International Journal of Innovative Engineering Applications

Journal homepage: https://dergipark.org.tr/ijiea

SPREAD EMBEDDING OF FRAGILE COPYRIGHT INFORMATION TO PROTECT AUDIO SIGNAL

Hüseyin Bilal Macit*[*1]

[1]*Burdur Mehmet Akif Ersoy University, Dept. of Information Systems and Technologies, Bucak ZTYO, Burdur*

Abstract

Original scientific paper

Nowadays, millions of gigabytes of new data are generated every second and a large part of it is multimedia data. The security of this large amount of data is an important problem as well as its transmission and storage. Data without proven authenticity should not be distributed or used without permission. Audio data, unlike other types of multimedia, is quite weak in terms of copyright protection. Industrial practices generally prioritize the quality of audio data over the security of copyright data, contrary to academic recommendations. In this study, a spread hash-supported copyright embedding algorithm is proposed to ensure the copyright protection of audio data. The proposed algorithm is tested on a total of 92.017 seconds of dataset consisting of 516 music files and the results are presented. The algorithm successfully performs copyright verification of any 2-millisecond fragment of the audio in any clipping attack. Despite the changes made to the entire audio data, a 6% Bit Error Rate and 0.9999 Normalized Correlation values are obtained. According to international standards for perceptual evaluation of audio quality, a score of ~-1.7 is obtained in the objective evaluation. All performance evaluations are presented with tables and graphs, and comparisons are made with similar models in the literature. This study is one of the first to use a spread-hashing technique for audio copyright protection and has shown high performance, especially against clipping attacks.

Keywords: *Copyright protection, cyber security, data security, hash function.*

SES SİNYALİNİ KORUMAK İÇİN KIRILGAN TELİF HAKKI BİLGİLERİNİN YAYGIN GÖMÜLMESİ

Özet

Orijinal bilimsel makale

Günümüzde her saniye milyonlarca gigabayt yeni veri üretilmektedir ve bunun büyük bir kısmı multimedya verisidir. Bu büyüklükte verinin iletilmesi ve depolanması kadar güvenliği de önemli bir problemdir. Aidiyeti kanıtlanmamış veri izinsiz dağıtılmamalı ve kullanılmamalıdır. Ses verisi, diğer multimedya türlerinin aksine telif hakkı korunması konusunda oldukça güçsüzdür. Endüstriyel uygulamalar genellikle akademik önerilerin aksine telif hakkı verisinin güvenliğinden ziyade ses verisinin kalitesine önem verir. Bu çalışmada, ses verilerinin telif hakkı güvenliğini sağlamak için hash destekli yaygın bir telif hakkı gömme algoritması önerilmiştir. Önerilen algoritma 516 müzik dosyasından oluşan toplam 92,017 saniyelik bir veri seti üzerinde test edilmiş ve sonuçları sunulmuştur. Algoritma herhangi bir kırpma saldırısında sesin herhangi 2 milisaniyelik parçasından bile telif doğrulamasını başarıyla gerçekleştirmiştir. Tüm ses verisinde yapılan değişikliğe rağmen %6 Bit Hata Oranı ve 0,9999 Normalize Korelasyon değerleri elde edilmiştir. Uluslararası ses kalitesinin algısal değerlendirmesi standartlarına göre nesnel değerlendirmede ~-1,7 skor elde edilmiştir. Tüm performans değerlendirmeleri tablolar ve grafikler ile sunulmuş, literatürdeki benzer modeller ile karşılaştırma yapılmıştır. Bu çalışma, ses sinyalinin telif hakkını korumak için bir yaygın-çırpı tekniği kullanan ilk çalışmalardandır ve özellikle kırpma saldırılarına karşı yüksek performans göstermiştir.

Anahtar Kelimeler: *Çırpı fonksiyonu, telif hakkı koruması, siber güvenlik, veri güvenliği.*

1 Introduction

The amount of data production and distribution has grown tremendously with the widespread use of the Internet in the 21st century. It is estimated that an average of 4.6 million GB of data will be produced per second in 2024 [1], and the vast majority of it will be multimedia data [2]. Ensuring the security of digital media is becoming increasingly difficult as computer networks are extremely susceptible to external attacks [3]. Although unauthorized access to multimedia data is prevented by cryptography [4], once the data is decrypted, it becomes vulnerable again to various attacks such as re-sampling, re-quantization, compression, and echo injection. A multimedia object that

*Corresponding author.

E-mail address: hbmacit@mehmetakif.edu.tr (H. B. Macit)

Received 24 February 2025; Received in revised form 21 April 2025; Accepted 16 June 2025

 Doi: https://doi.org/10.46460/ijiea.1645813

appears to be vulnerable can be made to protect itself in an undetectable manner. For this purpose, data hiding methods such as labeling, digital signing, digital watermarking, and steganography methods [5] based on ancient times have been adapted to the 21st century digital world. In addition to these methods, the security of multimedia is also protected by laws. For example, according to the laws of the United States; the developer of a story, picture, song or any other original work automatically owns the copyright from the moment this work is recorded in physical form [6]. However, if the authors wish to distribute their work, they must add a copyright notice to the work. In Turkey, Article 22 of the Law on Intellectual and Artistic Works No. 5846 clearly states the rights of the author: "The right to reproduce the original or copies of a work, in whole or in part, directly or indirectly, temporarily or permanently, by any means or method, belongs exclusively to the author. Making a second copy of the original work or recording the work on any known or future means that are used for signal, sound and image transmission and repetition, any sound and music recordings, and the implementation of plans, projects and sketches of architectural works are also considered copies.". Despite the laws, copyright violations are increasing, especially with the spread of mobile devices. Manufacturers are developing new algorithms for copyright protection. Most of these methods add an imperceptible piece of information representing the producer to the digital multimedia object [7] as shown in figure 1.

