байта указывают величину временного разрешения, т.е. количество временных импульсов (временных тиков), приходящихся на одну четвертную длительность.

За записью MThd следует MTrk-запись. Это единственный тип записи, отличающийся от MThd-записи, который определен для MIDI-файлов в настоящее время. Если по какой-либо причине в файле содержится идентификатор какой-либо записи другого типа, то, вероятно, эта запись создана для какой-либо другой программы, и эта запись должна быть проигнорирована в соответствии с указанной в ее заголовке длиной данной записи [7].

3.2. Предлагаемые методы занесения сообщения в MIDI-контейнер

Первый из предлагаемых методов скрытия информации в MIDI-файле основан на способности секвенсеров считывать информацию только из записей известных им типов. Можно стандартизировать какой-либо из несуществующих типов записей и использовать его в качестве хранилища данных, причем, при указании в начале такой записи ее размера, абсолютно исключается возможность обнаружения «различий на слух», т.к. все секвенсеры ее будут пропускать.

Однако использование данного метода влечет за собой двоякое понимание требования «прозрачности». По аналогии с файлами других форматов это должно означать отсутствие (точнее сведение к минимуму) отличий в восприятии незаполненного и заполненного файла-контейнера, другими словами этого не должно быть слышно. Как было сказано выше, в этом варианте требование «прозрачности» выполняется. Но в отличие от секвенсеров любой специалист наверняка знает, какой набор байт может, а какой не может быть самостоятельной и «безобидной» записью, поэтому при непосредственном просмотре двоичного кода такая запись с высокой вероятностью будет обнаружена, что заметно снижает эффективность ее использования.

Следующий из предлагаемых методов также использует свойства формата MIDI. Предлагается записывать информацию в файл посредством генерирования нот определенной высоты и, возможно, длительности. С целью уменьшения вероятности обнаружения сообщения генерацию нот следует проводить по определенным законам существующих музыкальных гармоний. Так, например, можно расставлять ноты в соответствии с любым натуральным ладом, в данном случае в пределе одной октавы будем иметь семь различных нот. Во избежание возможных больших «скачков» в тональностях двух соседних нот следует ограничиться рассмотрением лишь названия ноты, не принимая во внимание октаву за одним исключением: чтобы один ключ (т.е. одна нота в MIDI-файле) нес целое число бит информации, необходимо добавить еще одну ноту, обособив ее от остальных. К примеру, можно каждую ноту «до» четной октавы отличать от ноты «до» нечетной октавы. В данном случае получается метод записи, абсолютно не заметный при визуальном просмотре кода, но тем не менее ощутимый на слух, при котором каждый ключ MIDI-файла может нести в себе 3 бита дополнительных данных. Особо обратим внимание на то, что при этом методе передаваемая информация не просто записывается в контейнер, а сама впоследствии становится его частью, после чего файл с уже скрытой информацией может рассматриваться как пустой контейнер, в который можно записать другие данные, правда, уже иным методом. С другой стороны, в качестве контейнера можно рассматривать MIDI-файл без нот, в этом случае данный метод не является принципиально отличным, однако в такой пустой контейнер можно записать информацию далеко не всеми оставшимися методами.

Другими методами, использующими свойства формата MIDI, может служить запись данных в специальные поля, отведенные для текста композиции, названий треков, комментариев, информации об авторе, в значение темпа композиции, в переменные, отвечающие за расстановку нот по каналам, правда, при различной емкости этих элементов контейнера такие данные будут обнаружены с весьма высокой степенью вероятности, поэтому особое внимани на них заострять не будем.

Среди методов, основанных на избыточности информации в пустом контейнере, следует выделить запись данных в младшие биты временных меток (deltatime), а также статической характеристики каждой конкретной ноты - velocity (сила нажатия). Выделив по 3 младших бита в каждом элементе контейнера, можно получить весьма слабо заметные отличия заполненного контейнера от

пустого как на слух, так и при визуальном исследовании двоичного кода файла (изменения в delta-time визуально определить практически невозможно). В пользу этих методов выступает возможность записывать непосредственно сыгранные музыкантом партии в виде команд в MIDI-файл. Очевидно, что как сила нажатия на клавишу, так и интервалы между нотами в данном случае никак не могут быть абсолютно одинаковыми, что уменьшает вероятность обнаружения скрытой информации практически до нуля.

