Rafał Korycki, Ph.D.
Expert in the field of audio and image engineering from Forensic Research Office of the Internal Security Agency
r.korycki@abw.gov.pl

Identification of MP3 encoders and recording equipment based on examination subjected to lossy date compression[1]

Summary

The paper outlines the problem of identification of MP3 encoders and recording devices based on the analysis of audio recordings subjected to lossy date compression. The proposed method can be used as a support for other solutions used to detect double compression and discontinuities. The approach is based on the statistical analysis of the variables obtained directly from the MP3 data stream and constitute an inherent element of compression performance. Designated vectors consisting of 46 features were used as training sequences of linear discriminant analysis (LDA), one of the most popular supervised machine learning algorithms. The effectiveness of this algorithms for the identification of MP3 encoders and recording equipment was tested on a music database consisting of nearly one million MP3 files, specially prepared for this purpose. The results of the research were discussed in the context of the influence of compression parameters on the ability to detect falsification in digital audio recordings.

Keywords authenticity examination of digital audio recording, identification of MP3 encoders, identification of recording equipment, digital evidence analysis, linear discriminant analysis

Introduction and purpose of article

Examination of the authenticity of audio recording is based on the assessment of its originality and detection of signs of interference in its continuity. In accordance with the decision of the Supreme Court of Poland in relation to case REF. Act III K 49/61 of 10 March 1961 [16] the task of forensic expert delivering a forensic casework for judicial and law enforcement bodies is to determine whether the examined recording is original (specific, identical) and to detect the possible evidence of interference in its continuity. The definition of authentic recordings, according to standard AES27-1996 drawn up by the international organisation of the AES (Audio Engineering Society) [1] is also an extremely important matter. This definition indicated the necessity to study the originality of records and detect traces of editing, emphasising the need to analyse the compliance of recording techniques with the techniques applied during the recording of examined audio. In addition, attention was paid to the integrity of the examined recording with the acoustic sounds accompanying the recording process.

In publications [9] and [10], which describe the results of the examinations carried out within the framework of the research project, the author has developed a set of algorithms that allow, among others, to distinguish recordings subjected to single compression (e.g. files recorded with digital voice recorders) from recordings encrypted again, which may have been edited by using sound editing software. In addition, the author proposed a method for identification of discontinuities in the recordings subjected to lossy compression, which allows to detect deletions and insertions of fragments of recordings. The study uses a vector of features derived from the intrinsic parameters of MP3 files as well as the selected supervised machine learning methods. The conclusion derived from the above mentioned studies pointed to the fact that in order to determine the proper implementation of the initial bitrate prediction processes, detect multiple compression and edition of recordings subjected to multiple encoding and decoding processes, it is necessary to know the encoder used to create the examined audio file.

1 Research carried out within the framework of the project entitled: „Design of empirical research and analysis of the materials concerning the specificity of methods of forensic science in the work of the special services of public order services" funded by the National Centre for Research and Development in the form of a grant no. 0023/R/ID3/2012/02.

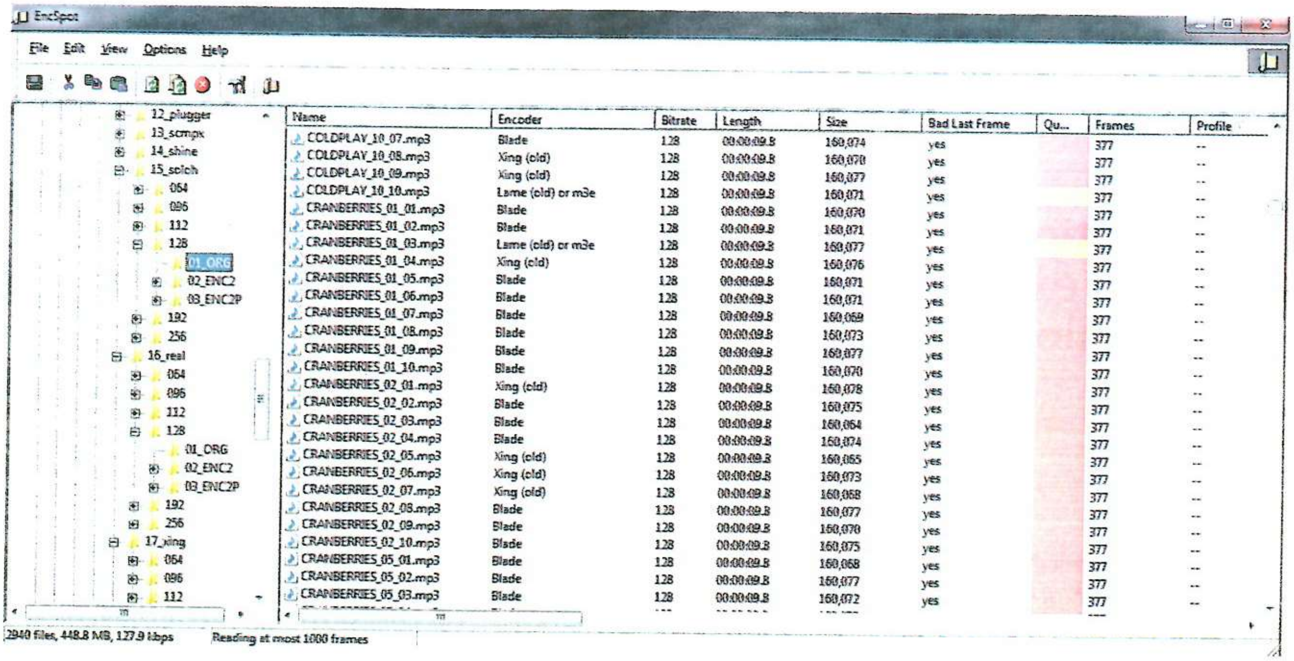

Fig. 1. EncSpot user interface with a report from the classification of MP3 files