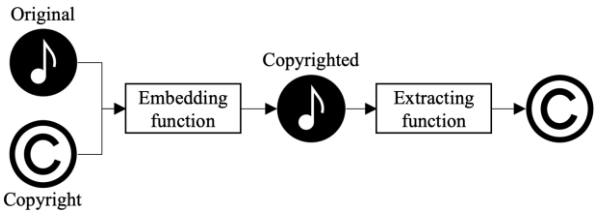

Figure 1. Overview of the basic multimedia copyright protection scheme.

However, the biggest risk of these methods is that the person who made the unauthorized copy is able to cut the copyright embedded part of the object. In this case, the person who purchased the multimedia object may not be able to decide whether it is the original or a pirate copy. In today's digital world, the best way to check copyright information is to consult a copyright database, which is a collection of copyrighted works and related information such as authors, publishers, publication dates, and copyrights. The General Directorate of Copyrights of the Ministry of Culture and Tourism of Turkey has provided access to the Database of Intellectual and Artistic Works via eser.telifhaklari.gov.tr [8]. This includes information such as registration and banderol information regarding processed works, repertoire records of some music workers' associations, names of works, authors, publishers and producers, years of publication and production. The performance of a copyright protection algorithm is calculated according to four criteria: Imperceptibility, robustness, payload and low computational time. Imperceptibility refers to the similarity of the original multimedia with copyright information added version [9]. A copyrighted audio data should not be distinguishable by the human auditory system. The ability of copyrighted multimedia to protect copyright during transfer and storage is called robustness. A malicious distributor can perform various transformations on the digital multimedia object to destroy copyright data [10]. In some applications, multimedia is expected to retain most of the copyright data despite these transformations. Payload is defined as the data embedding capacity of the algorithm and is measured as the number of bits embedded in one second of the audio signal (bps) [11]. The ideal amount of data payload is 1kbs for every 1kHz [12]. An algorithm cannot be expected to satisfy all four criteria at the same time [13]. Depending on the type of multimedia object, the distribution/storage medium and the distributor's demands, it can be decided which criterion is the priority. To date, many copyright protection algorithms have been proposed and used for audio. The biggest problem of most of them is the quality loss that may occur in the original audio data [14].

In this paper, a copyright protection method with an ideal payload of 1 bit per Hertz is proposed, which is almost imperceptible for the Human Auditory System (HAS). The Royalty-Free Audio (RFA) Dataset [15] downloaded from kaggle.com website was used to implement the method and test its results. The reasons for choosing this experimental dataset in this study are that it contains freely distributed sounds, all files have detailed copyright information, and they have a wide variety of file lengths. The dataset contains a total of 92,017 seconds of audio data sampled from Youtube Royalty-free videos. There are 516 music files, and a table including copyright texts, ranging from 104 characters to 386 characters. The shortest music file is 68 seconds, and the longest one is 1792 seconds. The proposed method is implemented with all elements in this dataset and the performance results are shown mathematically with various metrics, and visually with graphs.

2 Method

The proposed method consists of two phases: Embedding copyright information and extracting it. The schematic of the method is shown in figure 2.

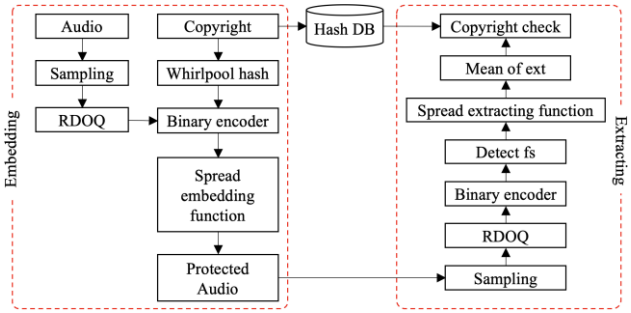

Figure 2. Overall schematic of the proposed method.