3.3. Реализация пороговой схемы

Для усложнения процесса обнаружения секретной информации неавторизованными пользователями предлагается реализовать пороговую схему методом интерполяционных полиномов Лагранжа. В этом методе используются полиномиальные уравнения в конечном поле. Выберем простое число p, которое больше количества возможных долей и больше самого большого из возможных секретов. Чтобы сделать секрет общим, сгенерируем произвольный многочлен степени $m-1$:

$$(a_1 x^{m-1} + a_2 x^{m-2} + ... + a_{m-1} x + M) \ mod \ p.$$

Здесь p — случайное простое число, большее любого из коэффициентов. Коэффициенты a_i выбираются случайным образом, они хранятся в тайне и отбрасываются после распределения долей. M — это сообщение. Простое число должно быть опубликовано. Доли получаются с помощью вычисления многочлена в n различных точках: $k_i = F(x_i)$. Иначе говоря, первой долей может быть значение многочлена при $x = 1$, второй долей — значение многочлена при $x = 2$ и т. д.

Поскольку в многочленах степени $m-1$ имеются m неизвестных коэффициентов $a_1, \ldots, a_{m-1}$ и M, для создания m уравнений можно использовать любые m долей. Меньше, чем m долей, не хватит, а от $m+1$ до n долей хватит с избытком.

Чтобы восстановить сообщение M по m долям, решается система линейных уравнений

$$a_1 x^{m-1} + a_2 x^{m-2} + ... + a_{m-1} x + M = k_i \ (mod \ p),$$

после чего становятся известными коэффициенты a_i и само сообщение M [1].

3.4. Распределение долей по слоям

Будем рассматривать любое сообщение в виде набора байт. Для каждого байта применим метод интерполяционных многочленов Лагранжа, в результате чего получим n новых байт (долей). Переходя к разделению следующего байта сообщения, можно либо использовать то же простое число, что и на предыдущем шаге, записав его однажды в MIDI-контейнер, либо генерировать новое, в результате чего помимо n долей получим и n различных простых чисел, своеобразных ключей для каждого байта исходных данных. Их тоже нужно

как-то сохранить в контейнере, обеспечив защиту от потери при передаче либо в результате враждебного воздействия. При реализации первой схемы (общее простое число p для всех байт сообщения) можно записать ключ в начале или конце каждой последовательности байт (долей), чем будет гарантирована их сохранность даже при уничтожении одного или нескольких слоев. Если же для каждого байта свой ключ, то получившиеся n чисел можно прописать в отдельном слое (слоях), однако в данном случае потеряется смысл пороговой схемы, т.к. при уничтожении всех слоев с ключами сообщение становится невосстанавливаемым, даже если все n слоев с долями останутся целыми. Предлагается два варианта решения этой проблемы.

Первый заключается в записи ключа после каждого байта в доле сообщения. Правда, основываясь на принципе реализации пороговой схемы, можно значительно сократить количество долей (а следовательно, и слоев), содержащих ключи. Пусть существует (m, n)-пороговая схема, в k слоях которой за каждым байтом сообщения записан ключ (еще один байт), оценим число k. Т.к. в (m, n)-пороговой схеме допускается уничтожение $(n - m)$ слоев без повреждения самого сообщения, то число слоев с ключом должно быть строго больше этой величины, но m, n, k - натуральные числа, значит, наименьшее такое число в точности равно $(n - m + 1)$. Очевидно, что доля с ключами по размеру будет ровно в два раза больше доли без ключа, поэтому имеет смысл распределять доли по слоям в соответствии со схожестью в емкости разных методов записи. Также можно две доли без ключа записать в один слой (тогда длина последовательности удвоится и станет равна длине доли с ключами), но это снова снижает надежность схемы.

Второй вариант состоит в записи отдельных слоев, состоящих из ключей, теми же способами, что и слои с долями сообщения, но в другие контейнеры либо в другие одноименные записи того же контейнера. Обратим внимание на то, что все многообразие предлагаемых методов скрытия сообщения, существующее в пределах одной записи, нисколько не меняется при переходе к другой одноименной записи, и если мы предположим наихудший для нас вариант — преднамеренную порчу некоторых слоев, то одинаковые методы в разных записях будут уничтожены одновременно. (Если противник поймет, что в данной записи есть скрытое сообщение или его доля, то по той же технологии удалит сообщение либо долю и из другой записи.) Таким образом, уничтожение всех слоев с ключами будет возможно лишь при уничтожении всех слоев с долями сообщения. Но обратим внимание, что и в этом случае для хранения ключей достаточно использовать лишь $(n - m + 1)$ слоев по причине, указанной выше.

В обоих вариантах требуемый размер дополнительного пространства для записи ключей одинаков и равен $(n - m + 1) \cdot s$ байт, где s - длина сообщения в байтах. Поэтому выбор наиболее эффективного варианта зависит скорее от конкретных условий передачи данных.

