УДК 004.02

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ СТЕГАНОГРАФИИ В АУДИОФАЙЛАХ

Войкель В.И., студент гр. 281078

Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь

Скудняков Ю.А. – канд. техн. наук, доцент, доцент каф. ИСиТ

Аннотация. В статье рассмотрены вопросы разработки программного средства для скрытного внедрения в аудиофайлы дополнительной информации, представлен пользовательский интерфейс приложения, обоснован выбор средств разработки, перечислены недостатки аналогичных разработок, представлены основные функциональные возможности программного средства. В качестве языка программирования выбран язык C#.

Ключевые слова. Программное обеспечение, разработка приложений, стеганография, криптография, C#.

Введение.

Появление современных методов обработки данных привело к активному развитию технологий скрытой передачи информации. Стеганография, как один из таких методов, позволяет скрывать данные в различных цифровых медиаформатах, таких как изображения, аудиофайлы, видео. В настоящее время стеганография находит применение в системах защиты данных, а также в областях, где необходимо скрытое или защищенное хранение и передача информации. Одним из востребованных направлений является использование аудиофайлов для сокрытия данных, что обусловлено широким применением аудиоконтента и возможностью скрытия информации в аудиосигнале без заметного ухудшения его качества [1].

Стеганография – это наука, посвященная скрытому внесению информации в различные носители так, чтобы она оставалась незамеченной для третьих лиц. В отличие от криптографии, которая направлена на защиту данных путем их зашифровки, стеганография нацелена на скрытие самого факта существования передаваемой информации, что делает этот метод особенно ценным в ситуациях, когда требуется передавать секретные данные, не привлекая к ним лишнего внимания. Интересно отметить, что стеганография как метод скрытия информации имеет древние корни. Первые упоминания о скрытой передаче сообщений можно найти в трудах Геродота, описавшего использование восковых табличек для сокрытия секретных посланий. В Древнем Китае аналогичным образом использовались шелковые ленты с текстом, которые вшивали в одежду. Эти исторические примеры подчеркивают древность и постоянную актуальность необходимости скрытия информации, существовавшую задолго до разработки современных технологий.

В условиях современного информационного общества, где конфиденциальность данных приобретает все большее значение, стеганография находит свое применение во множестве областей, включая правоохранительные органы, журналистику и защиту авторских прав. Растущая угроза нарушения конфиденциальности данных подчеркивает важность изучения методов скрытой передачи информации. На сегодняшний день с развитием цифровых технологий и увеличением вычислительных мощностей стеганография в аудиофайлах становится особенно актуальной. Аудиофайлы представляют собой удобный носитель для скрытой передачи информации, так как они могут быть редактируемыми без значительных изменений в качестве звука, которое может быть воспринято человеком.

Среди форматов цифровых аудиофайлов WAV занимает особое место благодаря своей простоте, высокой точности и возможности хранения данных без сжатия, что делает его идеальным кандидатом для внедрения скрытых данных. Разработка программного средства, использующего методы стеганографии для внедрения и извлечения скрытых данных в аудиофайлах, представляет собой актуальную задачу. При этом программное средство должно обладать высокой степенью маскировки информации, быть легким в использовании и надежным в процессе извлечения данных.

Разработка программного средства, использующего стеганографию для внедрения и извлечения скрытых данных в аудиофайлах, является актуальной задачей. При этом важно, чтобы такое средство обладало высокой степенью маскировки информации, легкостью использования и надежностью извлечения данных.

Существуют различные программные продукты для стеганографии, однако многие из них имеют недостатки. Основные проблемы заключаются в сложности настройки, отсутствии интуитивного графического интерфейса пользователя [2].

Целью данного проекта является создание программного обеспечения для стеганографии в аудиофайлах, которое будет устранять недостатки существующих решений. Данное приложение позволит пользователям без труда внедрять и извлекать скрытые данные из аудиофайлов, используя понятный и удобный графический интерфейс.

Для реализации проекта необходимо будет решить ряд задач, в том числе провести анализ предметной области, исследовать существующие аналогичные программные продукты и определить функциональные требования. Также потребуется спроектировать интерфейс пользователя, разработать алгоритмы работы программного обеспечения и приступить к его программной реализации, а затем провести тестирование и отладку созданного приложения.

