Оглавление

Перечень сокращений, символов и специальных терминов

Контейнер – цифровая информация, используемая для хранения другой цифровой информации.

Аудио-контейнер – аудиоданные, используемые для хранения цифровой информации.

ЦОС – цифровая обработка сигналов.

Стеганосистема – система, использующая стеганографические алгоритмы.

Стеганосредства – набор инструментов для стеганографической обработки.

ССЧ – слуховая система человека.

Стеганодетектор – устройство для обнаружения применения стеганографии.

Стеганометод – стеганографический метод.

Введение

В настоящее время вопрос об авторских правах, правах интеллектуальной собственности, о защите от несанкционированного доступа является одним из важнейших и всё ещё не решён. Информация преимущественно хранится в цифровом виде по причине интенсивного развития и распространения технологий, которые позволяют обрабатывать большое количество данных и воспроизводить различные типы сигналов, применяя современные вычислительные машины. Данные, представленные в цифровом виде, восстановимы, имеют значительную помехоустойчивость. Однако эти преимущества становятся незаметными, когда возникает угроза похищения и модификации. Для защиты информации разрабатываются и внедряются методы криптографии и стеганографии.

Криптографический подход не решает в полной мере задачу защиты данных, поскольку шифрование само по себе привлекает внимание и провоцирует злоумышленника, завладевшего секретной информацией к дешифровке.

Стеганографическая защита скрывает сам факт существования секретных данных. Наука изучает способы и методы скрытия конфиденциальных данных. Однако наилучшие результаты достигаются именно путём совместного использования стеганографии и криптографии. Стеганография применяется не только для скрытой передачи (или хранения) данных, но и для защиты от копирования, скрытой аннотации, устойчивой аутентификации, отслеживания распространения информации, поиска информации в мультимедийных базах данных, скрытой связи, преодоления систем мониторинга и управления сетевыми ресурсами, а также для «камуфлирования» программного обеспечения, создания скрытых каналов утечки информации.

Общей чертой всех способов стеганографирования является скрытие сообщения в непривлекающий внимание объект, который затем пересылается адресату по общедоступному каналу передачи.

1. Актуальность темы

Несмотря на то, что заложенная идея существует уже несколько веков, цифровая стеганография не так сильно популяризована и распространена, как шифрование, что делает её прекрасным кандидатом для исследований, а также для разработки алгоритмов и их улучшений. В вводном разделе были представлены разнообразные варианты применения стеганографии. Тем не менее, наиболее востребованной всё ещё является именно сфера защиты информация, а популярность исследований опирается на следующие причины:

- Ограничение на использование криптографических средств в некоторых странах мира;
- Существование проблемы защиты прав собственности на информацию, представленную в цифровом виде;

2. Постановка задачи

По итогам анализа предметной области и поиска существующих решений было выбрано направление в стеганографии, для которого пока не существует реализации и описания строго канонического вида, а также не были найдены свободно распространяемые продукты, что предоставляет простор для исследований и новых решений.

Цели выпускной работы:

- Разработка алгоритма для скрытия информации в режиме реального времени в аудиопотоке.

- Разработка программного обеспечения, использующего представленный алгоритм, для выполнения скрытой передачи/ хранения конфиденциальной информации.

3. Анализ предметной области

3.1. Ключевые направления

Наиболее обобщающая классификация стеганографии разделяет её на две группы:

- Связанное с ЦОС
- Не связанное с ЦОС

Цифровая обработка сигналов используется для встраивания секретного сообщения непосредственно в цифровые данные (аудиосигнал, видеозапись, изображение, речь, текстовый документ, исполняемые файлы программ).
Во второй категории конфиденциальная информация размещается в заголовках файлов или в пакетах данных. Однако из-за относительной простоты извлечения/уничтожения скрытых данных это направление не получило сильного развития.

3.2. Построение стеганографической системы

Построение стеганосистемы требует соблюдения ряда требований:

- Приемлемая вычислительная сложность реализации, а именно количество арифметическо-логических операций, необходимых для процесса встраивания/извлечения информации в/из контейнера.
- Обеспечение необходимой пропускной способности (максимального количества информации, которая может быть встроена в один элемент контейнера).
- Сохранение целостности секретной информации для авторизованного лица.
- Если злоумышленник имеет представление о деталях реализации стеганосистемы, то без ключа, он не должен установить факт наличия и содержание скрытого сообщения.
- При выявлении факта скрытия злоумышленнику не должно быть известно содержимое.

3.3. Классификация контейнеров

3.3.1. Фиксированный контейнер

Размеры и характеристики заранее известны. Такой контейнер может быть избранным, случайным или навязанным. Избранный контейнер зависит от встроенного сообщения и может иметь вид его функции. Случайный контейнер наиболее часто используется на практике. Навязанный контейнер может предоставляться при возникновении подозрений о возможности скрытой передачи и желании предотвратить её.

3.3.2. Потоковый контейнер

Представляет собой непрерывно меняющуюся последовательность битов, встраивание сообщения в которую происходит в реальном времени, поэтому размер контейнера заранее неизвестен, но может быть контролируем. Основная проблема потокового контейнера, которая одновременно является и его преимуществом с точки зрение безопасности это выполнение синхронизации, определение начала и конца встроенного сообщения.

Разумеется, что для удовлетворения всем требованиям стеганографии, а также деталям реализации сообщение перед интеграцией требуется зашифровать и привести к удобному для встраивания виду. Причём рекомендуется использовать помехоустойчивое кодирование, так как в процессе передачи контейнер может подвергаться модификациям: изменению объёма, преобразованию в другой формат, применению алгоритмов с потерей данных и др.

3.4. Проблема устойчивости стеганографических систем

Для стеганографии существует конфликт между размером встроенного сообщения и устойчивостью к внешним воздействиям. При неизменном размере контейнера зависимость между размером сообщения и устойчивостью выглядит следующим образом:

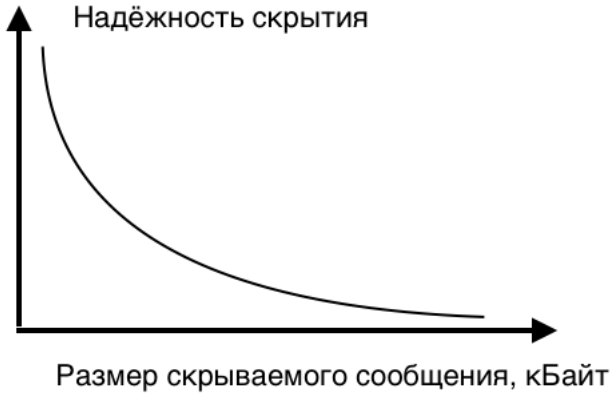

Рис. 3.1. Взаимосвязь между устойчивостью стеганосистемы и объёмом скрываемого сообщения при неизменном размере файла-контейнера. [1]

Благодаря избыточности информации, существует возможность повысить степень надёжности скрытия, жертвуя пропускной способностью. На отношение устойчивости к размеру сообщения также влияет тип контейнера.

3.5. Оценка качества стеганосистемы

Разработка надёжного стеганографического средства предусматривает наличие определённых инструментов для его контроля и оценки. Количественная оценка стойкости к внешним воздействиям - задача, реализуемая методами системного анализа, математического моделирования и экспериментального исследования.