2.1 Copyright Embedding Phase

The number of sound samples recorded in 1 second of an audio signal is called sampling rate and is measured in Hertz or samples/sec. For example, a 44.1kHz audio signal has 44100 digital sound samples per second. In the first step, the audio signal is sampled and the continuous-time

signal is reduced to a discrete-time signal. In other words, the analog signal is converted into a series of "samples". Each sample is the value of the signal in time space. The sample set is obtained by sampling the audio signal s(t) and multiplying it by the impulse sequence $\delta(t)$:

$$\delta(\mathrm{t}) = [\frac{1}{T_s} + \sum_{n=1}^{\infty} \frac{2}{T_s} cosn\omega_s t] \quad (1)$$

Here, ω_s is the value of each sample, T_s is the sample time, and n is the index number of corresponding samples. Sampled audio $y(t)$ is calculated in equation (2):

$$\begin{aligned} y(t) &= s(t).\delta(\mathrm{t}) \quad (2)\\ &= s(t).[\frac{1}{T_s} + \sum_{n=1}^{\infty} \frac{2}{T_s} cosn\omega_s t] \\ &= \frac{1}{T_s}[s(t) + 2cos\omega_s t.s(t) + 2cos2\omega_s t.s(t) + \cdots] \end{aligned}$$

Fourier Transform is performed on both sides of $y(t)$:

$$\begin{aligned} y(\omega) &= \frac{1}{T_s}[s(\omega) + s(\omega - \omega_s) + s(\omega + \omega_s) + s(\omega - 2\omega_s) + \cdots] \\ &= \frac{1}{T_s}\sum_{n=-\infty}^{\infty} s(\omega - n\omega_s) \quad (3) \end{aligned}$$

$y(\omega)$ is the sample set of s. Quantization process is performed to convert the amplitude values indicated by the sample set $y(\omega)$ into a numerical sequence. Thus, a finite number of output values are obtained from a continuous set. In this study, the quantization process is coded by 8 bits regardless of the size of the input data. During coding, a sequence of numbers called quantization noise is produced, which is sometimes modeled as an additive random signal called quantization noise. After adding copyright protection, Rate Distortion Optimized Quantization (RDOQ) was applied as a solution method for the problem of "minimum number of bits per symbol" [16] to reconstruct the same s signal. Thus, it was aimed to find the most appropriate set of transformation coefficients [17]. Because, RDOQ finds the optimal quantized level of each transform coefficient by minimizing the rate and distortion of s. The scale factor f_s is used to transform the resulting decimal number elements into their corresponding integers.

$$f_s = argmin|s - S|^2 \quad (4)$$

Here, S is the corresponding 8-bit transformed element of s. So, $S(t)$ is the optimized quantized array. Copyright information detects any unauthorized change or tampering, ensures the verification of the audio source, and ensures data integrity. Thus, the intellectual property rights of content creators are protected [18]. However, there is no global standardization for copyright information. Therefore, the copyright of each work is of different length and formation. The RFA Dataset used in this study contains 516 music files with copyright text varying from 104 characters to 386 characters. The shortest music file is 68 seconds and the longest one is 1792 seconds long, as shown in Table 1.

Table 1. Copyright information of the shortest and longest music in the RFA Dataset.

Audio	Copyright data	Character count
Far Away – MK2	Far Away - MK2 | Royalty Free Music - No Copyright Music | YouTube Music, No license provided/CC0 License	104
Enchanted Valley - Kevin MacLeod	Dnchanted Valley - Kevin MacLeod | Royalty Free Music - No Copyright Music | YouTube Music, Enchanted Valley by Kevin MacLeod is licensed under a Creative Commons Attribution license(https://creativecommons.org/licenses/by/4.0/)Source: http://incompetech.com/music/royalty-free/index.html?isrc=USUAN1200093Artist: http://incompetech.com/Support by RFM - NCM: https://youtu.be/RC4W3GDGZMg	386

Variable copyright text length makes it difficult to hide copyright information in audio data with a standard model. Many studies hide copyright information from the first sample of audio data. However, in this case, copyright information is lost by trimming the first seconds of the audio. When reading copyright data, the data length must be known in advance, which creates a standard data length requirement for adding copyright. In the proposed method, a hash function is used, and the copyright information is spread over the entire audio data to avoid these problems. A hash function maps data of different dimensions to fixed-size and irreversible values [19]. The most commonly used hash algorithms today are MD5 and SHA variants [20]. SHA-256 and SHA3-256 both have an output size of 256 bits (32 bytes) whereas MD5 has an output size of 128 bits (16 bytes). In this study, The Whirlpool Secure Hash Function [21], which is the hash algorithm with the longest output (512 bits) that can be used with SHA [22], I s used. Let's assume that C_o is the initial copyright text, and C_h is the encrypted hash data.

$$C_{int} = W(C_{i-1}, m_i) \oplus C_{i-1} \oplus m_i \quad (5)$$

C_{int} is the intermediate value where $m_1, m_2, \ldots, m_t$ are the time-based blocks of C_o. Each C_i is the value of iteration i, and C_{i-1} is the previous value. W is the Whirlpool block cipher function, which operates similarly with AES. Figure 3 shows the general structure of W.

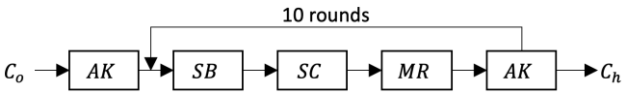

Figure 3. Whirlpool Cipher

C_o is transformed to 8x8 matrix as an input data to the first AK function. This input is called $CState$. The 8x8 matrix produced as a result of each AK function is called $KState$. In short, the first AK input is called $CState$. Let r represent the round number:

$$CState = KState_r, \text{ where } r = 1 \quad (6)$$

$KState$ is the input key for AK when $2 \leq r \leq 10$. A 16x16 Substitution box (S-box) table is used for the SB function. This table contains all possible 8-bit values. S-box is used

for nonlinear mapping of $CState$ by taking four leftmost bits and place them as the column indexes:

$$KState_r = SB(KState_{r-1}), b_{i,j} = S[a_{i,j}] \quad (7)$$

Here, $b_{i,j}$ is the value of the S-box, i,j represents the individual byte of CState. SC function:

$$KState_r = SC(KState_{r-1}) \leftrightarrow b_{i,j} = a_{(i-j)mod8,j} \quad (8)$$
$$0 < i,j < 7, 2 \leq r \leq 10$$