В качестве языка разработки был выбран C#, так как он является современным языком с поддержкой объектно-ориентированного программирования и наличием большого числа возможностей, включающих обработку исключений, сборщик мусора, асинхронность, кроссплатформенность [3].

Для построения пользовательского интерфейса будет использован фреймворкWinForms, который позволяет легко и быстро проектировать простые интерфейсы [4].

Практическая значимость результатов разработки заключается в возможности применения ПС в области стеганографии.

Основная часть.

Определим назначение и возможности ПС:

– осуществление функции загрузки файла в формате wav, в который будет производиться инъекция информации;

– возможность выбора текстового файла, информация из которого будет внедрена в целевой файл;

– проведение предварительного шифрования информации перед ее внедрением, а также осуществление записи информации не последовательно, а в случайном порядке.

При запуске программы пользователь может как внедрить данные в файл, так и извлечь их из файла, в котором они находятся.

Для внедрения данных пользователю необходимо выбрать файл WAV формата, в который необходимо внедрить данные. Затем необходимо выбрать текстовый файл, данные из которого будут интегрированы в целевой файл. После этого необходимо задать ключ, которым будут зашифрованы данные, а также указать паттерн, в соответствии с которым будет определен порядок изменения битов в целевом файле.

Интерфейс главного окна представлен на рисунке 1.

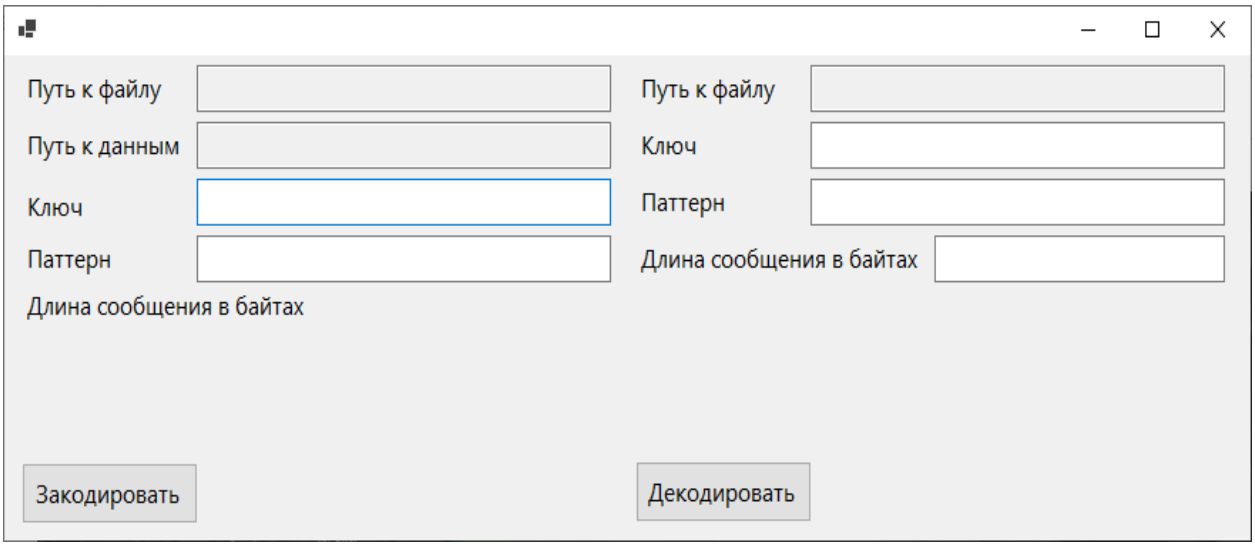

Рисунок 1 – Интерфейс главного окна

Рассмотрим предложенный концепт более подробно:

– в левой части окна можно указать целевой файл, файл с данными, ключ и паттерн, провести интеграцию в целевой файл;

– в правой части окна необходимо указать файл, из которого будут извлекаться данные, ключ и паттерн.

При выборе файла с данными рассчитывается их размер в байтах, если они не поместятся в целевой файл, то пользователь получит сообщение об этом.