In many publications, the authors have already pointed out the possibility of obtaining information about the applied compression algorithm based on analysis of a compressed file. Selected works discussed, among others, the determination of a few selected encoding parameters, e.g. quantization and scaling factors [5], [7], and the identification of different compression algorithms (e.g. MP3, AAC, OGG, WMA, AMR) [2], [6]. On the market there is also commercial software (e.g. EncSpot [4]) for recognizing the type of MP3 encoder used on the basis of a compressed MP3 file. However, it is worth noting that these programs are mainly based upon information contained in the headers and metadata of the analysed files (such as ID3 tags [14]). Although, the information obtained on this basis is very detailed, it may be freely modified without affecting the parameters of recordings. In addition, some implementations of the MP3 encoder algorithm identification are subject to significant error. For example, when trying to identify encoder SoloH, the application EncSpot pointed at random to one of the three other encoders: Blade, Xing and Lame. Figure 1 shows the EncSpot user interface with the results of a sample identification. Only Böhme and Westfeld proposed solution to distinguish between different implementations of the same compression algorithm [3]. However, the use of this method was limited to a specific list of encoders provided by the authors due to the use of the classification of only ten symbolic characteristics taking discrete value for classification.

The aim of the research described in the current publication was to develop methods for the detection of types and versions of various MP3 algorithm implementations used to create the examined audio files examined. Based on the solutions described in mentioned publications, the author has proposed his own set of characteristics, enabling both the correct detection of the encoder used for the compression of examined recordings, as well as quick adaptation of classifiers to new implementations of the MP3 algorithm or those not included in this study. To check the effectiveness of the developed methods, a proprietary database of audio recordings was created, consisting of nearly one million MP3 files. In the opinion of the author, this approach is conducive to the objectification of research and allows easy comparison of examination methods. In addition, the popular machine learning methods were used, which may be helpful in the diagnosis and classification of the various features of different encoders. Studies have shown that by properly prepared sequences it is possible to create an algorithm applied for identification of MP3 encoders and recording equipment on the basis of the examination of auiod files subjected to lossy compression.

Bearing in mind the increasingly widespread use of portable audio recorders in which lossy compression is used, the examination was based on the analysis of different coding algorithms. The author focuses on the analysis of MPEG-1 Layer 3 (MP3) recordings subjected to lossy compression[12]. This choice was dictated by a considerable amount of literature on the subject, the availability of documentation and the widespread use of MP3 algorithm. It is worth mentioning that the method presented in this paper can also be used for other compression algorithms (eg. AAC - Advanced Audio Coding, Ogg Vorbis), where the standard does not define an accurate and unambiguous manner of each operation performed by the algorithm.

For the purpose of describing the carried out examination, the author decided to introduce the following terms: "source recording" refering to fragments of music from original CDs, and "original recording" to those subjected to single compression.

Studies have been carried out within the framework of the project: "Design of empirical research and analysis of the materials concerning the specificity of methods of forensic science in the work of the special services of public order services" funded by the National Centre for Research and Development in the form of a grant no. 0023/R/ID3/2012/02.

Selected issues related to MP3 compression

MP3 algorithm described in ISO 11172-3 standard [12] defines the method for determining the large number of variables relevant to the compression process and defines the way of audio signal recording. This ensures that the binary stream generated by the encoder will be read and interpreted correctly. Although the standard is extremely detailed, it does not explain each step performed by the software, leaving application developers the freedom to implement every encoder block. As a result, the MP3 algorithm can be implemented in different ways, and yet the compressed audio files and these originating from different sources will be played by the decoder [12], [15].

MP3 file data stream consists of fixed length frames. The frame consists of two granules, which include data needed for decoding the value of MDCT coefficients for one (mono recording) or two channels (stereo recording). The frames are composed of a header, file integrity checking CRC (*cyclic redundancy code*), side information, and main data. The header is 32 bits long and it starts with 12 bit synchronization, so that each frame is individually localized by the decoder. This allows data streaming, and the recipient can start listening to the continuous transmission at any point of its duration. In addition, the file header contains, among others, information about the MPEG-1 layer, bit rate, sampling frequency, dual-channel signal encoding mode, the use of emphasis[2] and protecting the integrity of the file protection bit[3].

The side information contained in the next section of a frame is necessary for proper reading of the main data containing the quantized and coded spectral bands. Section starts with *main_data_end* variable, which is an indicator of the beginning of next frame header (measured from the end of the section occupied by the main data of current frame). In this way, the size of the container is defined, whose function is to virtually increase the capacity of the frame when forming the stream with a smaller volume would result in exceeding the quantization noise threshold value based on the estimated on the basis of the value of entropy spectral psychoacoustic portion of the signal, which is fixed in the psychoacoustic model block.