The MR function is a linear diffusion layer. Diffusion is a cryptographic property [23] that hides the statistical properties of the input key. For this purpose, it uses a standard transformation matrix M.

$$M = \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix} \quad (9)$$

For example, $KState_i$, which is obtained in round i is:

$$KState_i = KState_{r-1}.M \quad (10)$$

The AK function XORs the bits of the round associated with the $KState$. The AK function:

$$KState_{r-1} = AK[Key_i](A) \leftrightarrow b_{i,j} = a_{i,j} \oplus k_{i,j}, 0 \leq i,j \leq 7 \quad (11)$$

Here, Key_i is the round key, and round Constant rc is calculated to produce it in the corresponding round:

$$rc[r]_{0,j} = Sbox[8(r-1)+j] \quad (12)$$
$$0 \leq j \leq 7, 1 \leq r \leq 10$$

$$rc[r]_{i,j} = 0, 1 \leq i \leq 7, 1 \leq r \leq 10 \quad (13)$$

Key_i is calculated by obtained rc:

$$Key_r = RF[rc[r]](Key_{r-1}) \quad (14)$$

RF function in any r round using these values:

$$RF(K_r) = AK[K_r] \odot MR \odot SC \odot SB \quad (15)$$

Here, the operator $\odot$ indicates the iteration of the composition function with index r, and running from 1 through 10. The entire W function can be summarized as in equation 16:

$$W(K) = \left(O_{r=1}^{10} RF(Key_r)\right) \odot AK(Key_0) \quad (16)$$

The 128 character C_h array obtained with the C_0 copyright input to the Whirlpool hashing function is a hexadecimal array. However, $S(t)$ obtained with RDOQ is a binary array. To embed the hash data into $S(t)$, it is also encoded in binary form. Binary encoder converts C_h to 1-dimension binary array which contain 512 elements consisting of 1s and 0s.

$$C_h(t) = \{C_h(1), C_h(2), \ldots, C_h(512)\} \quad (17)$$

Least Significant Bit (LSB) modification was applied for copyright embedding, which is popular for data hiding due to its simplicity and readability [24]. LSB preserves the audio quality after copyright embedding, but is not robust against attacks such as noise and clipping [25]. LSB modification basically replaces the last bit of each 8-bit sample of $S(t)$ with the last bit of $C_h(t)$. So, LSB modification can change the numeric value of the corresponding sample of $S(t)$ by a maximum of ±1.

$$S(t)_i = \begin{cases} 1, C_h(t)_i = 1 \\ 0, C_h(t)_i = 0 \end{cases} \quad (18)$$

Most of the traditional methods hide the copyright information into the audio header. However, some methods hide it from the first second of the audio data. In this case, if a part of the first seconds of the audio is cut, copyright is lost. The proposed method divides $S(t)$ into equal parts of 512-bit size and embeds $C_h(t)$ into the LSBs of each part of it as shown in figure 4. Thus, even if any random part of the audio is clipped, copyright information can still be accessed.

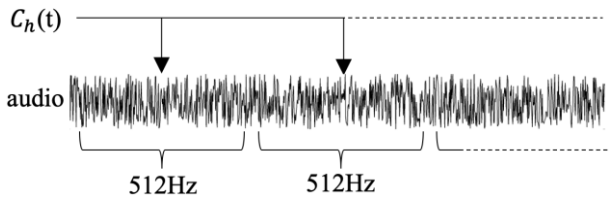

Figure 4. Spread embedding of binary encoded hash data into the audio.

The generated copyright protected signal is a discrete-time signal. It should be formed with the same sample rate of the original input signal. For example, if the original signal was 44.1kHz, the copyright embedded signal is also formed back as 44.1kHz.

2.2 Copyright Extracting Phase

This phase is the stage of reading the encrypted copyright information in the copyrighted audio. The main purpose is to compare the information extracted from the copyrighted audio with the encrypted copyright data in the hash database. In the first step, the copyrighted audio $S(t)$ is sampled using the method in the copyright embedding phase to obtain $y(\omega)$ samples. RDOQ is applied to convert the amplitude values indicated by the $y(\omega)$ sample set into a numerical array. Thus, a protected audio sequence $P(t)$ consisting of t samples is obtained. The possible problem here is that the audio may have been subjected to a clipping attack. In this case, it is not possible to detect which sample the copyright data starts from, but the proposed method offers a solution to this problem. $P(t)$ is divided into 512-bit pieces starting from the first sample and the $\forall(m, 512)$ matrix is created, with each piece being a row:

$$P(t) \to \forall(m,512) = \begin{bmatrix} \forall_{1,1} & \cdots & \forall_{1,512} \\ \vdots & \ddots & \vdots \\ \forall_{m,1} & \cdots & \forall_{m,512} \end{bmatrix} \quad (19)$$

$$= \begin{bmatrix} P(t)_i & \cdots & P(t)_{512+i} \\ \vdots & \ddots & \vdots \\ P(t)_{512m+i} & \cdots & P(t)_{513m-1} \end{bmatrix}, 1 \le i < 512$$