Достаточно эффективны в некоторых случаях методы оценки уровня скрытности стеганосредств на основании анализа их статистических характеристик. Статистическая теория даёт количественные критерии вероятности, что позволяет создавать детекторы, которые будут обнаруживать статистические расхождения между последовательностями. На начальном этапе анализа можно воспользоваться традиционными статистическими (хи-квадрат, тесты на запрещённые символы, на длину цикла и т.д.), эмпирическими (проверка частот, серий, интервалов, перестановок, проверка на монотонность, «покер-тест», тест «собирателя купонов») или спектральными тестами. Больше всего показателей разработано для стеганометодов, которые работают с изображениями или видео. Однако их можно адаптировать и для аудиостеганографии. Наиболее популярным показателем при анализе уровня искажений является соотношение «сигнал/шум» (из радиотехники), вычисленное в децибелах. Иногда методы специально разрабатываются под конкретную задачу [1]. Ниже представлена таблица наиболее распространённых показателей аудио искажения, основанные на анализе структуры контейнера.

Разностные показатели искажения	
Максимальная разность, **MD**	$MD = \max_n \lvert C_n - S_n \rvert$
Средняя абсолютная разность, **AD**	$AD = \frac{1}{N} * \sum_n \lvert C_n - S_n \rvert$
Нормированная средняя абсолютная разность, **NAD**	$NAD = \sum_n \lvert C_n - S_n \rvert \Big/ \sum_n \lvert C_n \rvert$
Среднеквадратическая ошибка, **MSE**	$MSE = \frac{1}{N} * \sum_n (C_n - S_n)^2$
Нормированная среднеквадратическая ошибка, **NMSE**	$NMSE = \sum_n (C_n - S_n)^2 \Big/ \sum_n (C_n)^2$
L^p - норма	$L^p = \left(\frac{1}{N} * \sum_n \lvert C_n - S_n \rvert^p \right)^{1/p}$

Отношение «сигнал/шум», **SNR**	$SNR = \sum_n (C_n)^2 \Big/ \sum_n (C_n - S_n)^2$
Максимальное отношение «сигнал/шум», **PSNR**	$PSNR = N * \max_n (C_n)^2 \Big/ \sum_n (C_n - S_n)^2$
Качество звучания, **AF**	$AF = 1 - \sum_n (C_n - S_n)^2 \Big/ \sum_n (C_n)^2$
Корреляционные показатели искажения	
Нормированная взаимная корреляция, **NC**	$NC = \sum_n (C_n * S_n) \Big/ \sum_n (C_n)^2$
Качество корреляции, **CQ**	$CQ = \sum_n (C_n * S_n) \Big/ \sum_n (C_n)$
Структурное содержание, **SC**	$SC = \sum_n (C_n)^2 \Big/ \sum_n (S_n)^2$

Таблица 3.1. Основные показатели искажения.

3.6. Скрытие данных в аудиосигналах

Цифровые методы в аудиосреде получили особое развитие и в настоящий момент являются особенно перспективными. Несмотря на то, что отклонения в звуковом файле могут быть выявлены до одной десятимиллионной (на 70 дБ ниже уровня внешних шумов), существуют определённые возможности для скрытия информации в аудиосреде. Хотя ССЧ имеет широкий динамический диапазон, она характеризуется достаточно малым разностным диапазоном. Как следствие громкие звуки маскируют тихие. ССЧ распознаёт относительную фазу, но не абсолютную. Многие искажения вызваны окружающей средой и поэтому в большинстве случаев игнорируются.

4. Разработка потокового алгоритма

Существующие решения по скрытию информации ориентированы в основном на работу с фиксированными контейнерами. А так как большая часть информации в открытом доступе посвящается именно встраиванию информации в контейнеры с фиксированной длиной и размером, системы, работающие в реальном времени, как правило, являются либо закрытыми коммерческими продуктами, либо исследовательскими работами в различных университетах мира.

Потоковая обработка сигнала означает, что входящий сигнал подаётся небольшими порциями (фреймами) в алгоритм обработки. В случае с аудиосигналом заводится буфер, сохраняющий звуковой сигнал длительностью в несколько миллисекунд. Размер его варьируется и определяется как поставленной задачей, требованиям алгоритма и частотой дискретизации аудиопотока, так и возможностями и ограничениями звуковой карты и процессора компьютера, на котором осуществляется обработка. При этом необходимо успеть модифицировать эту порцию сигнала до прихода следующей, иначе появляется эффект, получивший название «отброшенные кадры» (dropped frames). Он свидетельствует о потере аудио данных, которая происходит, если процессор не успевает обработать фрагмент сигнала за допустимое время.

Ещё один важный вопрос, с которым пришлось столкнуться при разработке потокового алгоритма, это вопрос о способе выполнения синхронизации, т.е. определении начала и конца сообщения. Для извлечения сообщения необходимо выяснить «точку входа», начиная с которой извлекаемые данные могут считаться частью сообщения и конец участка, содержащего скрытую информацию.

Этим проблемам было уделено немало времени при разработке алгоритма, поэтому им посвящены отдельные разделы (“4.2. Эффект dropped frames”, “4.3. Синхронизация”).

4.1. Структурная схема

Ниже представлена структурная схема алгоритма, в которой представлены основные его составляющие отдельно для режима встраивания и извлечения.

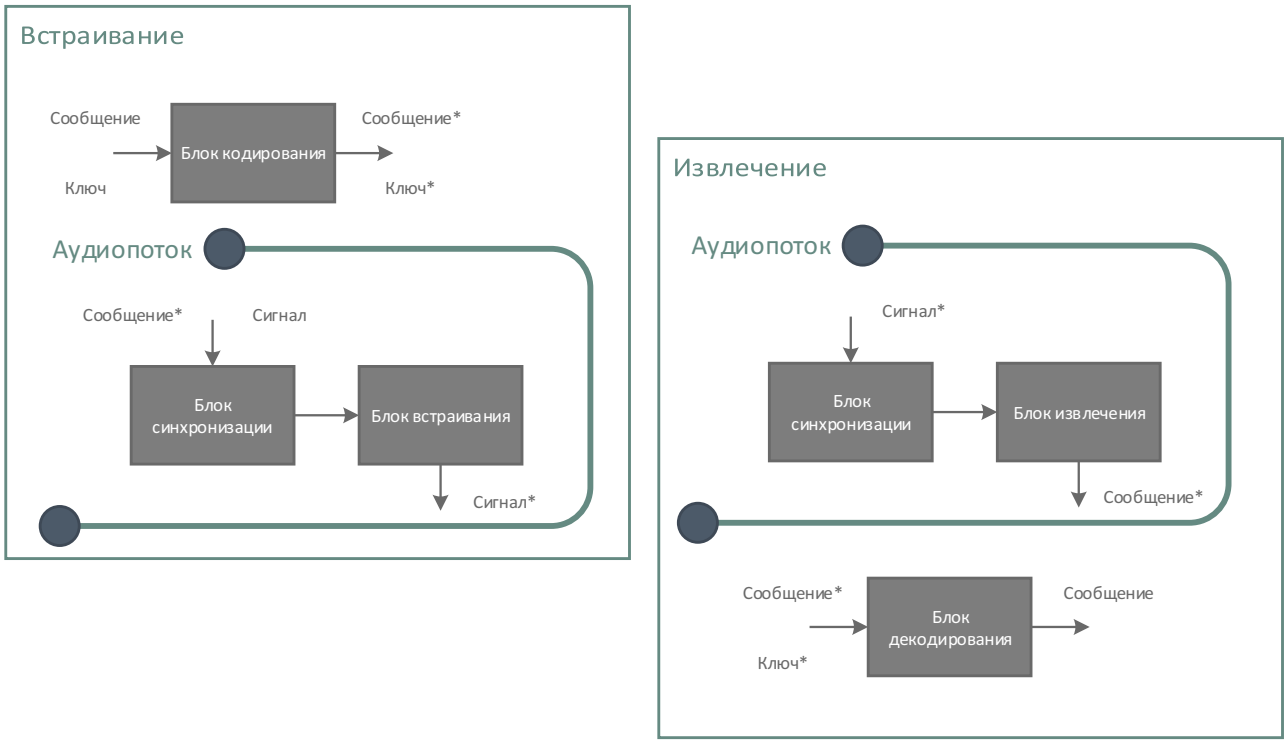

Рис. 4.1. Структурная схема алгоритма.

