

ВСТРАИВАНИЕ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В АУДИОФАЙЛЫ ФОРМАТА MP3 ДЛЯ ПРОТИВОДЕЙСТВИЯ РАСПРОСТРАНЕНИЯ КОНТРАФАКТНОЙ ПРОДУКЦИИ

С.А. Серебрякова

sofyasofya98@gmail.com

SPIN-код: 9808-2216

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрены методы сокрытия информации в аудиофайлах, предложено описание алгоритма встраивания цифровых водяных знаков в MP3-файлы. Рассмотрены структура MP3-файла и структура фрейма. Разработан программный комплекс для встраивания и извлечения цифровых водяных знаков применительно к MP3-файлам. Данный алгоритм обладает повышенной устойчивостью к попыткам обнаружения скрытой информации. В статье также представлено, каким образом обходятся способы обнаружения наличия скрытой информации. Актуальность статьи заключается в защите авторского права на музыкальные и кинематографические произведения, а также противодействию распространения контрафактной продукции.

Ключевые слова

Стеганография, цифровая стеганография, сокрытие информации, защита информации, MP3-файл, алгоритм, код, фрейм

Поступила в редакцию 04.05.2020

© МГТУ им. Н.Э. Баумана, 2020

Введение. В настоящее время широкое распространение получили так называемые пиратские аудио- и видеодиски. Использование подобной продукции нарушает авторские права на музыкальные и кинематографические произведения. Поэтому, несомненно, актуальным является противодействие распространению контрафактных дисков.

Один из способов решения данной проблемой заключается в использовании цифровых водяных знаков (ЦВЗ), подтверждающих авторство произведения. Цифровой водяной знак — это некоторое шифрованное сообщение, наличие которого позволяет отличить оригинал произведения от его подделки. При этом сам аудио- или видеофайл со встроенным в него сообщением внешне не отличается от файла без ЦВЗ. Совокупность методов незаметного сокрытия одних битовых последовательностей в других получила название цифровой стеганографии [1, 2]. Важным преимуществом стеганографии по сравнению с криптографическими методами является сокрытие самого факта встраивания кодового сообщения.

В работе рассмотрен вопрос сокрытия информации в аудиофайлах. В настоящее время наибольшую известность имеют аудиофайлы двух типов — WAV и MP3. Однако на практике для хранения аудиоданных используются в основном файлы типа MP3, которые получены в результате сжатия информации и

поэтому занимают существенно меньше памяти. Поэтому в статье представлен алгоритм сокрытия и извлечения информации именно для MP3-файлов.

Структура MP3-файла. MP3-файлы содержат различные аудиоданные — звуки, звуковые эффекты, музыку, а также записи голоса [3]. Структура файла изменялась по мере их совершенствования. В настоящее время можно выделить следующие основные компоненты: ID3-тег и последовательность фреймов [4].