4. Реализация алгоритма встраивания информации в MIDI-контейнер

4.1. Отбор методов. Алгоритм записи сообщения в MIDI-контейнер

В предлагаемом алгоритме записи информации в MIDI-контейнер используются как элементы избыточности пустого контейнера, так и некоторые конкретные свойства данного формата.

Из всех предложенных способов записи сообщения в MIDI-контейнер наиболее эффективными в смысле визуальной скрытности являются способы, основанные на изменении значения delta-time и velocity. Но для реализации пороговой схемы необходимо выбрать число n не менее трех. Относительно слухового восприятия абсолютно скрытыми являются способы записи в специальные поля файла либо в отдельную запись несуществующего типа. Последний вариант является более эффективным, однако стандартизирование более чем одного типа записи по понятным причинам бессмысленно.

Предлагается реализовать (2,3)-пороговую схему, использовав 3 вышеперечисленных способа записи информации в MIDI-контейнер: изменения delta-time, velocity и стандартизирование дополнительного типа записи. Можно было бы добавить в алгоритм несколько менее надежных способов без изменения высоты порога (т.е. для наращивания n), но во избежание провокации удаления всего контейнера в случае обнаружения в нем любых признаков содержания секретного сообщения мы этого делать не будем. Итак, сообщение разделяется на 3 слоя по схеме интерполяционных многочленов Лагранжа, ключи для каждого байта выбираются разными и записываются теми же способами в другую MTrk-запись того же контейнера (т.е. в качестве контейнера необходимо выбирать MIDI-файл формата 1), причем слоев с ключами будет ровно 2 и все они будут реализованы в отдельной MTrk-записи.

4.2. Программная реализация алгоритма встраивания информации в MIDI-контейнер

Предложенный алгоритм встраивания информации в MIDI-контейнер с некоторыми изменениями реализован в прикладном программном продукте ORCHID, MIDI Steganographer 2.0. Программа предназначена для скрытия информации в MIDI-файлах формата 0. Скрываемая информация считывается по байтам из текстового файла. Каждый байт разделяется на 3 доли по схеме интерполяционных многочленов Лагранжа, ключ для каждого байта остается одинаковым и помещается в начало каждого слоя. Таким образом, повреждение или уничтожение одного из слоев не повлечет за собой потерю ключа. При повреждении или уничтожении двух или более слоев сообщение восстановлению не подлежит. Конец каждого слоя помечен ASCII-байтом 1.

Первый слой записывается в младшие биты значения delta-time. Первоначальная реализация алгоритма использовала 3 младших бита, однако достаточно сильно страдала «прозрачность». Исследование параметров MIDI-файлов

показало, что меток delta-time в среднем в 2 раза больше, чем midi-событий "key on"(«нажатие клавиши»), но т.к. объемы всех долей секрета одинаковы, примерно половина меток delta-time остается неиспользованной. Исходя из этого было решено в delta-time записывать по 1 биту секрета. В результате емкость контейнера уменьшилась в 1,5 раза, но несравненно улучшилось требование «прозрачности».

Второй слой записывается в 3 младших бита значения velocity в событии «key on», третий - в новую запись. Для маскировки имя новой записи выбрано как MTrk, где отличие от имени стандартной MTrk-записи состоит в том, что вторая буква (T) - русская, а значит, имеет другую кодировку и секвенсером идентифицируется иначе.

В программе предусмотрен режим извлечения информации. В процессе извлечения все 3 слоя считываются (при отсутствии слоя, если он был удален, или его части, значение доли принимается равной нулю), вычисляется 3 варианта исходного сообщения по каждой паре долей в слоях. После все 3 сообщения сравниваются, и пользователю выводится результат:

1. Все сообщения одинаковы — повреждения в слоях не обнаружены, извлечение успешно.

2. Два сообщения одинаковы, третье отличается — один из слоев поврежден (номер слоя также идентифицируется), сообщение восстановлено по двум оставшимся слоям.

3. Все сообщения отличны — 2 или более слоев повреждены, восстановление информации невозможно.

4.3. Анализ эффективности реализованного алгоритма

Исследование параметров midi-файлов с целью определения потенциальной емкости для каждого слоя сообщения было начато выше. Можно добавить лишь, что емкость для третьего слоя не ограничена, следовательно, никак не влияет на два остальных.

Исследование на предмет «прозрачности» (при прослушивании опытными музыкантами) дало хорошие результаты. Возможные отклонения от нормы («плавающий темп», неоднородная игра инструментов) обнаружены не были.