Алгоритм может находиться лишь в одном из двух состояний:

- Состояние, при котором происходит встраивание секретной информации в аудиосигнал;
- Состояние, при котором происходит извлечение секретной информации из аудиосигнала.

На схеме видно наличие блоков кодирования и декодирования. Как уже упоминалось ранее, именно совместное использование стеганографии и криптографии показывает наилучшие результаты. Так как процесс встраивания /извлечения должен происходить достаточно быстро для того, чтобы уложиться во время допустимой задержки, блоки кодирования и декодирования вынесены за пределы «потоковой» части алгоритма и выполняются до начала встраивания сообщения и после завершения извлечения.

Любой стеганографический алгоритм содержит параметры, которые должны быть известны принимающей стороне для извлечения сообщения. В данном случае каждый из блоков содержит параметр, необходимый для извлечения сообщения, но на схеме, для сохранения ясности, они содержатся внутри «Ключ*». Все они, действительно, известны уже на этапе кодирования и могут быть переданы получателю ещё до начала процесса встраивания по открытому каналу, ввиду их неприметности.

Блок синхронизации выполняется всего лишь раз в режиме встраивания, перед интеграцией сообщения, а в режиме извлечения до тех пор, пока не будет выполнена синхронизация (подробнее в ”4.3. Синхронизация”).

4.2. Эффект dropped frames

Пусть имеется некоторый буфер, например, 10 мс. Если процессор не справляется и не может обработать 10 мс до прихода следующего буфера, то значения теряются.

В качестве языка программирования для разработки был выбран Python, так как на нём можно очень быстро и удобно проверить какую-либо математическую идею, выполнить обработку данных и сигналов, построить графики. Тем не менее, критические секции данного алгоритма требовали очень высокой скорости, которой с помощью Python добиться не удалось. Возник эффект отброшенных кадров в ситуации, когда существуют и отправитель и получатель и сигнал в реальном времени скрывается с одной стороны и извлекается с другой. Несмотря на этот эффект, работоспособность алгоритма удалось подтвердить, применяя его к файлу, записанному раннее и пропускаемому через алгоритм поочередно (т.е. сначала в режиме скрытия, а затем в режиме извлечения). Поэтому основная часть алгоритма была переписана на языке C++, что позволило добиться ускорения в несколько десятков раз и избавиться от эффекта. Совместное же использование двух языков позволило акцентировать внимание именно на сам алгоритм и не затратить больше чем нужно времени на вспомогательные элементы.

4.3. Синхронизация

Под синхронизацией подразумевается определение начала и конца встраиваемого сообщения с целью его извлечения. В данном алгоритме в целях безопасности используется лишь маркировка начала, а длина считается частью ключа, который должен быть известен принимающей стороне. Причиной такого выбора стал механизм шифрования, применяемый в блоке кодирования (подробнее в «4.4. Помехоустойчивое кодирование»), который позволяет извлечь сообщение, только если его длина соответствует оригинальной. Это делает алгоритм более устойчивым к атакам.

Маркировка происходит таким же образом, как и скрытие сообщения, однако сам маркер должен быть известен при извлечении. Если алгоритм работает в режиме извлечения, происходит сканирование входящего потока на наличие маркера. Поскольку при извлечении сообщения аудиопоток может быть смещён и входящий буфер отличается от исходящего, необходимо использовать накопительный массив при сканировании. В качестве примера снова можно привести ситуацию с отправителем и принимающей стороной. Алгоритм-отправитель модифицирует аудиопоток, поступающий извне и перенаправляет его получателю, который начал сканирование заблаговременно. В таком случае порции сигнала у получателя отличны от тех, что формируются у отправителя. Данное взаимодействие представлено на рис. 4.2.

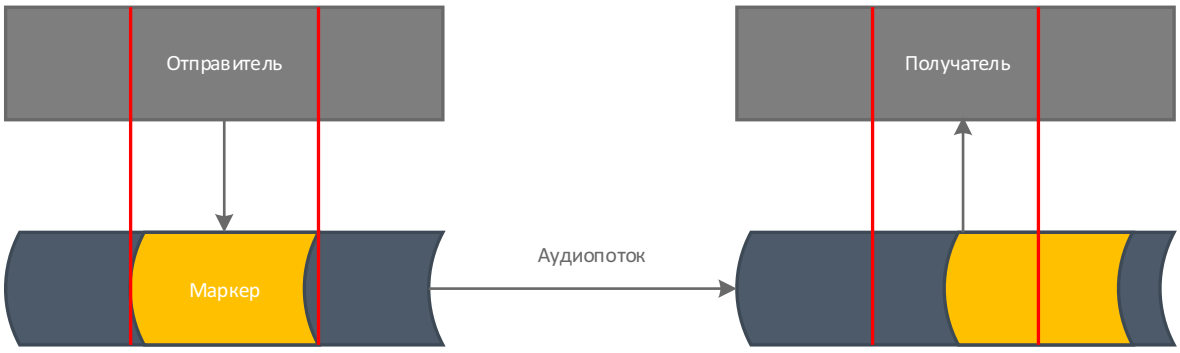

Рис. 4.2. Смещение секретной информации в аудиопотоке при извлечении.

Для того чтобы успешным образом извлечь сообщение достаточно сохранять дополнительно ещё хотя бы две порции аудиосигнала, последовательно поступающие на вход. Заранее зная значение маркера и, приведя его к нужному виду, его местоположение можно достаточно быстро

определить, не выполняя никаких операций декодирования, а просто осуществляя поиск внутри накопительного массива (рис. 4.3.).