The side information contains also private bits, and a *scfsi* variable (*scalefactor selection information*). It determines whether the same set of parameters defined as *scalefactors*, usewd to scale the spectrum bands in the critical band, is utilized in both granules, or whether for each of them it must be sent individually. Another set of data making up the side information is transmitted separately for each of granule. These include, among others, the *big_values* variable, which constitute the number of spectral bands and are encoded by more than three levels of quantization, the bits indicating the use of one of four predefined time window in the second filter bank or variables recorded as *table_select*, which constitute the numbers in Huffman table to be used for decoding the main data encoded in the MP3 file using the Huffman compression algorithm.

The last section of the frame consists of the scaling factors, binary data encoded using Huffman compression algorithm and ancillary data. The task of the scaling factors is to amplify and damp MDCT spectrum bands that are included in the ranges defined as *scalefactor bands* in order to effectively mask the quantization noise based on data provided by the psychoacoustic model. The ranges of the *scalefactor bands* are selected in such a way that helps to reproduce the critical bandwidth as closely as possible [12], [15].

Extraction of the feature vector

Based on the analysis of ISO / IEC 11172-3 standard [12] and preliminary experimental examination, a vector was constructed to recognize 46 features enabling the implementation of the selected MP3 algorithm that was used to create the examined audio file. In order to construct the vector, the values extracted from each frames were used. The values were selected in such a way that the persons editing the recordings had no ability to modify them without adding distortion (e.g. while editing the binary data stream of MP3 file in the editor). Examination results obtained on the basis of a set of ten features of symbolic taking discrete

2 Emphasis mode means that the file is compressed using one of the two modes of noise reduction, involving the strengthening of high frequencies during encoding and reduction during decoding. Similar solutions (e.g. Dolby-B™) are widely used in analogue systems.

3 If the protection bit is set, a 16 bit CRC is added to enable checking the 16 bits of the header and 16 bits of side information. These are the data relevant to correctly restore the contents of the frame. In the event of an error, the frame can be muted or replaced by the previous frame.

values [3] served as a point of reference. This solution, although allows for effective classification within the framework of proposed by the authors list of encoders, it significantly impedes the extension of an existing set with new implementations of MP3algorithm. Therefore, when creating a new vector of the length of 46 items, the author used also continuous variables, and extended the catalogue of parameters extracted from the MP3 files. This allows to add a new encoder to the recognized list only by generating an appropriate training sequence for the machine learning algorithms.

Due to the differences between MP3 compression algorithm implementations, actual value of bit rate differ from their nominal counterparts[4], stored in the header of each frame. It is connected to with the manner of determination of control parameters defined for nested quantization loop and compression, which helps the algorithm to a compromise between file size and the level of distortion introduced. The above relationship was used to create the first features that was determined according to the relationship:

$$F1 = \frac{(f_{size} - h - m) \cdot f_s}{1152 \cdot \sum_{i=1}^{p} \beta_{nom}^{(i)}}, \quad (1)$$

where f_{size} is the file size, h is the size of the header, m means the metadata stored in a file, f_s is the frequency of the sampling rate, p is the number of frames in the file, and $\beta_{nom}^{(i)}$ is the value of the nominal throughput for i-th frame.

During execution of the compression algorithm the granule size is chosen individually in such a manner as to fulfil the criterion of matching the length of a frame at the lowest possible level of distortion. Depending on the implementation, the first granule in each frame may be larger or smaller than the other, therefore the following three F2:F4 features determine the number of frames, in which the first granule is greater than the second, determined for both channels and separately for left and right channel. Further eight elements of the feature vector F5:F12 take discrete values determined by the choice of one of the techniques used during the recording of two-channel audio files. The method of processing data from each channel is stored in the header of each frame. It is worth mentioning that in some MP3 encoders, for example 8 Hz, Blade, Shine, the mode is not applied to *joint stereo*[5].

The next group of features is based on the variable *main_data_end*, which specifies the size of the reservoir. The control of this mechanism largely dependents on the programmer creating the encoder. Thus, the rate of filling in the reservoir in the case of an audio signal may be dependent on the type and version of the MP3 algorithm implementation. For example, encoders 8 Hz, Blade and Shine, divide between frames much less data than encoders of the Lame family. In view of the foregoing, the features of F13:F14 were defined, which specify the number bytes shared between the first two pairs of frames, and the features of F15:F18, which constitute respectively the mean value and standard deviation of the size of the reservoir and its maximum and minimum capacity. Figure 2 shows the distribution of feature F15 values (average value of the reservoir's size) for the first seventeen coders identified in the literature [9] and [10] generated using *boxplot* function of MATLAB computing environment [11]. This function creates a set of graphs, in which the central rate (horizontal red line) is the median, the edges of the box are the 25th and 75th percentile, stripes define the maximum values, and the values of outliers were marked with red crosses.

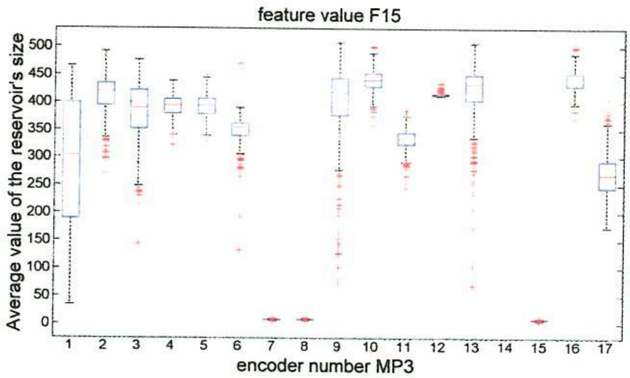

Fig. 2. Distributions of F15 feature value for the first 17 MP3 encoders indicated in publications [9] and [10].