Повышение устойчивости по отношению к попыткам удаления встроенной информации обеспечивается пороговой схемой. Робастность исследовалась посредством конвертирования заполненных файлов-контейнеров в другие форматы (MIDI 1, RIFF MIDI, WRK) и обратно, т.к. программа работает только с файлами формата MIDI 0. В результате первые два слоя всегда оставались целыми, а третий, помещенный в отдельную запись, постоянно удалялся секвенсером. Тем не менее сообщение всегда удавалось восстановить по двум слоям. Но если в том же секвенсере изменить какой-нибудь элемент из первых двух слоев, то сообщение теряется, т.к. поврежденными оказываются уже 2 слоя.

Возможный (максимально возможный) объем встраиваемой информации определен разработчиком и неизменен. В силу специфических свойств формата MIDI он не является функцией размера файла-контейнера, а зависит от конкретных параметров: количества временных меток delta-time и midi-событий «key on». Исследования различных midi-контейнеров дают усредненный результат 1:30. Для сравнения: в графических и мультимедиа файлах - 1:10. Учитывая кажущуюся непригодность и, как следствие, малое исследование данного направления в стеганографии — это весьма приемлемый результат. Сюда же стоит отнести и немалый потенциал MIDI-стеганографии, ведь исследования в данной области только начались.

Секретность маркировки не обеспечена, однако большой сложности данная задача не вызывает и сводится к совместному использованию криптографических и стеганографических методов.

В реализованном алгоритме соблюдены требования основных положений современной компьютерной стеганографии: аутентичность и целостность файла обеспечена, после применения стеганографического преобразования сохраняются основные свойства открыто передаваемого файла, если факт скрытия сообщения станет известен противнику через сообщника, благодаря пороговой схеме извлечение самого секретного сообщения будет представлять сложную вычислительную задачу. Для защиты от повреждения при передаче данных и от преднамеренного враждебного воздействия реализована (2,3)-пороговая схема. Схема интерполяционных многочленов Лагранжа выбрана из-за наименьшей трудоемкости при реализации.

5. Заключение

Хотя цифровая стеганография является относительно новой областью исследований, развитие цифровых сред и связанные с этим практические потребности стимулируют интенсивный рост интереса к этой области во всем мире и чрезвычайно высокую активность исследователей в последнее десятилетие. К настоящему моменту в этой области предложено достаточно большое множество методов, технологий и инструментальных средств. Огромный интерес к стеганографии в значительной мере стимулируется быстрым распространением Интернет, возрастанием роли Интернет-технологий и соответствующих цифровых сред в практической жизни общества. Последнее ведет к необходимости защиты аудио- и видеоинформации перед тем, как она будет публиковаться в Интернет или распространяться средствами Интернет. Существует также много других причин, обуславливающих чрезвычайно высокий интерес к использованию методов стеганографии, в частности, в военном деле, в юриспруденции, электронной коммерции, а также в других областях, где необходимо обеспечивать секретность коммуникаций, защищать авторские права, предотвращать пиратское использование программного продукта и т.п.

Использование MIDI-форматов для стеганографических контейнеров позволяет скрывать меньший объем секретной информации, чем использование графических и аудиоформатов (примерно 1:30 часть объема контейнера про-

тив 1:10). Тем не менее при огромной распространенности различных методов стеганографии в последних разработках алгоритмов для MIDI предоставляет весьма большие возможности, в первую очередь из-за кажущейся непригодности формата для данных целей, а следовательно, и гораздо более низкой степени контроля при проверке файлов на предмет содержания скрытых сообщений. Не следует упускать из виду и немалый потенциал MIDI-стеганографии как одного из наименее исследованных направлений современной компьютерной стеганографии.

Литература

1. Голубев В. Компьютерная стеганография – защита информации или инструмент преступления? – http://www.crime-research.ru/
2. Городецкий В.И., Самойлов В.И. Стеганография на основе цифровых изображений. Санкт-Петербургский институт информатики и автоматизации РАН.
3. Барсуков В.С., Романцов А.П. Компьютерная стеганография вчера, сегодня, завтра. Технологии информационной безопасности 21 века // Специальная техника. 1998. N.4-5. – http://st.ess.ru/
4. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Издательство ТРИУМФ, 2003.
5. Новосельский А. Форматы звуковых файлов // Компьютеры+Программы. 1996. N.1. – http://www.aip.mk.ua/
6. Будило В. Формат файлов MIDI. – http://www.comprice.ru/
7. Dustin Caldwell. Standard MIDI File Format. – http://www.mp.dpt.ustu.ru/