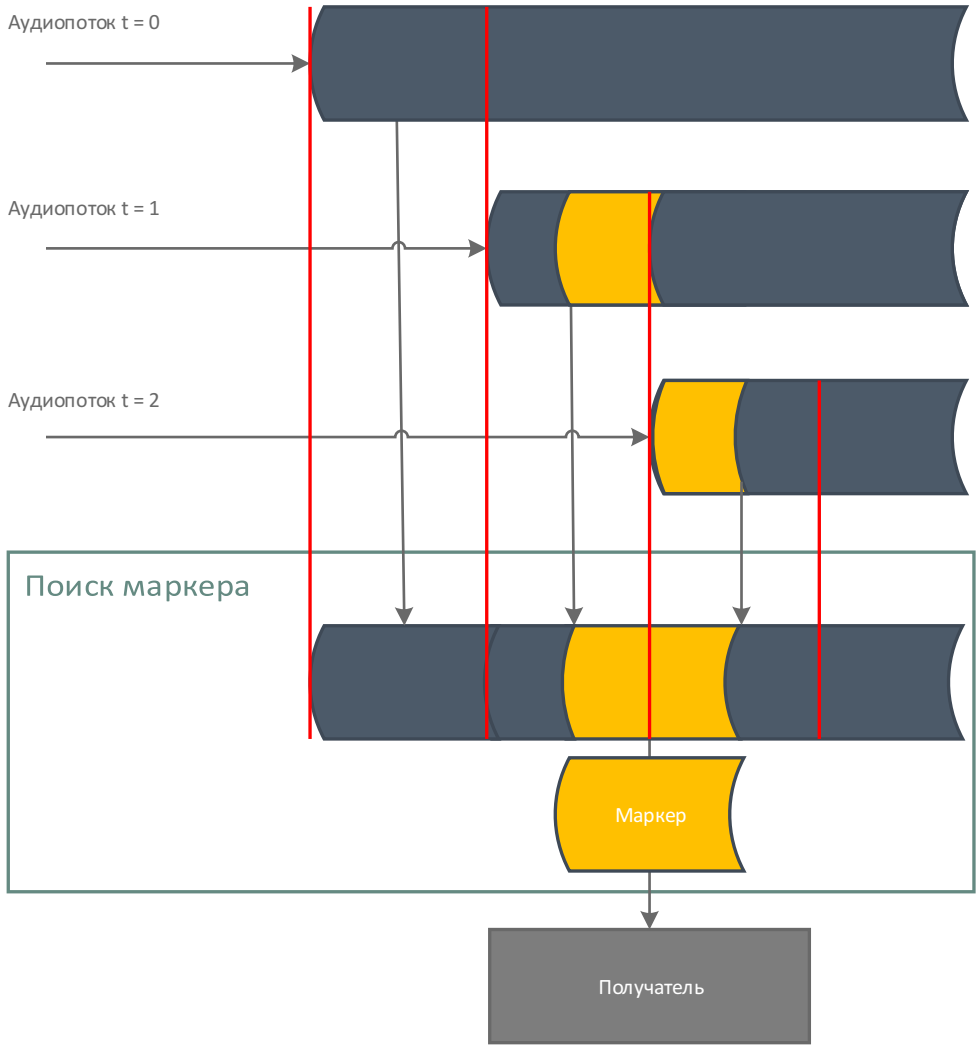

Рис. 4.3. Выявление маркера в аудиопотоке.

Поиск маркера должен быть быстрым, чтобы не превысить допустимую задержку. Длина «порции» звука (фрейма) может быть равна frame_size = 512, 1024, 2048 … p, где p ограничено частотой дискретизации аудиофайла и возможностями системы. Для накопительного буфера достаточно размера, превышающего frame_size в 3 раза. Допустимую скорость можно получить, реализовав поиск, схожий с поиском подстроки:

```
template <typename IntegerType>
bool contains(IntegerType* small, size_t size_small, IntegerType* big, size_t size_big, size_t& out_pos)
{
   bool matches = false;
   out_pos = NOT_FOUND;
   for (size_t i = 0; i < size_big-size_small+1; ++i) {
      matches = true;
      for (size_t j = 0; j < size_small; ++j) {
         if (big[i+j] != small[j]) {
            matches = false;
            break;
         }
      }
      if(matches) {
         out_pos = i;
         break;
      }
   }
   return matches;
}
```

Листинг 4.1. Реализация поиска маркера внутри «накопленного» звука.

Далее, после нахождения маркера, осуществляется накопление сигнала, несущего скрытую информацию. На этом этапе уже не требуется повышенный размер накопительного буфера, который превышал бы значение frame_size. Пусть frame_size = 1024, тогда каждые 1024 отсчёта выполняется извлечение секретной информации с помощью «Блока извлечения», указанного в структурной схеме на схеме 4.1. Подробнее об этой составляющей алгоритма описано в разделе “4.5. Скрытие / Извлечение”.

4.4. Помехоустойчивое кодирование

Одно из требований к стеганографической системе это шифрование скрываемой информации. Это позволяет достичь наилучших результатов, используя совместно две науки информационной безопасности. В данной работе будет применяться шифрование, основанное на обобщённых матрицах Фибоначчи, представленных Алексеем Павловичем Стаховым в 1996 г.

Далее следует краткое описание классическое Q-матрицы, называемой матрицей Фибоначчи.

Q-матрица – квадратная матрица следующего типа:

$$Q = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad (4.1)$$

Q-матрица обладает следующим важным свойством:

$$Q^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}, \quad (4.2)$$

где F_{n-1}, F_n, F_{n+1} - числа Фибоначчи.

Это означает, что при возведении в степень Q-матрицы, она обнаруживает связь с числами Фибоначчи. Иногда такую матрицу используют для поиска чисел Фибоначчи.

Определитель любой Q-матрицы равен:

$$\det\left(Q^n\right) = (-1)^n \qquad (4.3)$$

В 1996 г. Алексей Павлович Стахов обобщил понятие Q-матрицы и ввел понятие Q_p-матрицы (p = 0, 1, 2, 3, ...), которая включает в себя классическую Q-матрицу для случая p = 1. В основе Q_p-матрицы лежат расширенные p - числа Фибоначчи, которые представляют собой общий случай классической последовательности Фибоначчи и задаются, в том числе, и на отрицательной области.

Введенные Q_p-матрицы обладают рядом уникальных математических свойств:

- Если в Q_p-матрице вычеркнуть последний столбец и предпоследнюю строку, то мы получим матрицу Q_{p-1}. Это означает, что все Q_p - матрицы связаны между собой, то есть каждая Q_p-матрица содержит в себе все предыдущие и входит во все последующие.
- Следующим уникальным свойством Q_p-матриц есть значение ее детерминанта, которое равно: $\det Q_p = (-1)^p$, то есть детерминанты матриц Q_p равны 1 для всех четных p и (-1) для всех нечетных p.
- Основным результатом теории Q_p-матриц является выражение для n-й степени матрицы Q_p. Доказано, что все элементы такой матрицы являются соответствующими p-числами Фибоначчи [2]:

$$Q_p^n = \begin{bmatrix} F_p(n+1) & F_p(n) & \cdots & F_p(n-p+2) & F_p(n-p+1) \\ F_p(n-p+1) & F_p(n-p) & & F_p(n-2p+2) & F_p(n-2p+1) \\ \vdots & & \ddots & \vdots & \\ F_p(n-1) & F_p(n-2) & & F_p(n-p) & F_p(n-p-1) \\ F_p(n) & F_p(n-1) & \cdots & F_p(n-p+1) & F_p(n-p) \end{bmatrix} \quad (4.4)$$

- Кроме того доказано, что детерминант матрицы Q_p в степени n задается следующим выражением: $\det Q_p = (-1)^{pn}$.

- Кодирование информации:

$$\begin{cases} M * Q_p^n = E \\ E * Q_p^{-n} = M \end{cases} \quad (4.5)$$

- $detE = \det\left(M * Q_p^n\right) = detM * detQ_p^n = detM * (-1)^{p*n}$ (4.6)

Между элементами исходной матрицы $\boldsymbol{M}$ и кодовой матрицей $\boldsymbol{E}$ устанавливаются строгие математические соотношения, которые могут быть использованы для обнаружения и исправления ошибок. В формуле (4.6), вычисляя детерминант $\boldsymbol{M}$ и используя его в качестве «контрольного сообщения», обнаружение и исправление становится весьма эффективным.

Именно 5ое и 6ое свойства, а также компактность метода и его относительная редкость являются основополагающими для выбора алгоритма шифрования информации. Благодаря свойству восстановления при нарушении целостности последовательности стеганографическая система может претендовать на звание идеальной с точки зрения устойчивости к искажениям. Чтобы это действительно было так, необходимо в первую очередь обеспечить стабильную синхронизацию при наличии потерь и сильных искажений. Этот вопрос отнесён к категории дальнейших разработок.