The value of spectrum bands for each of granules are grouped into bands and then encoded using different Huffman tables. This division is carried out considering maximum value of the MDCT and assuming that in the range of high frequencies quantized spectral coefficients with small amplitudes occure, some of

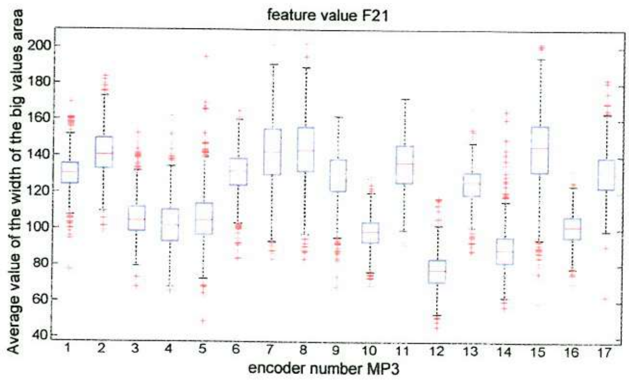

Fig. 3. Distributions of F21 feature value for the first 17 MP3 encoders indicated in publications [9] and [10].

4 Standardized values of bit rate for MP3 files are as follows: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320 kbps.

5 The *joint stereo* is a separate part of the common encoding set each for both channels and the difference between them. It comes in two versions: *intensity* and *mid/side*.

which do not need to be recorded at all. The most significant frequency range is referred to as *big_values* and refers to the bands represented by more than three levels of quantization. Their lowest and highest values, as well as the mean value and standard deviation are describe by the features F19: F22. Figure 3 shows the distributions of values of feature F21 (the average value of the width of the big_values area) for the first 17 coders identified in publications [9] and [10] generated using the *boxplot* function [11].

The variables F23:F24 are determined by the entropy of histogram which is determined for each channel on the basis of the number of occurance of the first 60 MDCT coefficients [3]:

$$F23 = -\sum_{j=1}^{60} d_j^{(L)} \log_2 d_j^{(L)}, \quad F24 = -\sum_{j=1}^{60} d_j^{(R)} \log_2 d_j^{(R)} \tag{2}$$

where $d_j^{(L)}$ and $d_j^{(R)}$ denote the frequency of the j-th band of the MDCT spectrum respectively for the left ("L") and right ("R") channel.

The range specified as *big_values* is divided into three areas by two variables, *region_address1* and *region_address0*, determining the number of *scalefactor bands* they occupy. Their minimum and maximum values are reflected in the features F25:F28. Furthermore, depending on the energy distribution of the spectrum of the audio signal, various Huffman tables are used for encoding. However, it is acceptable to use 32 tables but a part of them remains unused in some coders. Accordingly, the following five featuress F29:F33 were assigned to use tables 1 and 9:12 in the first area of *big_values*, table 0 in all three areas, and table 23 in the second and third areas.

ISO 11172-3 specification [12] indicates that the use of emphasis consisting in strengthening the high frequencies during encoding and reducing their level during decoding to reduce noise is dependent on the will of the programmer creating the encoder. As a result, the next two features were determined based on the variable *preflag* describing the use of this function in each of the frames:

$$F34 = \frac{1}{p}\sum_{i=1}^{p} preflag(b_i^{(L)}), \quad F35 = \frac{1}{p}\sum_{i=1}^{p} preflag(b_i^{(R)}) \tag{3}$$

where p is the number of frames in the file, and $b_i^{(L)}$ and $b_i^{(R)}$ represent the data blocks for the left and right channels. Figures 4 and 5 show the percentage of distribution of F34 and F35 feature values for the first 17 encoders, as indicated in publication [9] and [10] generated using *boxplot* function.

The next tested variable (*scfsi*) is also optional and allows to specify within which frames the same set of parameters, defined as scalefactors and used to scale the values of spectrum bands in a given critical

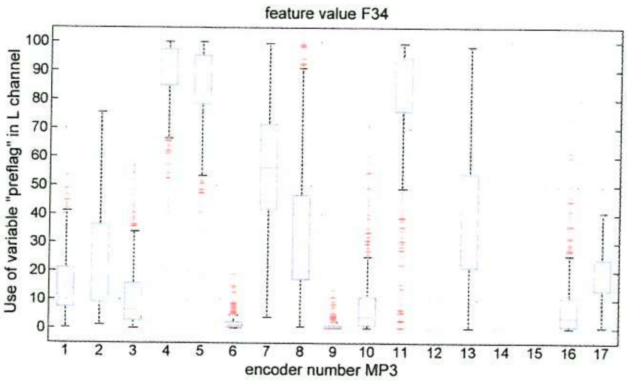

Fig. 4. Distributions of F34 feature value for the first 17 MP3 encoders indicated in publications [9] and [10].