If the copyright information is placed starting from the ith sample of $P(t)$, each row of the $\forall$ must be identical or largely similar to each other. To evaluate this, each column is selected in turn and the standard deviation of the selected column is taken as $\sigma\forall(i)$:

$$\sigma\forall(i) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\forall(j) - \overline{\forall})^2} \quad (20)$$

The column with the smallest standard deviation is actually considered to be the starting point $pos(C_e(t))$ of the section where all the rows are most similar to each other, which theoretically contains the new copyright information C_e.

$$pos(C_e(t)) = \min(\sigma\forall) \quad (21)$$

Starting from $pos(C_e(t))$, the average of 512 columns is taken one by one and the extracted copyright C_e is obtained.

$$C_e(i) = \frac{\sum_{j=1}^{m} \forall_{i,j}}{512}, 1 \le i < 512 \quad (22)$$

C_e is compared with the data in the hash database to check the ownership of the sound. Here, the similarity between the C_h retrieved from the database and the extracted C_e is evaluated. The desired similarity rate can be determined by experimental studies or by the threshold value determined by the user using the algorithm. Since both vectors are of the same type of data, Bit Error Rate (BER) and Two-dimensional correlation analysis (CC) were calculated to determine their experimental similarity rate. BER is a bitwise calculation metric which is used to measure the number of bits that change between two signals. As the BER value decreases, the probability of C_e being copyrighted increases.

$$BER = \frac{C_e \oplus C_h}{512} \quad (23)$$

Here, the $\oplus$ operator indicates the number of bits that differ between two vectors. CC analysis is a mathematical technique that shows the amount of change between two signals. CC is also referred to as covariance or correlation in classical mathematics. Let $\tilde{d}(C)$ be the reference difference between C_e and C_h:

$$\tilde{d}(C) = C_e - C_h \quad (24)$$

Synchronous spectrum is calculated in:

$$\phi(C_e, C_h) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\tilde{d}(C) - C_e\right)\left(\tilde{d}(C) - C_h\right) = \frac{1}{511}\sum_{i=1}^{512}\left(\tilde{d}(C) - C_e\right)\left(\tilde{d}(C) - C_h\right) \quad (25)$$

Here, ϕ is the CC value and as it approaches to 1, the similarity between C_e and $C_h$increases. Therefore, low BER and high CC results were targeted in the test results of the proposed algorithm.

3 Experimental Results

The RFA Dataset was used to test the performance of the proposed method. The method was run on all 516 music files in the dataset and all statistical results obtained were shown with tables and graphs. Avalanche effect test was performed with Dataset inputs to measure the encryption performance of the hash algorithm. Segmental Signal to Noise Ratio (SSNR) test was performed to mathematically measure the quality of the copyrighted audio produced. Random Simulation was performed to monitor the amount of noise caused by the method. Additionally, Objective Difference Grade (ODG), which is an international sound quality measurement standard, was measured.

A hash function should be able to change at least 50% of the output data even with a 1-bit change in the input data [26]. This feature is called Avalanche Effect [27] and is calculated in the proposed implementation as in Equation (26).

$$Avalanche\ Effect = \frac{Modified\ bits\ (or\ hex')\ of\ hash}{Total\ bits\ (or\ hex')\ of\ copyright} \quad (26)$$

Each character of the hash text is 1 hex long. Therefore, the avalanche effect test was performed both bitwise and hexwise. Copyright data character length and avalanche effect graphs for a total of 516 audio files in the RFA Dataset are shown in figures 5 and 6.

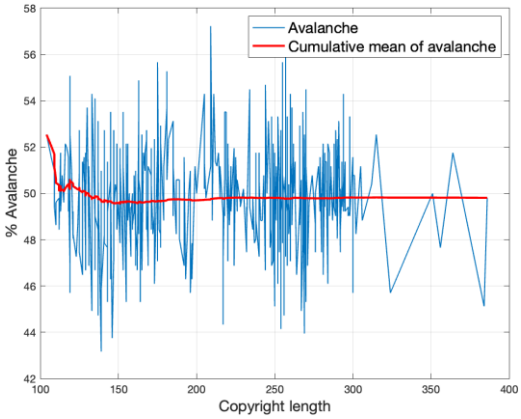

Figure 5. Avalanche effect percentage of 1-bit modification.

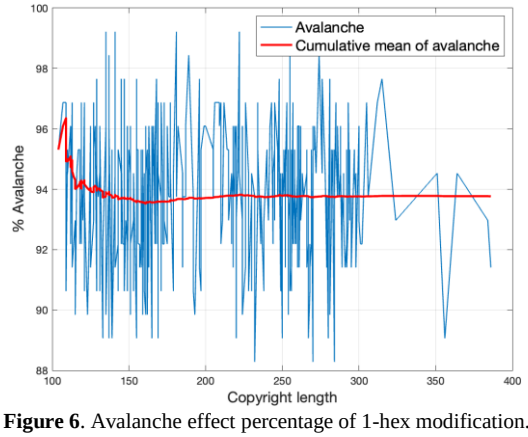

Figure 6. Avalanche effect percentage of 1-hex modification.