ID3-тег, как правило, располагается в начале файла. Он содержит различную справочную информацию. Например, для музыкальных дисков ID3-тег включает название песни, название альбома, исполнителя и некоторые другие данные. На рис. 1 представлен фрагмент тега, полученный с помощью программы WINHEX [5]. Слева указан адрес каждой строки текста в байтах, в середине — байты данных в шестнадцатеричном коде, а справа — расшифровка этих данных.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	49	44	33	03	00	00	00	00	1F	76	54	49	54	32	00	00	ID3 vTIT2
00000016	00	0C	00	00	00	4D	61	63	72	6F	76	69	73	69	6F	6E	Macrovision
00000032	54	59	45	52	00	00	00	05	00	00	00	32	30	30	35	54	TYER 2005T
00000048	50	55	42	00	00	00	08	00	00	00	52	65	70	72	69	73	PUB Repris
00000064	65	54	43	4F	4E	00	00	00	05	00	00	00	28	35	32	29	eTCOM (52)
00000080	54	41	4C	42	00	00	00	12	00	00	00	50	6C	61	79	69	TALB Playi
00000096	6E	67	20	54	68	65	20	41	6E	67	65	6C	54	52	43	4B	ng The AngelTRCK
00000112	00	00	00	02	00	00	00	36	50	52	49	56	00	00	00	27	6PRIV '
00000128	00	00	57	4D	2F	4D	65	64	69	61	43	6C	61	73	73	50	WM/MediaClassP
00000144	72	69	6D	61	72	79	49	44	00	BC	7D	60	D1	23	E3	E2	rimaryID 4) `Ñ#ãã
00000160	4B	86	A1	48	A4	2A	28	44	1E	50	52	49	56	00	00	00	Kt;H#*(D PRIV
00000176	29	00	00	57	4D	2F	4D	65	64	69	61	43	6C	61	73	73) WM/MediaClass
00000192	53	65	63	6F	6E	64	61	72	79	49	44	00	00	00	00	00	SecondaryID
00000208	00	00	00	00	00	00	00	00	00	00	00	00	50	52	49	56	PRIV
00000224	00	00	00	1F	00	00	57	4D	2F	57	4D	43	6F	6E	74	65	WM/WMConte
00000240	6E	74	49	44	00	13	2D	79	A5	4A	3C	EA	4F	8C	C2	A8	ntID -y#J<èOQĀ`
00000256	7B	22	9A	A6	05	50	52	49	56	00	00	00	8A	00	00	57	{"š; PRIV Š W
00000272	4D	2F	55	6E	69	71	75	65	46	69	6C	65	49	64	65	6E	M/UniqueFileIden
00000288	74	69	66	69	65	72	00	41	00	4D	00	47	00	61	00	5F	tifier A M G a _
00000304	00	69	00	64	00	3D	00	52	00	20	00	20	00	20	00	37	i d = R 7
00000320	00	39	00	34	00	38	00	30	00	33	00	3B	00	41	00	4D	9 4 8 0 3 ; A M
00000336	00	47	00	70	00	5F	00	69	00	64	00	3D	00	50	00	20	G p _ i d = P
00000352	00	20	00	20	00	20	00	20	00	34	00	30	00	37	00	31	4 0 7 1
00000368	00	3B	00	41	00	4D	00	47	00	74	00	5F	00	69	00	64	; A M G t _ i d
00000384	00	3D	00	54	00	20	00	20	00	38	00	34	00	31	00	36	= T 8 4 1 6
00000400	00	37	00	38	00	31	00	00	00	54	50	45	32	00	00	00	7 8 1 TPE2
00000416	0D	00	00	00	44	65	70	65	63	68	65	20	4D	6F	64	65	Depeche Mode
00000432	50	52	49	56	00	00	00	22	00	00	57	4D	2F	57	4D	43	PRIV " WM/WMC
00000448	6F	6C	6C	65	63	74	69	6F	6E	49	44	00	8D	76	3B	67	collectionID v;g
00000464	56	A2	D6	4C	85	44	A2	BC	5D	61	56	D4	50	52	49	56	VcÖL...Dc*]aVÖPRIV
00000480	00	00	00	27	00	00	57	4D	2F	57	4D	43	6F	6C	6C	65	' WM/WMColle
00000496	63	74	69	6F	6E	47	72	6F	75	70	49	44	00	8D	76	3B	ctionGroupID v;
00000512	67	56	A2	D6	4C	85	44	A2	BC	5D	61	56	D4	50	52	49	gVcÖL...Dc*]aVÖPRI
00000528	56	00	00	00	14	00	00	57	4D	2F	50	72	6F	76	69	64	V WM/Provid
00000544	65	72	00	41	00	4D	00	47	00	00	00	54	43	4F	4D	00	er A M G TCOM
00000560	00	00	0F	00	00	00	4D	61	72	74	69	6E	20	4C	2E	20	Martin L.
00000576	47	6F	72	65	54	50	45	31	00	00	00	0D	00	00	00	44	GoreTPE1 D
00000592	65	70	65	63	68	65	20	4D	6F	64	65	00	00	00	00	00	epeche Mode
00000608	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Рис. 1. Структура ID3-тега

За ID3-тегом располагается последовательность фреймов, в которых содержатся аудиоданные. Между заголовком ID3-тега и первым фреймом имеются незаполненные байты, в которые можно поместить дополнительную информацию.

Рассмотрим подробно структуру фреймов [4, 6]. Каждый фрейм содержит заголовок размером 4 байта, служебную информацию и аудиоданные. Описание полей заголовка представлено в таблице.

Структура фрейма

Название поля	Начальный- конечный байты	Размер (в бай- тах)	Значение
Framesync	0–12	1	Sync Header, всегда 0xFFEх
Идентификатор версии MPEG Audio	12–14	1	00=MPEG-2.5 (44100 Hz) 01=MPEG-? 10=MPEG-2.0 (22050 Hz) 11=MPEG-1.0 (11025 Hz)
Идентификатор уровня	14–15	1	00=Layer-? 01=Layer-3 10=Layer-2 11=Layer-1
Защита CRC	16–18	2	0=Error Protection Bit CRC is ON 1=Error Protection Bit CRC is OFF
Индекс битрейта	18–22	2	Bitrate
Индекс частоты дискретизации	22–24	2	Sample Rate Index
Бит заполненности	24–25	2	0=Unused bits are filled 1=All bits in frame are used
Персональный бит	25–27	—	0=Extension None 1=Extension Private
Каналы	27–29	3	00=Stereo 01=Joint stereo 10=Dual channel 11=Single Channel (Mono)
Расширение режима	29–31	3	Mode extension
Авторское право	31–32	3	0=Copyright NO 1=Copyright YES
Оригинал	32–33	3	0=Original NO 1=Original YES
Выразительность	33	3	00=Emphasis None 01=Emphasis 50/15 microseconds 10=Emphasis Dunno 11=Emphasis CITT j.17