При извлечении данных необходимо знать ключ, паттерн, длину сообщения, которое было интегрировано в файл.

В процессе шифрования данных применятся симметричный алгоритм шифрования AES – симметричный алгоритм шифрования, которой является надежным и хорошо изученным на криптостойкость, что делает его отличным кандидатом для использования.

На множестве современных процессоров реализованы команды для поддержки алгоритма AES на аппаратном уровне, что ускоряет процесс шифрования, следовательно, можно относительно быстро шифровать большие объемы данных в сравнении с другим подобными алгоритмами.

Методы стеганографии.

В данном программном средстве применяется метод стеганографии LSB. Этот метод заключается в выделении наименее значимых бит контейнера и последующей их замене на биты сообщения.

Поскольку для замены выбираются только наименее значимые биты, разница между исходным и измененным контейнерами оказывается незначительной и, как правило, незаметна для человеческого слуха.

Однако данный метод подходит исключительно для звуковых файлов в форматах без сжатия или со сжатием без потерь, так как наименее значимые биты значений сэмплов могут быть утеряны при сжатии с потерями.

Также рассмотрим другие методы стеганографии помимо LSB.

Методы частотного преобразования.

В этих методах скрытые данные встраиваются в частотные компоненты сигнала с использованием дискретного косинусного преобразования, вейвлет-преобразования или преобразования Фурье. Это делает их более устойчивыми к изменениям, таким как сжатие или фильтрация. К преимуществам можно отнести высокую стойкость к обработке файлов, возможность сокрытия больших объемов данных. К недостаткам можно отнести сложность реализации, снижение качества контейнера при некорректном внедрении.

Метод фазового кодирования. В данном методе изменяется фаза аудиосигнала таким образом, чтобы передать скрытую информацию. Восстановление данных возможно только при наличии оригинального сигнала. К преимуществам можно отнести высокийуровень скрытности, отсутствие заметных изменений в амплитуде аудиосигнала. К недостаткам можно отнести сложность реализации, зависимость от фазы оригинального сигнала.

Кратко опишем возможности метода эхо-модуляции. Данный метод использует искусственно добавленные слабые эхосигналы, которые содержат закодированную информацию. Эти эхосигналы не воспринимаются слухом, но могут быть извлечены специальными алгоритмами. К преимуществам можно отнести высокую стойкость к сжатию и шумовым атакам. К недостаткам можно отнести возможность появления заметных акустических эффектов, ограниченный объем внедряемых данных.

Метод LSB наиболее применим для WAV-файлов, так как в них сохраняются все биты звуковых данных без потерь. Однако при обработке таких файлов следует избегать постобработки, например сжатия.

Методы частотного преобразования, такие как DCT и DWT, также могут применяться в WAV-файлах, поскольку аудиоданные можно анализировать и изменять в частотной области, но это требует сложных алгоритмов.

Фазовое кодирование возможно для WAV, так как данный формат позволяет сохранять точную фазу аудиосигнала, но метод требует высоких вычислительных ресурсов.

Метод эхо-модуляции подходит для WAV-файлов, так как изменения фазы и добавление слабых эхосигналов остаются незаметными для слуха.

С целью увеличения сложности выявления факта внедрения дополнительной информации в файл используется не последовательная запись данных, а запись в случайном порядке, порядок которой задается на основе паттерна в виде числа, вводимого пользователем. На основе него формируется случайная последовательность, которая распределит данные по всему свободному месту в контейнере случайным образом.

Формат WAV – это формат без сжатия, что приводит к тому, что данные, хранимые в таком формате, занимают ощутимо больше места, что предоставляет возможность встраивания большого количества данных даже в относительно короткие по длительности аудиофайлы, хранимые в данном формате.

В таблице 1 представлено более подробное описание формата WAV.

Пример осуществления кодирования последовательности данных представлен на рисунке 2. На рисунке 2 был введен ключ, паттерн для кодирования информации в файл, далее проведено его декодирование. В случае декодирования необходимо также указать длину сообщения, которое было внедрено в файл.