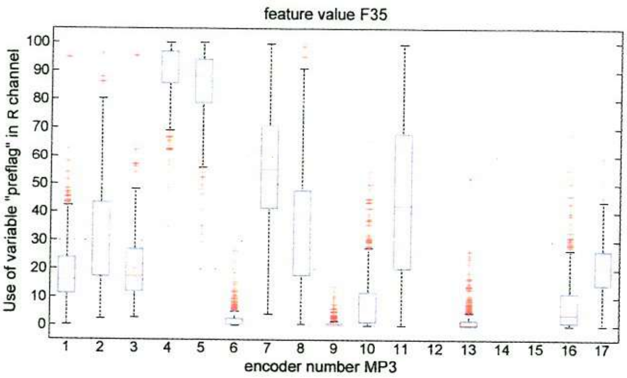

Fig. 5. Distributions of F35 feature value for the first 17 MP3 encoders indicated in publications [9] and [10].

band, is used in both granule and within which frames the set of parameters must be send to each granule individually. Thus, another component of the feature vector is determined as follows:

$$F36 = \frac{1}{p}\sum_{i=1}^{p} scfsi(b_i), \tag{4}$$

where p is the number of frames in the file and b_i constitutes the data blocks. Figures 6 shows the percentage distributions of feature F36 values for the first 17 encoders, as indicated in publications [9] and [10] generated using *boxplot* function.

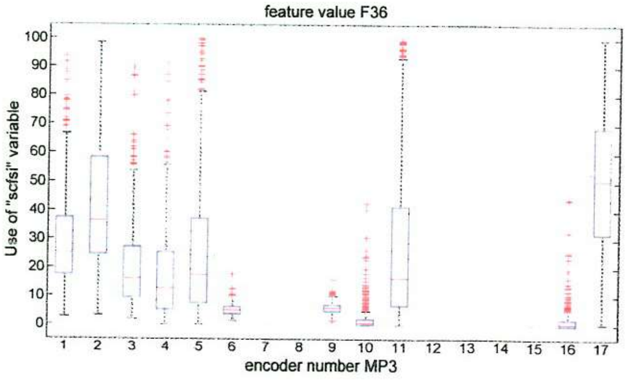

Fig. 6. Distributions of F36 feature value for the first 17 MP3 encoders indicated in publications [9] and [10].

The length and the number of frames per one second are not always identical with the corresponding number of samples of the input audio signal. Therefore it is necessary to add, usually every few frames, the relevant window (*slot*), which aims to extend it. This operation, referred to as padding, is implementation-dependent, and the details of its use can be used to identify the type and version of the implementation of MP3 algorithm. For example, for a sampling rate of 44100 Hz and a throughput of 128 kbps, at the beginning of the files compressed with Blade and iTunes encoders, occur successively 24 and 25 frames padded and separated by one frame without the so-called slot. However in Helix encoder the baove sequence is preceded by a single frame without padding. There are also encoders (e.g. Adobe CS6) in which all frames have the same length. Therefore, a further six features F37:F42 contains, respectively the minimum and maximum length of the frame, the size of the first frame and the number of padded frames in the first three groups of mentioned sequence.

The MP3 algorithm provides for the use of long and short time windows in the second group pf filters before appointing the MDCT transformation, which results in the suitable appointment of respectively one implementation of 576 spectral bands or three implementations of 192 spectral bands each. Due to this solution, the blocks with larger number of samples are used for better representation of signal tones, whereas the short windows are used to ensure good reproduction of signals rapidly changing in time. In order to switch between the mentioned types of analysis smoothly and without any distortions, transitional windows are applied (start and stop). However, in many versions of codecs (e.g. Xing) short blocks do not apply. Still, in implementations such as Plugger some forbidden transitions may occur, e.g. directly between long and short blocks. The above observations were used during the creation of subsequent feature [3]:

$$F43 = \begin{cases} 1 \text{ for } & type(b_i) = 0 \text{ for all } i \\ 2 \text{ for } & type(b_i) = 0 \wedge type(b_{i+1}) = 2 \\ 3 \text{ for } & type(b_i) = 2 \wedge type(b_{i+1}) = 3 \\ 4 \text{ for } & |\{b_i | type(b_i) = 2\}| = \\ & = |\{b_i | type(b_i) = 3\}| = 1 \\ 5 \text{ } \textit{in other cases} & \end{cases}, \quad (5)$$

where 0 and 2 are, respectively, long and short windows, while 1 and 3 relate to the transition windows.

Regardless the manner of using the reservoir, while executing the MP3 compression algorithm there may appear a space in the frame unfilled with data. ISO 11172-3 specification [12] does not specify byte values that should be used to fill these spaces. Programmers are therefore free to choose these values (ie. *stuffing bits*). For example, in the files compressed using Gogo encoder one can find 47 and 4F sequence of stored in hexadecimal code, which means "GO" in ASCII code. Therefore, the next variable of feature vector was used to describe the above data:

$$F44 = \begin{cases} 1 & \text{for zero bites padding} \\ 2 & \text{for one bite padding} \\ 3 & \text{for padding in with sequence} \\ & \text{0x47 i 0x4F} \\ 4 & \text{for padding in with sequence} \\ & \text{0xAA lub 0x55} \\ 5 & \text{for lack of padding in} \\ 6 & \text{in other cases} \end{cases}. \quad (6)$$

The last two features F45:F46 take binary values and denote the defective last frame or synchronization error that most often result from improper determination of the end of the file and the beginning of the next header.

Classification of MP3 encoders

Based upon the analysis of the division of values of five features (Fig. 2-6) significant differences can be observed in the taken values depending on the encoder used. But in order to make a precise identification of the version of the MP3 algorithm used it is necessary to use a solution that allows for the automatic classification of the multi-dimensional data vectors. For this purpose the supervised machine learning methods are used that rely on deduction of an unknown function on the basis of a training data sequence. The training data consist of a set of examples that contain a data vector and the desired output state (called the control signal). The learning algorithm analyses the training sequences and deduces an unknown function (of the model), which is called a classifier (in the case of discrete data) or the regression function (for continuous data). A properly selected classifier should correctly predict the output state for each correct set of data [13].