Первые два байта заголовка открывают начало каждого фрейма. Они содержат следующие данные в шестнадцатеричном коде — FFFB. Таким образом, появляется возможность определить начало каждого фрейма. Другой важный параметр содержит поле Bitrate, определяющее количество информации, пере-

данное за секунду воспроизведения. Ниже приведены возможные значения этого параметра:

$$\{0, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320, -1\}. \quad (1)$$

Также для реализации представленного алгоритма потребуется параметр Sample Rate Index, определяющий частоту дискретизации. В зависимости от значения этого параметра частота дискретизации принимает одно из следующих четырех значений:

$$\{44\ 100, 48\ 000, 32\ 000, 50\ 000\}. \quad (2)$$

Для каждого фрейма значения этих полей могут быть разными, но определяться из указанных выше списков. Зная значения выше полей, можно определить размер данных фрейма в байтах по следующей формуле [6]:

$$D = 144\ 000 * bt / freq + p; \quad (3)$$

где bt — значение параметра поля $Bytrate$, которое определяется из множества (1), $freq$ — значение параметра поля Sample Rate Index, определяемое из множества (2), а p обычно равен единице.

В программном коде удобно реализовать содержимое приведенной выше таблицы в виде структуры, представленной на рис. 2 [7].

```
struct mp3_frame_hdr {
    unsigned short sync; //12 бит синхронизации + номер уровня MPEG
    unsigned char priv:1; // зарезервированный для частного использования бит
    unsigned char pad:1; // 1: фрейм дополняется одним байтом
    unsigned char freq:2; // частота семплирования, Гц
    unsigned char rate:4; // битрейт, кбит/с
    unsigned char mode:2; // режим стерео
    unsigned char mode_ext:2; // опции режима стерео
    unsigned char copy:1; // флаг копии
    unsigned char orig:1; // флаг оригинального материала
    unsigned char emph:2; // эмфазис
};
```

Рис. 2. Структура, реализующая содержимое таблицы

За байтами заголовка следует служебная информация, первые два поля которой содержат смещение начала данных фрейма от первого байта заголовка.

Анализ существующих методов сокрытия данных в аудиофайлах. В настоящее время разработано большое количество методов сокрытия информации в аудиофайлах. К ним относятся методы четного кодирования, фазового кодирования, расширения спектра и многие другие [8, 9]. Однако все эти схемы применимы для несжатых WAV-файлов. В файле типа MP3 аудиоданные претерпевают процедуру сжатия, в результате которой часть информации теряется. Однако вследствие несовершенства слуховой системы человека эти потери остаются незамеченными.

Наиболее часто ЦВЗ встраивается в различные служебные области MP3-файла. Например, как показано в предыдущем разделе, в ID3-теге существуют свободные байты между заголовком и первым фреймом. Именно в эту область может быть записан ЦВЗ. Однако такой прием страдает одним существенным недостатком: при поиске встроенной информации злоумышленник в первую очередь проверяет именно служебные области.

В данной статье предложен алгоритм сокрытия информации в MP3-файлах, который является более устойчивым к попыткам ее обнаружения.

Описание алгоритма сокрытия информации в MP3-файлах. Основная идея представленного алгоритма базируется на возможности увеличения пространства аудио данных путем изменения параметра *bt*, который находится в поле *Bitrate* заголовка каждого фрейма. Если значение этого параметр увеличить по сравнению с заданным, то согласно формуле (3) увеличится и размер области аудио данных. В результате между концом данных одного фрейма и началом следующего возникает свободная область, в которую можно записать скрываемую информацию.

При разработке алгоритма учитывалось требование противодействия попыткам обнаружить встроенный ЦВЗ. Подобные атаки обычно происходят по следующим направлениям.

Во-первых, под подозрения попадают фреймы, в которых существует свободная область между аудиоданными и следующим фреймом. Подобная ситуация может свидетельствовать о наличии встроенной информации, поскольку в оригинальном аудио файле такой свободной области нет. Фреймы следуют друг за другом без пробелов.