As seen in the graphs in figures 5 and 6, there is no relationship between Copyright Length (CL) and avalanche effect. Table 2 shows the maximum, minimum and average results of bitwise and hexwise avalanche tests performed on RFA Dataset.

Table 2. Avalanche test results of RFA dataset.

	1-bit modification	1-hex modification
Min percentage	43.16%	88.28%
Min count	221	113
Avg percentage	49.81%	93.77%
Avg count	255.02	120.02
Max percentage	57.23%	99.22%
Max count	293	127

It is clearly seen in the table that the hash data changes by approximately 50% on average in 1-bit modification and by approximately 95% on average in 1-hex modification. In other words, even if 50% of the C_e is similar to the C_h, this audio data is likely to be copyrighted.

SSNR is one of the widely used objective methods for measuring sound quality [29]. To calculate SSNR, copyright embedded signal is first divided into m segments with n samples each. Then SSNR is calculated in Equation (28):

$$SSNR = 10log_{10}\left\{\frac{1}{m}.\sum_{i=1}^{m}\left(\frac{\sum_{j=1}^{n}[x_i(j)]^2}{\sum_{j=1}^{n}[x_i(j)-s_i(j)]^2}\right)\right\} \quad (28)$$

Here, $x(i)$ and $s(j)$ are the original and copyright embedded signals, respectively. $\sum[x_i(j)-s_i(j)]^2$ is the noise power, which refers to mathematical difference of two signals. As shown in figure 7, as the SSNR value increases in the positive direction, the mathematical similarity between the original sound and the copyrighted sound increases, and as it decreases in the negative direction, the similarity decreases.

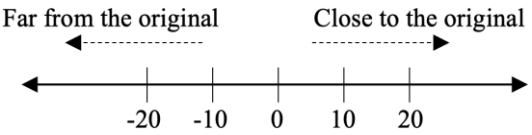

Figure 7. SSNR Scale [29].

In some special cases, such as phase encoding, SSNR measurement is meaningless. Because the waveform of the embedded signal changes a lot due to the phase change, SSNR is underestimated. The proposed method is extremely suitable for SSNR measurement, because it applies LSB modification.

In the early years of digital technology, there were no International Standards for measuring sound quality. Quality measurement was done with listening tests relying on human perception. The first methods for testing telephone band speech signals were standardized within ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) Recommendation P.800 in 1993 [30]. Between 1994 and 1998, the Perceptual Evaluation of Audio Quality (PEAQ) method was proposed to objectively measure perceived sound quality [31] and this method was accepted as a standard. PEAQ simulates the human ear's perceptual properties. The model uses a metric called Subjective Difference Grade (SDG). This metric measures the distance between two sound signals and produces a reference score. The SDG score and the ODG score are produced as shown in table 3.

Table 3. PEAQ Scoring [9].

Audio quality	SDG	ODG
Imperceptible	5	0
Perceptible, but not annoying	4	-1.0
Slightly annoying	3	-2.0
Annoying	2	-3.0
Very annoying	1	-4.0

Copyright embedding was performed with the proposed method on all sounds in the RFA Dataset and the maximum, minimum, and average mathematical and perceptible high precision results obtained are shown in table 4.

Table 4. Performance test results of the proposed method.

	Worst	Average	Best
SSNR	12.12885955	23.97756593	33.05281978
BER	0.061658518	0.062416802	0.062965252
CC	0.998574061111926	0.99990356	0.999971083931624
ODG	-3.9049	-3.4719	-1.7073

The proposed method was applied to random audio in the RFA Dataset to simulate the amount of noise it causes. To show the amount of noise, the amplitude versus time graph of a small section of the original audio and the copyrighted audio were plotted over each other, as shown in figure 8.

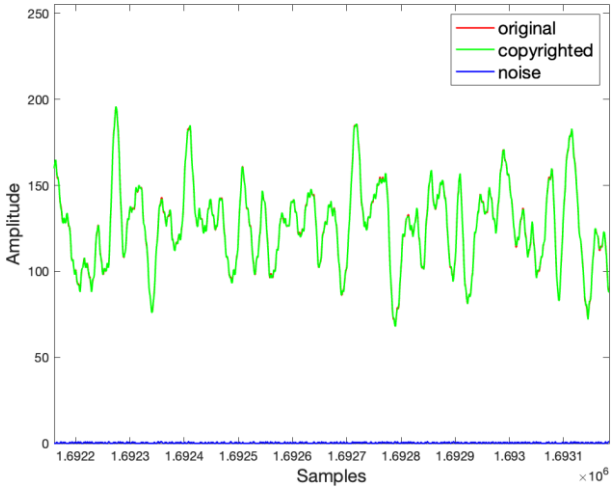

Figure 8. Noise graph of the proposed method on a random section of a random audio from the dataset.