Ниже представлена схема работы блока кодирования / декодирования:

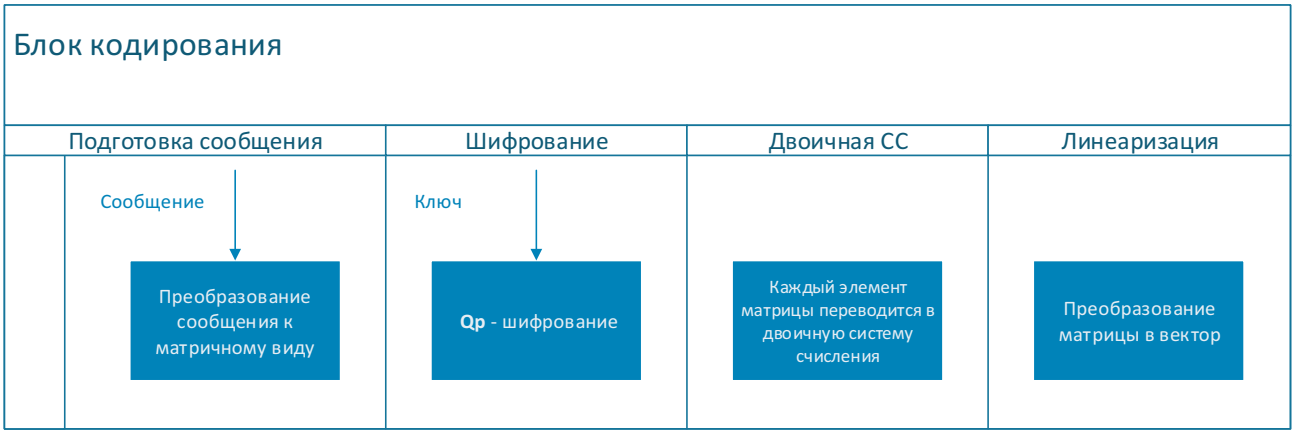

Рис. 4.4. Функциональная схема блока кодирования.

Сообщение представляет собой байтовый массив. Это может быть текстовая информация, медиафайл, исполняемый файл и т.д. На 1ом этапе он преобразуется к квадратной матрице, где каждое значение представлено в виде целого числа. При необходимости матрица дополняется нулями для достижения квадратного размера.

Далее выполняется шифрование, основанное на обобщённых матрицах Фибоначчи.

На третьем этапе каждый элемент полученной матрицы переводится в двоичную систему счисления.

Последний этап алгоритма преобразует матрицу к линейному виду. Размер этой последовательности сохраняется и должен быть известен при дешифровке.

Реализацию класса QpMatrix, с помощью которого осуществляется шифрование можно изучить в разделе «Приложение I», «Листинг П. 1.1».

Функциональная схема блока декодирования представлена на рис. 4.5.

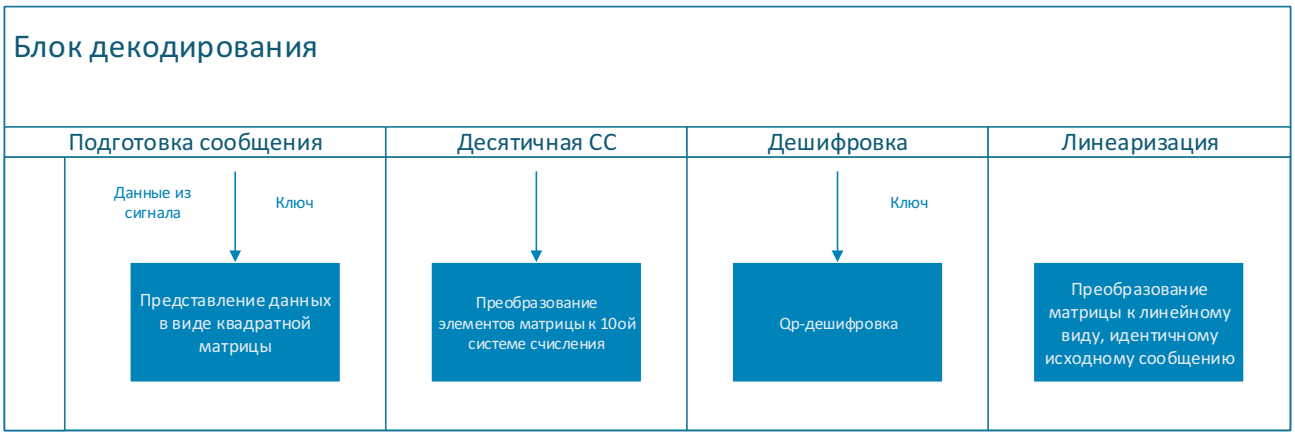

Рис. 4.5. Функциональная схема блока декодирования.

4.5. Скрытие / Извлечение

Существуют две области для скрытия информации в аудиосигнале: временная и частотная.

Для временной области можно выделить следующие базовые алгоритмы:

- Кодирование наименее значащих бит (НЗБ).
  Наименее значащие биты в исходном сигнале заменяются битами, которые соответствуют встраиваемому сообщению.
- Метод расширения спектра.
  Шифрование потока информации путём «рассеивания» кодированных данных по всему возможному частотному спектру. В этом случае приём сигнала возможен даже при наличии помех на определённых частотах.

Для частотной:

- Метод фазового кодирования.
  Замена фазы исходного звукового сегмента на опорную фазу, характер изменения которой отражает собой данные, которые необходимо скрыть.
- Скрытие данных с использованием эхо-сигнала.
  Встраивание данных в аудиосигнал-контейнер путём введения в него эхо-сигнала. Данные скрываются изменением трёх параметров эхо-сигнала: начальной амплитуды, скорости затухания и сдвига.

Оценка алгоритмов для потоковых аудио контейнеров				
Алгоритм	Вместительность	Уровень скрытности	Устойчивость	Применение в потоковых контейнерах
НЗБ	Выс.	Сред.	Низ.	Выс.
Фазовое кодирование	Низ.	Выс.	Выс.	Сред.
Расширение спектра	Выс.	Низ.	Выс.	Низ.
Эхо-сигнал	Выс.	Низ.	Сред.	Сред.

Таблица 4.1. Оценка алгоритмов для потоковых аудио контейнеров [4].

Стеганография реального времени добавляет требования к стеганосистеме, а именно: сложность системы, пропускная способность, полоса пропускания, задержка, отсутствие дублирования, время восстановления, время на подготовку системы.

В таблице 4.1. представлена оценка базовых алгоритмов для использования в потоковом режиме [4].

Из всех алгоритмов лучше всего удовлетворяет требованиям НЗБ метод. Основными недостатками этого метода являются: относительно низкая устойчивость к искажениям и к стеганодетекторам. Однако они скорее характерны для контейнеров фиксированного типа. В потоковых контейнерах сложнее определить наличие скрытого сообщения. Проблема искажений в первую очередь актуальна для интеграции синхронизирующих маркеров. Для реализации потокового алгоритма был выбран именно НЗБ метод, в качестве базового механизма встраивания.

Обыкновенное встраивание НЗБ-методом осуществляется следующим образом:

Пусть имеется порция сигнала **p**, подлежащая обработке.

p = {z_1, z_2, z_3, … }

$z_1 = 134_{10}$

$z_2 = 78_{10}$

$z_3 = 229_{10}$

...

Тогда в двоичной системе счисления эти числа представимы в виде:

$Z_1 = 0110\ 0001_2$

$Z_2 = 0111\ 0010_2$

$Z_3 = 1010\ 0111_2$

...