For the purpose of evaluating the effectiveness of the classification of the MP3 encoders and recording equipment a special database was designed and created. 2940 pieces of music (10 seconds each) from 36 albums from the private collection of the author were used in the exaination. All source recordings (2940) were preserved in linear and lossless LPCM (Linear Pulse Code Modulation) and saved in files with the extension ".wav" and compressed using 21 different MP3 encoder implementations[6]. The examination was based on the MATLAB computing environment, in particular, the following tool packages: Signal Processing Toolbox, Parallel Computing Toolbox, Statistics Toolbox and the Bioinformatics Toolbox [23]. The classification of the feature vectors was performed

6 A list of MP3 encoders used in the study can be found in literature [9] and [10] of the author.

Table 1 Classification matrix of the encoders using linear discriminant analysis. Files compressed using 112 kbps bit rate

Anticipated encoder	Actual encoder (throughput: 112 kbps)																
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8 Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing
Lame 396	57.82	0.68	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	42.18	99.32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	0.68	-	-	-	-	-	-	-	-
8 Hz	-	-	-	-	-	-	96.94	-	-	-	-	-	-	-	-	-	-
Blade	-	-	-	-	-	-	1.02	100	-	-	-	-	-	-	1.08	-	-
GoGo	-	-	-	-	-	-	-	-	99.32	-	-	-	-	-	-	-	-
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
SoloH	-	-	-	-	-	-	2.04	-	-	-	-	-	-	-	98.98	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100

Table 2 Classification matrix of the encoders using linear discriminant analysis. Files compressed using 128 kbps bit rate

Anticipated encoder	Actual encoder (throughput: 128 kbps)																
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8 Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing
Lame 396	57.48	2.38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	42.52	97.62	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-
8 Hz	-	-	-	-	-	-	92.52	-	-	-	-	-	-	-	-	-	-
Blade	-	-	-	-	-	-	1.36	100	-	-	-	-	-	-	3.06	-	-
GoGo	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
SoloH	-	-	-	-	-	-	6.12	-	-	-	-	-	-	-	96.94	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100

Table 3 Classification matrix of the encoders using linear discriminant analysis. Files compressed using 192 kbps bit rate

Anticipated encoder	Actual encoder (throughput: 192 kbps)																
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8 Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing
Lame 396	60.20	-	2.38	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	39.80	-	97.62	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-
8 Hz	-	-	-	-	-	-	74.83	-	-	-	-	-	-	-	30.27	-	-
Blade	-	-	-	-	-	-	1.70	100	-	-	-	-	-	-	1.70	-	-
GoGo	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	0.34
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	3.74	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
SoloH	-	-	-	-	-	-	23.47	-	-	-	-	-	-	-	68.03	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	96.26	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	99.66

using linear discriminant analysis with matrix inversion function using the Moore-Penrose method [8], [11]. The feature vectors representing different encoders were divided into two equal sequences: training and testing. The training sequence was used to create models with LDA algorithm, whereas the effectiveness of classification and accuracy of detecting encoders, was examined with test sequence.

The performed classification was based on the prediction of types and versions of different implementations of MP3 algorithm that have been used to create the examined audio files. Tables 1-3 show the results of the encoders prediction (expressed in percent) depending on the bit rate, respectively 112 kbps (table 1), 128 kbps (table 2) and 192 kbps (table 3). For example, 96.94% of all compressed files by the 8 Hz encoder using 112 kbps bit rate, 1.02% of the files identified as generated using Blade encoder and 2.04% compressed with SoloH implementation (see Table 1) were classified correctly. Furthermore, in the case of the use of Real encoder version it was not possible to compress a file at 112 kbps bit rate (software does not provide this option).

Based on the analysis of the above results it can be seen that the prediction of the MP3 encoders used to compress the audio recordings is possible, and its effectiveness for the vast number of implementation exceeds 95% of the correct indications. In the case of encoders 8 Hz and SoloH, the number of correctly classified files decreases with the increase of bit rate. This may be related to the high similarity between these MP3 algorithm implementations (SoloH encoder is based on the source code of encoder 8 Hz). It should also be noted that the used classification method (linear discriminant analysis) was not fully effective in detecting the selected encoders. In Figures 4 and 5, the distribution of F34 and F35 feature values are different for encoders 7 (8 Hz) and 15 (SoloH), which should allow the precise distinction between these two implementation of MP3 algorithm. However, the results presented in Tables 1-3 show that each of the examined cases presents false indication. Therefore, it would be advisable to seek other methods of supervised machine learning, which would better exploit the relationship between the examined features, e.g. Binary decision tree (BDT) as described by the author in publication [9].