The signal shown in the graph is approximately 2 milliseconds random piece of a random sound in the RFA dataset. The original signal is plotted in red, and the embedded signal is in green. The amount of difference is also plotted in blue. As can be seen, the perceptual difference between the signals is quite low, which can also be shown with Spectrograms. Spectrograms are the visual representations of audio, which are very detailed and accurate images that have been widely used in audio classification tasks [32]. Very similar sounds can be distinguished by a spectrogram. A spectrogram is typically produced using a short-time Fourier transform with a fixed window size, the square of which gives the magnitude of

the spectrogram [33]. The y-axis of the spectrogram simply represents the time, the x-axis represents the frequency, and the color of each point represents the amplitude of that point. The spectrogram of the audio whose amplitude/time graph is given in figure 8 is shown in figures 9 and 10.

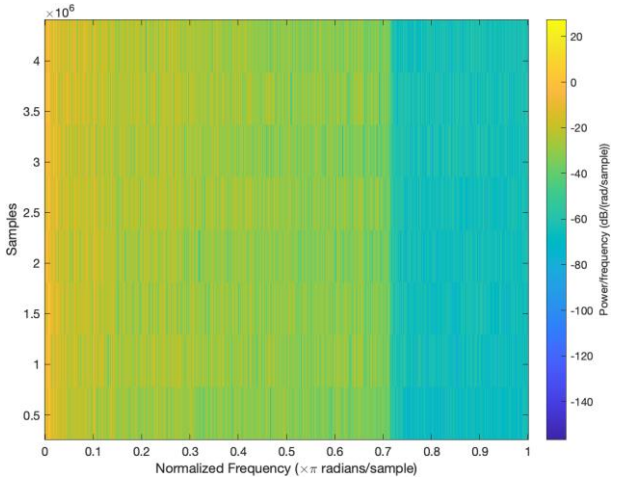

Figure 9. Spectrogram of the original audio.

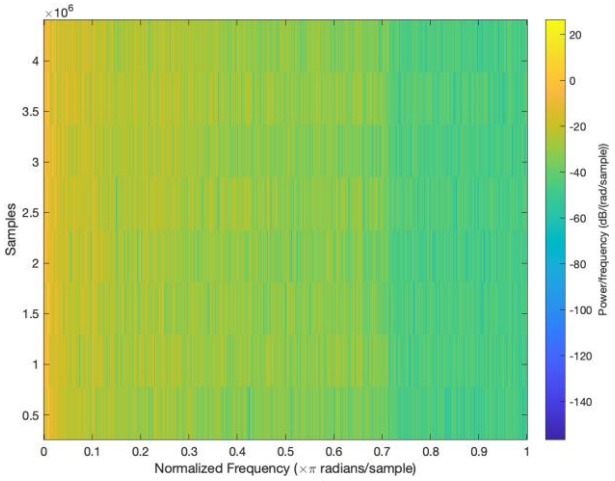

Figure 10. Spectrogram of the copyrighted audio.

The proposed method was performed on 516 sound files in the RFA dataset. The sound files that have minimum and maximum features and that are produced maximum and minimum results in mathematical measurements and the results obtained from them are shown in table 5.

Table 5. Audio files with marginal features and results in RSA Dataset.

Feature	Name	Duration (sec)	CL	BER	CC	SSNR	ODG
Min duration	Prelude No 1 - Chris Zabriskie	68	351	0,062243754	0,999038298	23,47209493	-2,615307505
Max duration	Top 10 Songs Of Ikson	1792	187	0,06247304	0,999936699	22,26100438	-2,43847257
Min CL	Far Away - MK2	105	104	0,062380556	0,999944469	21,21885178	-2,746567625
Max CL	Enchanted Valley - Kevin MacLeod	190	386	0,062425941	0,999631367	28,51836632	-3,643323121
Min BER	Lilac Skies - Corbyn Kites	109	283	0,061658518	0,999959743	16,89904238	-3,674572915
Max BER	Hot Coffee - Patrick Patrikios	194	153	0,062965252	0,999944051	21,42182483	-3,869273264
Min CC	Shattered Paths - Aakash Gandhi	178	155	0,062441933	0,998574061	20,91147021	-3,597474417
Max CC	First Of The Last - Silent Partner	127	177	0,062546669	0,999971084	12,12885955	-3,811454725
Min SSNR	First Of The Last - Silent Partner	127	177	0,062546669	0,999971084	12,12885955	-3,811454725
Max SSNR	Take Your Pick - Aaron Lieberman	109	157	0,062367196	0,999886119	33,05281978	-3,573908485
Min ODG	Forgiveness - Patrick Patrikios	203	155	0,062533817	0,999909641	30,64274854	-3,904917952
Max ODG	Mirror Mirror - Diamond Ortiz	185	145	0,062443396	0,999948195	22,49870419	-1,707289275

The shortest sound duration in the RFA Dataset is 68 seconds, and the longest one is 1792 seconds. The mathematical results obtained in the proposed method are expressed according to the sound duration in figures 11 to 15.

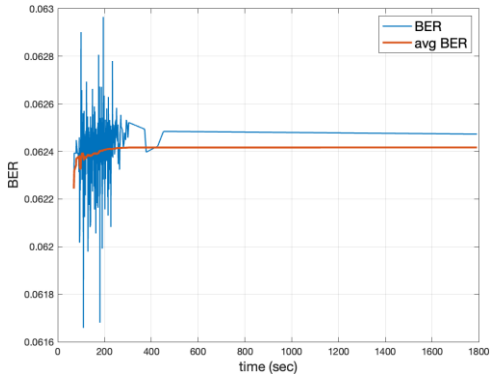

Figure 11. BER-duration.