Выделенные биты заменяются на биты скрываемого сообщения:

m = {k_1, k_2, k_3…}, $k_1=1$, $k_2=1$, $k_3=0$.

Модифицированная порция сигнала равна:

$Z^*_1 = 1110\ 0001_2 = 135_{10}$

$Z^*_2 = 1111\ 0010_2 = 79_{10}$

$Z^*_3 = 0010\ 0111_2 = 228_{10}$

…

Ключевой вопрос заключается в том, с каким шагом осуществлять встраивание информации, ведь именно от этого зависит пропускная способность и уровень скрытности. В данной работе предлагается использовать сдвиговую функцию Альтера-Джонсона для вычисления шага, с которым будут встраиваться биты.

4.5.1. Сдвиговая функция Альтера-Джонсона

Основное применение функции Альтера-Джонсона – определение значений, наиболее близких к периодам, в данных, имеющих нелинейные колебания [3].

Для дискретного случая, если **n** - общее число отсчётов функции **f(t)**, заданной экспериментальными значениями, вводится следующая функция для определения почти периодов:

$$a(\tau) = \frac{1}{n-\tau} * \sum_{t=1}^{n-\tau} |f(t+\tau) - f(t)| \qquad (4.7)$$

Система почти - периодов τ функции **f(t)** может быть определена как совокупность локальных минимумов сдвиговой функции

$$\tau = argmin(a(\tau)),\ \tau_{min} \leq \tau \leq \tau_{max},$$

где τ_{min} и τ_{max} - естественные пределы поиска периода.

Полученный почти-период используется в качестве шага для осуществления встраивания сообщения. Такой подход позволяет всегда получать уникальную схему встраивания, что увеличивает устойчивость к стеганодетекторам. Пропускная способность сильно зависит от типа аудиоконтейнера и размера фрейма (порции звука, поступающей на обработку). Возникновение крайних, нежелательных случаев возможно, например, когда $\tau \cong 1$, но их можно искусственно изолировать. Во время проверки алгоритма, как правило, приходилось сталкиваться с обратной ситуацией. Чем выше значение τ, тем меньше пропускная способность, но выше уровень скрытности. Так как алгоритм ориентирован на потоковый контейнер, то предпочтение отдаётся именно безопасности. И хотя в случае с потоковым контейнером его длина заранее неизвестна и нельзя гарантировать, что сообщение полностью поместится в контейнере, тем не менее, возможно контролировать длину этого контейнера, если влиять на потоковый контейнер (например, самостоятельно осуществлять запись).

Операция определения шага с помощью сдвиговой функции происходит как при встраивании, так и при извлечении сообщения. Чтобы значение шага при извлечении совпадало с исходным значением, в алгоритме используются два аудиоканала. Например, левый для определения шага, а правый для скрытия данных. Левый канал при этом не претерпевает изменений. Схема 4.6 демонстрирует этот процесс.

Рис. 4.6. Схема работы блока скрытия / извлечения.

На рис. 4.7. представлена полная схема стеганографического алгоритма.

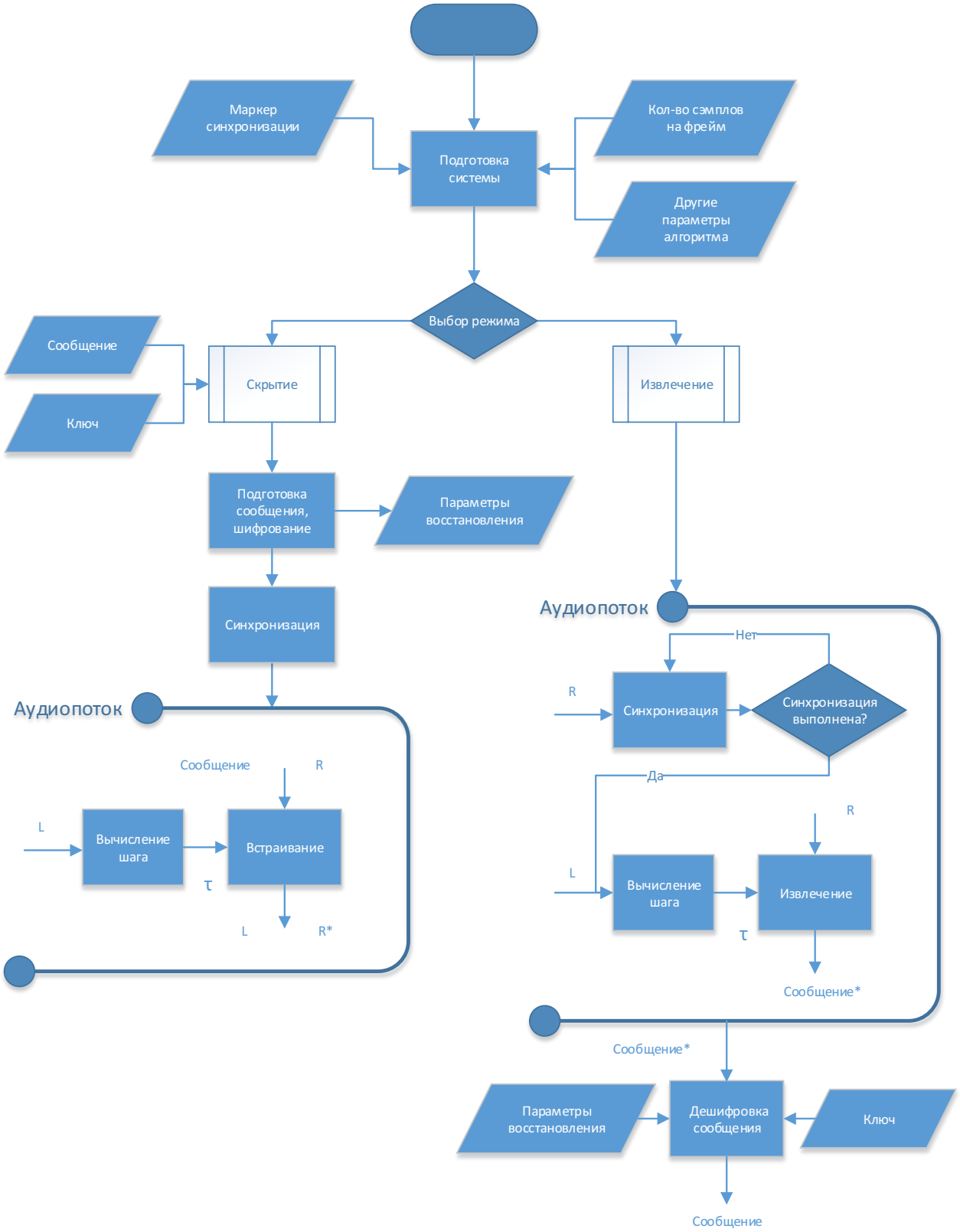

Рис. 4.7. Полная схема работы представленного стеганографического алгоритма.

4.6. Совместное использование с другими приложениями

Возможность совместно использовать программное обеспечение, использующее представленный алгоритм, с другими программами позволит существенно расширить область применения и удобство в использовании. Для этого необходимо установить расширение, для операционной системы, которая добавит к стандартным аудио-входу (например, линейный вход) и аудио-выходу (наушники, динамики) дополнительные виртуальные аудиоустройства. В любой аудиопрограмме можно выбрать входное и выходное устройство. Если требуется перенаправить звук с скрытым сообщением в другую программу, то в этой программе в качестве входа будет выбрано виртуальное аудиоустройство. Если звук принимается из другой программы в качестве контейнера для секретного сообщения, либо уже содержит скрытую информацию внутри и требует извлечения, то в этой программе виртуальное аудиоустройство устанавливается в качестве выхода. Например, для операционной системы OS X в качестве системного расширения можно использовать SoundFlower, которое добавит 2х-канальное и 64-канальное аудиоустройства.