В случае, если при попытке декодирования будет совершена ошибка при вводе хотя бы одного из параметров, декодировать данные не выйдет.

Это объясняется следующим:

– ввод неверного ключа приведет к неверному дешифрованию последовательности;

– ввод неверного паттерна приведет к выбору зашифрованного сообщения в неверном порядке, что также приведет к невозможности извлечения данных;

– ввод длины сообщения в байтах необходим, по причине того, что конец внедряемой последовательности никак не обозначается при внедрении данных, неверное значение также приведет к ошибке извлечения данных.

Таблица 1 – Описание WAV-формата

Поле	Размер в байтах	Смещение	Описание
Chunk ID	4	0	Этот фрагмент содержит буквы «RIFF» в кодировке ASCII.
ChunkSize	4	4	Это размер оставшейся части фрагмента, следующего за этим номером – размер всего файла в байтах минус 8 байт (ChunkID, ChunkSize).
Format	4	8	Этот фрагмент содержит символы «WAVE».
Sub-Chunk1 ID	4	12	Этот фрагмент содержит символы «fmt».
Sub-Chunk1 Size	4	16	16 для PCM
AudioFormat	2	20	Используется для указания типа кодировки–1для PCM, 2 для ADPCM, 3 для IEEEfloatingpoint, 7 для u-law, и 67734 для WaveFormatExtensible.
NumChannels	2	22	1 для моно, 2 для стерео.
SampleRate	4	24	8000, 16000, 44100 и т. д. Гц. Типичное значение – 44100, что соответствует аудио CD. Значение для телефонной и беспроводной микрофонной передачи, адекватное для человеческой речи – 8000.Значение 16000 используется в большинстве современных коммуникационных устройствах, которые работают на основе VoIP и VVoIP.
ByteRate	4	28	Это значение равняется Sample Rate * Num of Channel * Bytes per Sample.
BlockAlign	2	32	Это значение равняется Num of Channel * Bytes per Sample.
BitsPerSample	2	34	Количество бит на семпл. 8 для 8 бит (1 байт), 16 для 16 бит (2 байта) ит. д.
Sub-Chunk2 ID	4	36	Этот фрагмент содержит символы «data».
Sub-Chunk2 Size	4	40	Это значение равняется Num of Samples * Num of Channels * Bytes Per Sample
Data	*	44	Сами звуковые данные.