Classification of the recording equipment

The aim of this study was to develop an algorithm which allows for both effective classification within the defined list of encoders, and expansion of this classification with new MP3 algorithm implementations. Therefore, the vector of 46 features was designed so that its complexity will allow to reveal the unknown dependencies, which can be represented by new MP3

Table 4 Classification matrix of the encoders using linear discriminant analysis. Files compressed using 128 kbps bit rate [9]

Anticipated encoder	Actual encoder (128 kbps bit rate)																				
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8 Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing	Zoom	Tascam	Olympus	Sony
Lame 396	53.07	3.74	7.14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	35.03	95.58	6.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	11.90	0.68	86.74	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	-	-	-	-	0.68	-	-	-	-	-	-	-	-
8 Hz	-	-	-	-	-	-	93.54	4.42	-	-	-	-	-	-	-	-	-	-	-	-	-
Blade	-	-	-	-	-	-	-	92.18	-	-	-	-	-	-	-	-	-	-	-	-	-
GoGo	-	-	-	-	-	-	-	-	99.66	-	-	-	-	-	-	-	-	-	-	-	-
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	0.34	-	-	-	-	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	11.54	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	99.32	-	-	-	-	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-
SoloH	-	-	-	-	-	-	6.46	3.40	-	-	-	-	-	-	100	-	-	-	-	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	99.66	-	-	-	-	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Zoom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	98.70	-	-	-
Tascam	-	-	-	-	-	-	-	-	0.34	-	-	-	-	-	-	-	-	-	87.16	-	-
Olympus	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1.30	-	100	-
Sony	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1.30	-	100

algorithm implementations or those not included in this study. The addition of a new encoder to the list of recognized ones requires only the generation of appropriate training sequence for machine learning algorithms.

To demonstrate the possibility of adapting the proposed solution to the list of 17 encoders, four types of equipment that are portable recording devices were added: Zoom, Tascam, Olympus, and Sony. This time, the MP3 files were created directly by registering about 10 seconds of music, played from Adam P22 speakers, using the above-mentioned voice recorders. The performed classification was based on the prediction of types and versions of different MP3 algorithm implementations that have been used to create the examined files. Table 4 presents the results of prediction of the encoders (given in percent) for 128 kbps bit rate. For example, 93.54% of all the files compressed by the 8Hz encoder were classified correctly, and 6.46% were classified as encoded using the SoloH implementation.

Based on the analysis of these results it can be seen that the prediction of the MP3 encoders used to compress the audio recordings is possible, and its effectiveness for the vast number of implementations exceeds 95% of the correct indications. It is worth mentioning that despite the lack of detailed information on the four added MP3 algorithm implementations, the classification of files created with recording devices (Zoom, Tascam, Olympus and Sony) has also proved to be possible, and its effectiveness was the lowest in case of encoder used in the Tascam DR-40 and amounted to approx. 87%.

Information on the encoder vs. double compression detection

Within the framework of the examination carried out in connection to the implementation of research project, the results of which have been described in publications [9] and [10], the author has developed a set of algorithms to detect the source of bit rate of double compression MP3 recordings, distinguish between recordings of single compression from the recordings encoded twice and the identification of gaps in the MP3 recordings subjected to lossy compression. An important conclusion from the above examination, formulated in the publications, was the fact that for the proposed algorithms to work properly it is necessary to have knowledge of the encoder used to create the examined file. Most crucial in this case are the type and version of the MP3 algorithm implementation which was used as the last during the compression process of analysed recordings.

Bearing in mind the above considerations, the author decided to demonstrate the impact of a training model generated for the linear discrimination analysis algorithm on the effectiveness of the process of binary classification which aims to distinguish the original recordings (i.e. source recordings subjected to single compression) from audio recordings of double compression. The examination was carried out on the basis of the recordings created from the database created by the author (see [9], [10]). Fragments of songs (source recording) were compressed using three selected implementations of the MP3 algorithm: 399 Lame, iTunes and Adobe CS6, decoded into a lossless format, and then re-encoded using the same three selected encoders. From the created MP3 files vectors of 228 features lenght were extracted (see. [9], [10]), which was subjected to classification using linear discriminant analysis (LDA). In the case of each performed classification, the feature vectors were divided into two equal sequences: training and testing. During the process of classifier learning, three training models were created, corresponding to three implementations of MP3 algorithm: 399 Lame, iTunes and Adobe CS6. In other words, each training model was created as a result of comparison of the original recording, with those subjected to the double compression, which were created with one of the three encoders. Thanks to this, a percentage of correctly classified MP3 files present in the test sequence was determined depending on the model of training. Examinations were repeated for three selected bit rates: 112, 128 and 192 kbps.

Tables 5-7 present the results of the detection of double compression (given in percent) depending on the training model used for three selected bit rates: 112 kbps (tab. 5), 128 kbps (tab. 6) and 192 kbps (tab. 7). For example, in the course of the classification of recordings subjected to compression using Lame 399 encoder and 112 kbps bit rate when during the examination a model was used, which was created by application of the recordings encoded with the same codec, 98.81% the files has been properly assigned to the class. However, changing the model into the one created with the use of compressed files using Adobe CS6 encoder reduced the proportion of correctly classified files to 52.04% (see table 5).

As it can be seen, for proper implementation of the double compression detection process it is necessary to know the type and version of MP3 implementation algorithm that was used to create the examined file. It turns out that only the training model, which was generated on the basis of recordings compressed with the same encoder that was used to compress the files in the test sequence, can perform the classification, which will be burdened with the least error.