Во-вторых, пытаются найти несоответствия между заголовком фрейма и аудиоданными. В первую очередь анализируются параметры *Bitrate* и *Sample Rate Index*. Именно эти параметры определяют размер аудиоданных.

Ниже дано описание предложенного алгоритма встраивания ЦВЗ в MP3-файл. Показано, каким образом обходятся указанные выше способы обнаружения наличия скрытой информации.

1. На первом этапе в MP3-файле выделяются все фреймы и сохраняются в объекте контейнерного класса **vector**, входящего в библиотеку шаблонных классов C++. Отделение одного фрейма от другого выполняется по первым двум байтам заголовка каждого фрейма — FFFB.

2. На следующем шаге с помощью датчика случайных чисел определяется номер фрейма, в который будет встраиваться ЦВЗ. Для того чтобы данный номер генерировался и при извлечении ЦВЗ, инициализация генератора выполняется одним и тем же секретным ключом. Таким образом затрудняется поиск фрейма со встроенными данными.

3. Для выбранного фрейма в заголовке в поле *Bitrate* выбирается параметр *bt* и анализируется возможность его увеличения в соответствии с данными (1). Например, если значение *bt* было равно 192, то выбирается величина 224. Это значение сохраняется далее в поле *Bitrate*. В результате размер области данных

согласно формуле (3) увеличивается и появляется возможность записать в свободную область скрываемую информацию. При этом сохраняется соответствие между параметрами заголовка и аудиоданными фрейма. Если для данного фрейма нет возможности увеличения параметра bt (его значение оказалось равным 320), то определяется номер другого фрейма. В большинстве файлов типа MP3 значения поля *Bitrate* для разных фреймов отличаются друг от друга.

4. В образовавшуюся свободную область выбранного фрейма записывается секретная информация (ЦВЗ). При этом размер ЦВЗ целесообразно подбирать таким образом, чтобы образованная свободная область заполнялась полностью. В этом случае не будет пустых пространств между фреймами, что затруднит обнаружение ЦВЗ.

Также был разработан и алгоритм извлечения сокрытой информации из MP3-файла. Он во многом повторяет алгоритм сокрытия (а именно пункты 1 и 2), только вместо записи выполняется чтение ЦВЗ.

На основе предложенного алгоритма был разработан программный комплекс для встраивания и извлечения ЦВЗ применительно к файлам типа MP3. Выполненные с помощью этого комплекса тесты показали, что восприятие MP3-файлов со встроенным ЦВЗ не отличается от восприятия оригинальных файлов. Кроме того, во всех тестах извлеченный ЦВЗ полностью совпадает с ЦВЗ, встроенным в MP3-файл.

Заключение. В статье представлен алгоритм встраивания ЦВЗ в аудиофайл типа MP3. Показано, что данный алгоритм обладает повышенной устойчивостью к попыткам обнаружения скрытой информации. Продемонстрирована работоспособность программного комплекса, созданного на основе представленного алгоритма. Данный комплекс создан для противодействия распространения объектов авторского права в сети Интернет [10].

Литература

- [1] Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. М., Солон-пресс, 2002.
- [2] Шелухин О.И., Канаев С.Д. Стеганография. Алгоритмы и программная реализация. М., Горячая линия-Телеком, 2017.
- [3] Воронкова Д.К., Воронков А.С. Правовое регулирование и особенности производства комплексной компьютерно-технической и видеотехнической экспертизы мультимедиа файлов. *Всерос. студ. конф. «Студенческая научная весна»*. М., Научная библиотека, 2019, с. 124–127.
- [4] Jayant N., Johnston J., Safranek R. Signal compression based on models of human perception. *Proc. IEEE*, 1993, vol. 81, no. 10, pp. 1385–1422. DOI: <https://doi.org/10.1109/5.241504>
- [5] Карлова А.В. Некоторые особенности исследования графических файлов в шестнадцатеричном формате. *Политехнический молодежный журнал*, 2019, № 7. DOI: <http://dx.doi.org/10.18698/2541-8009-2019-7-501>
- [6] Finlayson R. A more loss-tolerant RTP payload format for MP3 audio. Internet Engineering Task Force, 2008.

- [7] Структура MP3 файла. *audiocoding.ru: веб-сайт*. URL: <https://audiocoding.ru/articles/2008-05-22-mp3-file-structure/> (дата обращения: 23.11.2019).
- [8] Кокорин П.П. О методах стегаанализа в аудиофайлах. *Труды СПИИРАН*, № 4, с. 239–246.
- [9] Чичварин Н.В., Волосатова Т.М. Метод сокрытия данных в стереофонических файлах. *Инженерный вестник*, 2015, № 9. URL: <http://engsi.ru/doc/792392.html>
- [10] Тарасов Д.А. Об эффективности судебной компьютерной экспертизы в процессе доказывания деяний, связанных с «компьютерным пиратством». *Вестник Московского Университета МВД России*, 2017, № 2, с. 100–101.