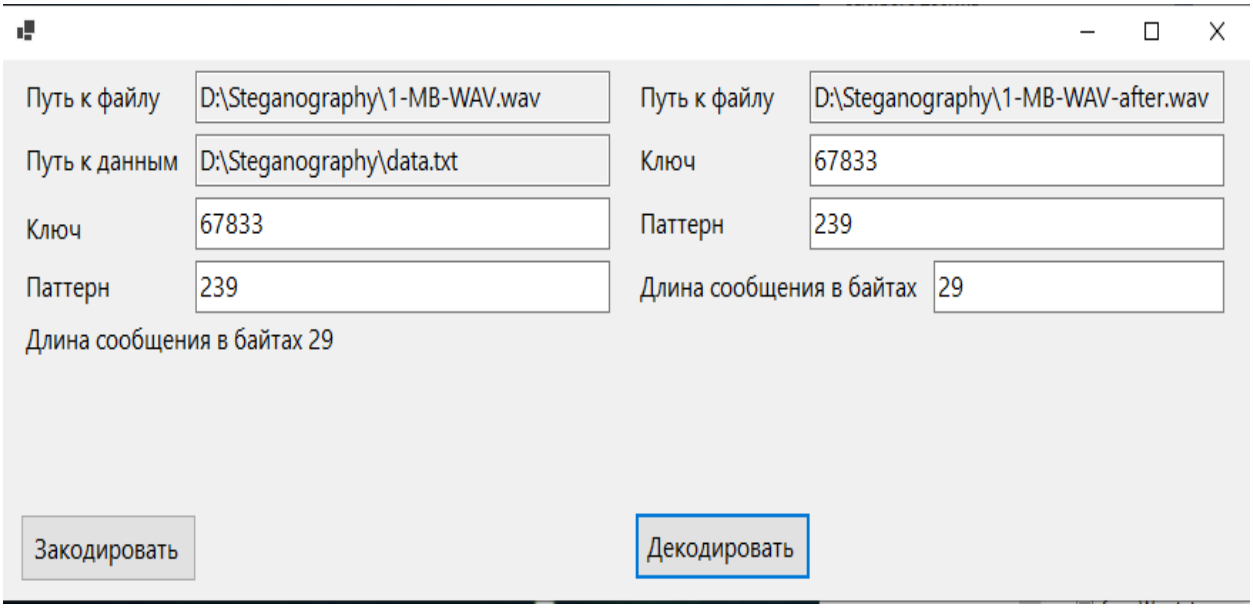

Рисунок 2 – Кодирование последовательности данных

После внедрения данных размер файла не изменится, причиной этого является изменение уже имеющихся данных, а не добавление новых или удаление уже имеющихся.

С целью проверки совпадения файлов после проведения внедрения данных можно рассчитать значение контрольной суммы для исходного и итогового файлов, рассчитанные значения не должны совпасть, результат представлен на рисунке 3.

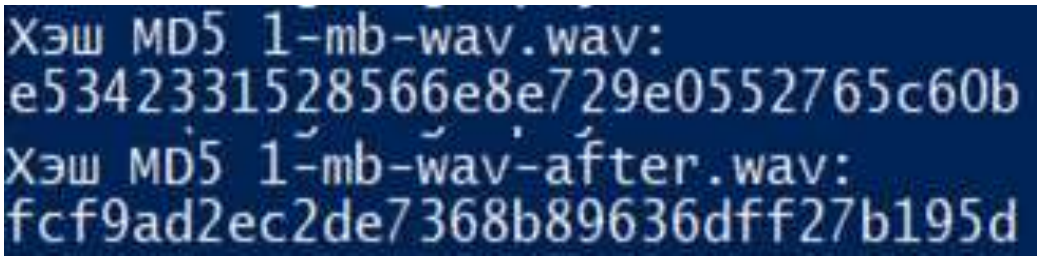

Рисунок 3 – Значение хеш-функции для файлов

Стеганография в аудиофайлах имеет широкие возможности применения, она может быть использована:

– в качестве средства для скрытой передачи информации, в случае ее фильтрации или недопущения передачи в явном виде;

– внедрение дополнительных данных в какие-либо файлы, с целью длительного хранения, например, в музыкальные произведения, которые будут записаны на диск;

– с целью защиты авторских прав путем внедрения скрытых водяных знаков или меток;

– в качестве подтверждения авторства и подлинности аудиоматериала.

Заключение.

Разработанное приложение позволяет быстро и удобно закодировать информацию в файл, а также извлечь ее из уже имеющегося файла.

В качестве возможных улучшений разработанного программного средства можно реализовать работу с другими алгоритмами шифрования, а также с другими алгоритмами стеганографии в аудиофайлах.

Список использованных источников:

1. Стеганография в XXI веке. Цели. Практическое применение. Актуальность [Электронный ресурс]. — Режим доступа: https://habr.com/ru/articles/253045. — Дата обращения: 1 апреля 2025 г.

2. 21 Best Steganography Tools To Keep Your Data Safe [Электронный ресурс]. — Режим доступа: https://technicalustad.com/steganography-tools. — Дата обращения: 1 апреля 2025 г.

3. Прайс, М. C# 10 и .NET 6. Современная кроссплатформенная разработка : учеб. пособие / М. Прайс. — Санкт-Петербург : Питер, 2023. — 848 с.

4. Либерти, Д. Программирование на C# : учеб. пособие / Д. Либерти. — Санкт-Петербург : Символ-Плюс, 2003. — 688 с.

UDC 004.02

SOFTWARE TOOL FOR STEGANOGRAPHY IN AUDIO FILES

Voykel V.I., student

Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. The article discusses the development of a software tool for covertly introducing additional information into audio files, presents the user interface of the application, substantiates the choice of development tools, lists the disadvantages of similar developments, and presents the main functional capabilities of the software tool. The C# was chosen as the programming language.

Keywords. Software, Application Development, Steganography, Cryptography, C#.