Table 5 Binary Classification of recordings subjected to single and double MP3 compression using the LDA algorithm and the various training models. The study used MP3 files compressed with 112 kbps bit rate [9]

The encoder used to train the classifier	**The encoder used to test the classifier**		
	Lame 399	Adobe CS6	iTunes
Lame 399	98.81	65.99	37.59
Adobe CS6	52.04	91.67	54.93
iTunes	47.62	41.67	97.79

Table 6 Binary Classification of recordings subjected to single and double MP3 compression using the LDA algorithm and the various training models. The study used MP3 files compressed with 128 kbps bit rate [9]

The encoder used to train the classifier	**The encoder used to test the classifier**		
	Lame 399	Adobe CS6	iTunes
Lame 399	98.30	61.90	35.88
Adobe CS6	50.34	87.93	42.01
iTunes	49.15	56.12	97.79

Table 7 Binary Classification of recordings subjected to single and double MP3 compression using the LDA algorithm and the various training models. The study used MP3 files compressed with 192 kbps bit rate [9]

The encoder used to train the classifier	**The encoder used to test the classifier**		
	Lame 399	Adobe CS6	iTunes
Lame 399	87.93	51.87	49.49
Adobe CS6	59.69	79.93	55.95
iTunes	49.49	62.93	98.98

Summary

The results of the examination carried out within the framework of research project show that it is possible to detect the types and versions of different implementations of the MP3 algorithms used for the examined file. Based on the available solutions, the author has proposed its own set of characteristics, enabling not only the correct detection of compression parameters, but also quick adaptation to the new MP3 algorithm implementations or those not included in this study. To check the effectiveness of the developed methods, a database of audio files was created, consisting of nearly one million MP3 files. The research has shown that from properly prepared sequences it is possible to create a learning algorithm to identify MP3 encoders and recording equipment on the basis of the examination of lossily compressed audio files. This fact is of vital importance in relation to other studies of the author, which are described in [9]. Therefore, having the ability to identify the used MP3 encoder, the method, described in this publication, may be an important support for forensic experts who examine the authenticity of digital audio recordings.

The method discussed in this paper, which is used for detecting the type and version of the MP3 algorithm implementations, can be used for examination of MP3 files, as well as other compression algorithms (e.g. AAC, Ogg Vorbis, etc.), in case of which the current standard does not precisely define each operation performed by the algorithm. It is also worth emphasizing that in relation to applying features of binary data saved in MP3 files for creating the vector, effectiveness of the suggested method is independent to the boundary of spectral characteristics of examined audio recordings. This makes the method independent of the content of the examined recordings, which undoubtedly is the case, among others, with speaker recognition methods based on voice and other solutions based on the analysis and classification of parameters determined on the basis of the audio spectrum. Accordingly, the presented method can be used to examination of MP3 files subjected to compression process, regardless of the audio material it contains.

Bibliography

1. Audio Engineering Society: AES27-1996 (2007) "Recommended Practice for Forensic Purposes – Managing Recorded Audio Materials Intended for Examination", 1996 (2007).
2. Avcibas I., Hicsonmez S., Sencar H.T.:"Audio codec identification from coded and transcoded audios", *Digital Signal Processing*, vol. 23 (5), 2013, 1720-1730.
3. Böhme R., Westfeld A.: "Feature-based encoder classification of compressed audio streams, *Multimedia Systems*, vol. 11 (2), 2005, 108-120.
4. EncSpot – An MP3 Analyzer, 2002, http://www.guerillasoft.nstep.com/EncSpot2
5. Geiger R., Herre J. and Moehrs S.: "Analysing decompressed audio with the Inverse Decoder - towards an operative algorithm", in *Proc. 112th AES Convention*, Munich, May 10-13, 2002.
6. Grigoras C.: "Statistical tools for multimedia forensics", in *Proc. 39th AES International Conference of Audio Forensics*, Hillerød, Denmark, June 17-19, 2010, 27-32.
7. Herre J., Schug M.: "Analysis of Decompressed Audio – The Inverse Decoder", in *Proc. 109th AES Convention*, Los Angeles, USA, Sept. 22-25, 2000.
8. Ji S., Ye J.: "Discriminant Analysis for Dimensionality Reduction: An Overview of Recent Developments", in *Biometrics: Theory, Methods, and Applications*, N. V. Boulgouris, K. N. Plataniotis and E. Micheli-Tzanakou, Eds., John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009, chapter 1, 1-17.
9. Korycki R.: "Authenticity examination of compressed audio recordings using detection of multiple compression and encoders' identification", *Forensic Science International*, vol. 238 (1-3), 2014, 33-46.
10. Korycki R.: "Badanie autentyczności cyfrowych nagrań fonicznych utrwalonych w plikach MP3", *Problemy Kryminalistyki*, vol. 283, 2014.
11. Mathworks R2013a Documentation, Statistics Toolbox and Bioinformatics Toolbox, http://www.mathworks.com/help/documentation-center.html
12. MPEG, "Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, part 3: Audio", International Standard IS 11172-3, ISO/IEC JTC1/SC29 WG11, 1992.
13. Nilsson N.J.: "Introduction to machine learning", Artificial Intelligence Laboratory, Department of Computer Science, Stanford University, 2010.
14. Nilsson M.: "ID3v2 - The Audience is Informed", 1998, http://www.id3.org.
15. Sieler M., Sperschneider R.: "MPEG-Layer3 Bitstream Syntax and Decoding", Wyd. 1.2, 1997.
16. Supreme Court [Poland] judgement of 10 March 1961, ref. III K 49/61, OSNKW, 1962, no. 1, item. 8.

Source

Fig. 1: author, figs. 2–6 [9, 10]
Tab. 1–3: author, 4–7 [9]

Translation Ronald Scott Henderson