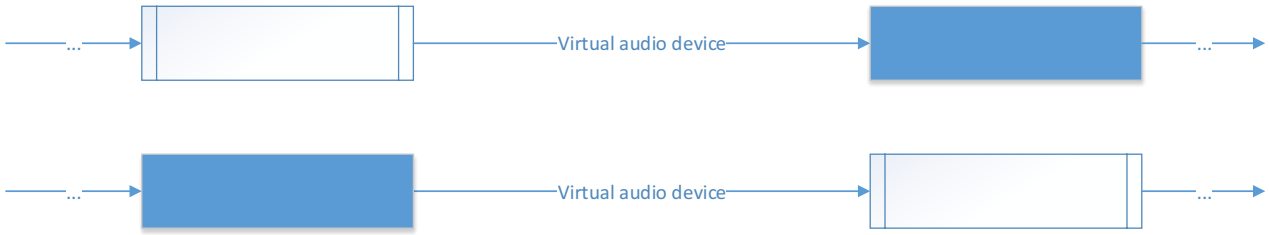

Рис. 4.8. Перенаправление аудиосигнала между программами.

5. Оценка качества системы

Для оценки качества полученной системы воспользуемся наиболее распространёнными показателями аудио искажения, которые основаны на анализе структуры контейнера. Специально для этих целей был разработан отдельный программный модуль.

Ниже, в подразделах, представлены две группы показателей: разностные и корреляционные. В каждой группе содержатся: наименование показателя; его формула; программная реализация на языке Python из разработанного модуля и результат анализа для указанных данных. Для оценки выбрано два типа контейнера: контейнер, содержащий речь и музыку. Для каждого из них применялся один из следующих двух режимов алгоритма: с упором на безопасность и с повышенным уклоном на ёмкость (таблица 3).

№	Содержимое контейнера	Длительность, сек.	Размер, Кб	Режим алгоритма
1	Речь	120	21 708.8	• безопасность
2	Музыка	9	537	• вместительность

Таблица 5.1. Характеристика экспериментальных данных.

В режиме повышенной вместительности, значение почти-периода (см. «4.5.1. Сдвиговая функция Альтера-Джонсона») применяется только для обозначения начала встраивания, а сам шаг равен 1. В режиме повышенной безопасности шаг равен почти-периоду.

В качестве скрываемой информации был выбран следующий текст:

«In the field of audio steganography, fundamental spread spectrum (SS) techniques attempts to distribute secret data throughout the frequency spectrum of the audio signal to the maximum possible level.» Для интеграции такого текста потребовалось 1600 бит. Сам процесс встраивания секретной информации можно охарактеризовать следующими величинами:

- Количество сэмплов в одном фрейме, **SPF**;
- Частота дискретизации аудиопотока, **SR**;
- Количество фреймов, в которых были спрятаны части информации, **FC**;
- Длительность той части аудиосигнала, которая содержит секретную информацию, **D**;
- Среднее количество бит, которое было спрятано в одном фрейме, **BPF**;

- Средняя скорость, с которой происходило встраивание информации, **BPS**.

№ контейнера	SPF	SR, Гц	FC	D, мс	BPF, бит/фрейм	BPS, бит/сек
1	1024	44100	1469	34110	2	50
2	1024	44100	10	232	360	8000

Таблица 5.2. Отчёт об интеграции сообщения в контейнеры (1) и (2).

Вычисление длительности D производилось по формуле:

$$D = FC * \left(\frac{SPF}{SR}\right) * 1000 \qquad (5.1)$$

5.1. Разностные показатели искажения

Разностные показатели искажения базируются на отличии между контейнером-оригиналом (неискажённый сигнал) ***C*** и контейнером-результатом (искажённый сигнал) ***S***.

Максимальная разность, **MD**

Формула	Программа
$MD = \max_n \lvert C_n - S_n \rvert$	def MD(original, modified): diff = np.absolute(np.subtract(original, modified)) return np.amax(diff)

Оригинал	Результат 1	Результат 2
0	1	1

Средняя абсолютная разность, **AD**

Формула	Программа
$AD = \frac{1}{N} * \sum_n \lvert C_n - S_n \rvert$	def NAD(original, modified): n = len(original) diff = np.absolute(np.subtract(original, modified)) return np.sum(diff) / float(n)

Оригинал	Результат 1	Результат 2
0	0.0005	0.02

Нормированная средняя абсолютная разность, **NAD**

Формула	Программа
$NAD = \sum_n \lvert C_n - S_n \rvert \Big/ \sum_n \lvert C_n \rvert$	def NAD(original, modified): diff = np.absolute(np.subtract(original, modified)) sum1 = np.sum(diff) sum2 = np.sum(np.absolute(original)) return sum1 / float(sum2)

Оригинал	Результат 1	Результат 2
0	$5.5 * 10^{-7}$	$4.1 * 10^{-6}$

Среднеквадратическая ошибка, **MSE**

Формула	Программа
$MSE = \frac{1}{N} * \sum_n (C_n - S_n)^2$	def MSE(original, modified): n = len(original) diff = np.subtract(original, modified) diff **= 2 return np.sum(diff) / float(n)

Оригинал	Результат 1	Результат 2
0	0.0005	0.02

Нормированная среднеквадратическая ошибка, **NMSE**

Формула	Программа

$NMSE = \sum_n (C_n - S_n)^2 \Big/ \sum_n (C_n)^2$	def NMSE(original, modified):
	  orig = np.array(original, dtype=np.int32)
	  orig **= 2
	  sum1 = np.sum(orig)
	  diff = np.subtract(original, modified)
	  diff **= 2
	  sum2 = np.sum(diff)
	  return sum2 / float(sum1)

Оригинал	Результат 1	Результат 2
0	$1.4 * 10^{-10}$	$5.6 * 10^{-10}$

L^p- норма

p = 2

Формула	Программа

$L^p = \left(\frac{1}{N} * \sum_n \lvert C_n - S_n \rvert^p\right)^{1/p}$	def LpNorm(original, modified, p):
	  n = len(original)
	  diff = np.absolute(np.subtract(original, modified))
	  diff = np.power(diff, p)
	  s = np.sum(diff)
	  return np.power(s / float(n), 1.0 / p)

Оригинал	Результат 1	Результат 2
0	0.02	0.15

Отношение «Сигнал/шум», **SNR**

Формула	Программа

$SNR = \sum_n (C_n)^2 \Big/ \sum_n (C_n - S_n)^2$	def SNR(original, modified):
	  orig = np.array(original, dtype=np.int32)
	  orig **= 2
	  sum1 = np.sum(orig)
	  diff = np.subtract(original, modified)
	  diff **= 2
	  sum2 = np.sum(diff)

Оригинал	Результат 1	Результат 2
∞	$7.0 * 10^9$	$1.8 * 10^9$

Максимальное отношение «сигнал/шум», **PSNR**

Формула	Программа

$PSNR = N * \frac{max(C_n)^2}{n} \Big/ \sum_n (C_n - S_n)^2$	def PSNR(original, modified):
	  n = len(original)
	  orig = np.array(original, dtype=np.int32)
	  orig **= 2
	  m = np.amax(orig)
	  diff = np.subtract(original, modified)
	  diff **= 2
	  sum2 = np.sum(diff)
	  return n * m / sum2

Оригинал	Результат 1	Результат 2
∞	$2 * 10^{12}$	$4.9 * 10^{10}$

Качество звучания, **AF**

Формула	Программа

$AF = 1 - \sum_n (C_n - S_n)^2 \Big/ \sum_n (C_n)^2$	def AF(original, modified):
	  orig = np.array(original, dtype=np.int32)
	  orig **= 2
	  sum1 = np.sum(orig)
	  diff = np.subtract(original, modified)
	  diff **= 2
	  sum2 = np.sum(diff)
	  return 1 - sum2 / float(sum1)

Оригинал	Результат 1	Результат 2
1	≈ 1	≈ 1

5.2. Корреляционные показатели искажения

К этой группе относятся показатели, основанные на корреляции между оригинальным и искажённым сигналами.