Серебрякова Софья Андреевна — студентка кафедры «Юриспруденция, интеллектуальная собственность и судебная экспертиза», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Филиппов Михаил Владимирович, кандидат технических наук, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Серебрякова С.А. Встраивание цифрового водяного знака в аудиофайлы формата MP3 для противодействия распространения контрафактной продукции. *Политехнический молодежный журнал*, 2020, № 05(46). <http://dx.doi.org/10.18698/2541-8009-2020-05-610>

EMBEDDING A DIGITAL WATERMARK IN MP3 AUDIO FILES TO COUNTER THE SPREAD OF COUNTERFEIT PRODUCTS

S.A. Serebryakova

sofyasofya98@gmail.com

SPIN-code: 9808-2216

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper considers methods of hiding information in audio files, proposes an algorithm for embedding digital watermarks in MP3 files. The structure of the MP3 file and the structure of the frame are considered. A software package has been developed for embedding and extracting digital watermarks to MP3 files. This algorithm is highly resistant to attempts to detect hidden information. The article also provides ways to bypass the detection of the presence of hidden information are bypassed. The relevance of the article is to protect copyright in musical and cinematographic works, as well as to counteract the spread of counterfeit products.

Keywords

Steganography, digital steganography, information hiding, information protection, MP3 file, algorithm, code, frame

Received 04.05.2020

© Bauman Moscow State Technical University, 2020

References

- [1] Gribunin V.G., Okov I.N., Turintsev I.V. Tsifrovaya steganografiya [Digital stenography]. Moscow, Solon-press Publ., 2002 (in Russ.).
- [2] Shelukhin O.I., Kanaev S.D. Steganografiya. Algoritmy i programmaya realizatsiya [Stenography. Algorithms and software realization]. Moscow, Goryachaya liniya-Telekom Publ., 2017 (in Russ.).
- [3] Voronkova D.K., Voronkov A.S. [Legal regulation and special aspects of manufacturing complex computer and video-technical expertise of multimedia files]. *Vseros. stud. konf. "Studencheskaya nauchnaya vesna"* [Russ. Students Conf. "Students science spring"]. Moscow, Nauchnaya biblioteka Publ., 2019, pp. 124–127 (in Russ.).
- [4] Jayant N., Johnston J., Safranek R. Signal compression based on models of human perception. *Proc. IEEE*, 1993, vol. 81, no. 10, pp. 1385–1422. DOI: <https://doi.org/10.1109/5.241504>
- [5] Karlova A.V. Some features of study of image files in hexadecimal format. *Politekhnicheskiiy molodezhnyy zhurnal* [Politechnical student journal], 2019, no. 7. DOI: <http://dx.doi.org/10.18698/2541-8009-2019-7-501> (in Russ.).
- [6] Finlayson R. A more loss-tolerant RTP payload format for MP3 audio. Internet Engineering Task Force, 2008.
- [7] Struktura MR3 fayla [Structure of MP3 file]. *audiocoding.ru: website* (in Russ.). URL: <https://audiocoding.ru/articles/2008-05-22-mp3-file-structure/> (accessed: 23.11.2019).
- [8] Kokorin P.P. About methods of steganalysis in audio files. *Trudy SPIIRAN* [SPIIRAS Proceedings], no. 4, pp. 239–246 (in Russ.).
- [9] Chichvarin N.V., Volosatova T.M. Data hiding method in stereophonic files. *Inzhenernyy vestnik* [Engineering Bulletin], 2015, no. 9. URL: <http://engsi.ru/doc/792392.html> (in Russ.).

- [10] Tarasov D.A. On computer forensics efficiency in the course of proving an act associated with "software piracy". *Vestnik Moskovskogo Universiteta MVD Rossii*, 2017, no. 2, pp. 100–101 (in Russ.).

Serebryakova S.A. — Student, Department of Law, Intellectual Property and Forensic Examination, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Phillipov M.V., Cand. Sc. (Eng.), Assoc. Professor, Department of Computer Software and Information Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Serebryakova S.A. Embedding a digital watermark in MP3 audio files to counter the spread of counterfeit products. *Politekhniicheskiy molodezhnyy zhurnal* [Politechnical student journal], 2020, no. 05(46). <http://dx.doi.org/10.18698/2541-8009-2020-05-610.html> (in Russ.).