Нормированная взаимная корреляция, **NC**

Формула	Программа
$NC = \sum_n (C_n * S_n) \Big/ \sum_n (C_n)^2$	def NC(original, modified): multiply = np.multiply(original, modified) sum1 = np.sum(multiply) orig = np.array(original, dtype=np.int32) orig **= 2 sum2 = np.sum(orig) return sum1 / float(sum2)

Оригинал 1	Результат 1	Оригинал 2	Результат 2
0.0004	0.0004	$1.15 * 10^{-5}$	$1.16 * 10^{-5}$

Качество корреляции, **CQ**

Формула	Программа
$CQ = \sum_n (C_n * S_n) \Big/ \sum_n (C_n)$	def CQ(original, modified): multiply = np.multiply(original, modified) sum1 = np.sum(multiply) orig = np.array(original, dtype=np.int32) sum2 = np.sum(orig) return sum1 / float(sum2)

Оригинал 1	Результат 1	Оригинал 2	Результат 2
255409	255410	-5.15	-5.19

Структурное содержание, **SC**

Формула	Программа
$SC = \sum_n (C_n)^2 \Big/ \sum_n (S_n)^2$	def SC(original, modified): orig = np.array(original, dtype=np.int32) orig **= 2 sum1 = np.sum(orig) mod = np.array(modified, dtype=np.int32) mod **= 2 sum2 = np.sum(mod) return sum1 / float(sum2)

Оригинал	Результат 1	Результат 2
1	≈ 1	≈ 1

5.3. Подведение итога

№	Название показателя	Оригинал	Результат 1	Результат 2
1	Максимальная разность	0	1	1
2	Средняя абсолютная разность	0	0.0005	0.02
3	Нормированная средняя абсолютная разность	0	$5.5 * 10^{-7}$	$4.1 * 10^{-6}$
4	Среднеквадратическая ошибка	0	0.0005	0.02
5	Нормированная среднеквадратическая ошибка	0	$1.4 * 10^{-10}$	$5.6 * 10^{-10}$
6	L^p- норма, p = 2	0	0.02	0.15
7	Отношение «сигнал/шум»	∞	$7.0 * 10^{9}$	$1.8 * 10^{9}$
8	Максимальное отношение «сигнал/шум»	∞	$2 * 10^{12}$	$4.9 * 10^{10}$
9	Качество звучания	1	≈ 1	≈ 1
10	Нормированная взаимная корреляция	0.0004 $1.15 * 10^{-5}$	0.0004	$1.16 * 10^{-5}$
11	Качество корреляции	255409 -5.15	255410	-5.19
12	Структурное содержание	1	≈ 1	≈ 1

Таблица 5.3. Итоговая таблица показателей.

Результаты для обоих случаев практически не отличаются от оригинала, что доказывает работоспособность системы и допустимый уровень качества.

6. Руководство пользователя

Результат работы представлен в виде программы с интерфейсом командной строки.

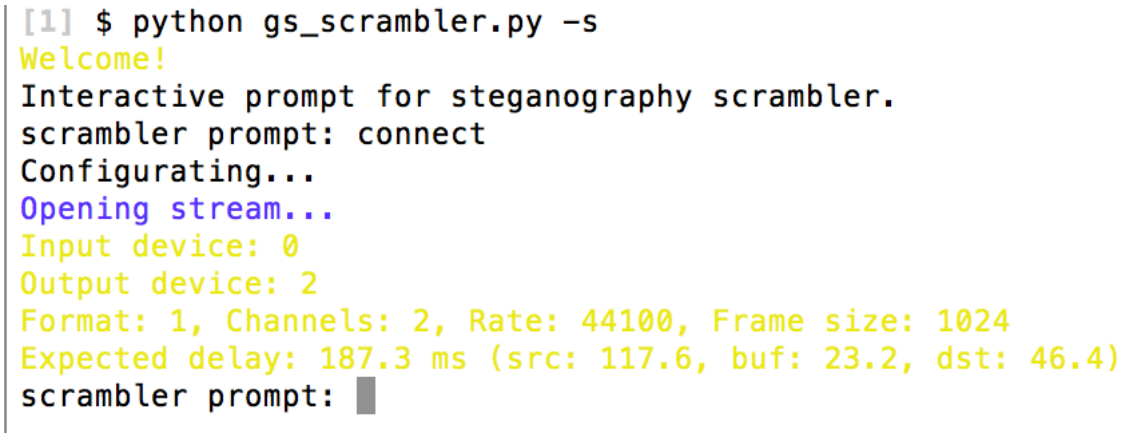

Рис. 6.1. Вид приложения

Имеется встроенная документация, доступ к которой можно получить, введя команду *help*. Программа позволяет выбирать в качестве аудиовхода/выхода не только встроенные и виртуальные аудиоустройства, но и файлы формата wave. С помощью инструкций *hide/recover* выполняется скрытие/извлечение. С описанием каждой инструкции и её параметрами можно ознакомиться, введя в командной строке *help hide/help recover*.

Имеются следующие ограничения:

- Если одновременно существуют отправитель и получатель, то канал связи между ними не должен прерываться для успешного выполнения синхронизации;
- Аудиоконтейнер должен иметь 2 канала.

Заключение

В результате проделанной работы предложен алгоритм, удовлетворяющий основным требованиям цифровой стеганографии и выполняющий решение поставленной задачи в режиме реального времени, а именно выполняет скрытие информации внутри аудиосигнала в потоковом режиме. Такой подход позволяет не только контролировать длину и содержимое аудиоконтейнера для хранения конфиденциальной информации, но и осуществлять потоковую передачу секретных данных. Помимо скрытой передачи/хранения данных, возможна защита авторских прав аудиозаписей, что позволяет обезопасить интеллектуальную собственность от незаконного копирования, а совместное использование с другими аудиопрограммами расширяет область применения.

Оценка качества алгоритма, которая была проведена, опираясь на разностные и корреляционные показатели, доказала работоспособность системы.

Перечень использованной литературы

1. Г.Ф. Конахович А.Ю. Пузыренко, Компьютерная стеганография, теория и практика, МК-Пресс, Киев, 2006
2. А.П. Стахов, Компьютеры Фибоначи и новая теория кодирования: история, теория, перспективы.
3. В.И.Кузьмин, А.Ф. Гадзаов, Методы построения моделей по эмпирическим данным, МИРЭА , Москва, 2012 — 96 стр.
4. A Survey on Steganography Techniques in Real Time Audio Signals and Evaluation, International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012
5. Выполнение организационно-экономической части дипломного проекта, МИРЭА, Москва, 2007 — 20 стр.