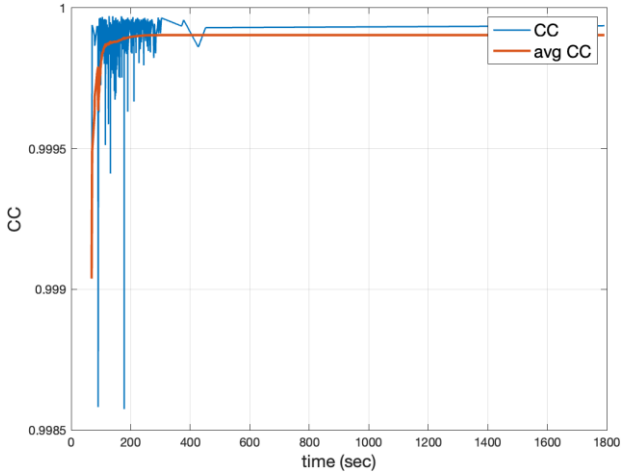

Figure 12. CC-duration.

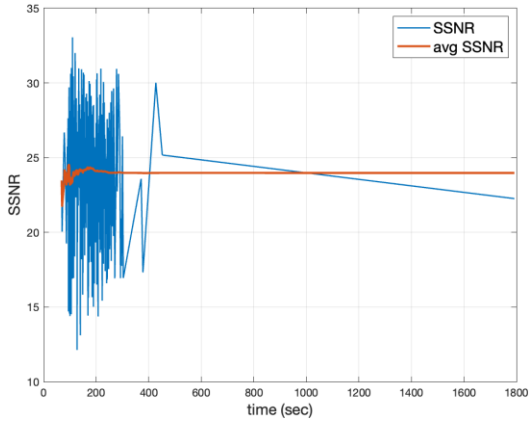

Figure 13. SSNR-duration.

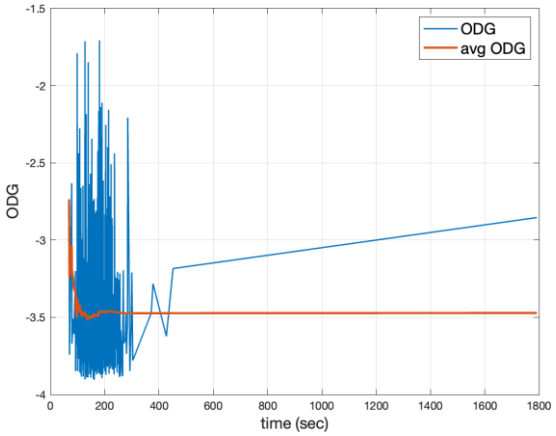

Figure 14. ODG-duration.

While the mathematical measurements are far apart for shorter audio files, the longer ones appear to be quite close to the cumulative average. However, it is not possible to infer that the audio file duration directly affects the mathematical scores in the proposed method. There are many methods of embedding some kinds of data into audio files in the literature. Many of them have been tested on different data sets. Since there is no standardized metric for testing these methods, each has performed tests with its own chosen metric. Therefore, it is not possible to directly compare the proposed method in this study with the literature. However, a state-of-the-art comparison is shown in table 6, based on all metrics between the best and worst results declared. Another important point to note that almost all of these studies are not spread. Therefore, they cause interference only in a limited area of the audio. This also makes it difficult to compare the methods with each other. Moreover, as it is obvious, no method can be successful in all criteria at the same time. In order to approach an objective comparison result, the best and worst experimental results of the studies are shown in table 6. However, each study used different mathematical performance metrics to express experimental results. When the best and worst results are averaged, the proposed method is ranked second in three studies that give BER scores, and second in five studies that give CC scores. The obtained SSNR score was compared with the average of the state-of-the-art Peak Signal to Noise Ratio (PSNR) score, the proposed method ranked first in eight studies. Both methods express perceptual performance in signal processing methods. The proposed method is ranked third in three studies in the ODG standard. However, all of these methods do not have as much bit density as the proposed method because they do not spread copyright data to whole signal. The main goal of the proposed method is to obtain copyright information even when a large part of the audio data is cropped.

Table 6. State-of-art comparison.

Paper	Method	Value	Metric				
			BER	CC	PSNR	SSNR	ODG
Proposed	-	Best	0.061658	0.99997	NA	33.0528	-1.7073
		Worst	0.062965	0.99857	NA	12.1288	-3.9049
[18]	DCT	Best	NA	NA	41.5638	NA	NA
		Worst	NA	NA	3.21	NA	NA
[9]	SVD-DWT based	Best	NA	NA	39.02	NA	-0.67
		Worst	NA	NA	37.5	NA	-0.91
[12]	M16M	Best	NA	1.0000	72.0019	NA	NA
		Worst	NA	0.9743	37.3739	NA	NA
[2]	Iterative Filtering	Best	NA	0.9999	40.05	NA	NA
		Worst	NA	0.8698	NA	NA	NA
[3]	FFT	Best	NA	0.9999	37.98	NA	NA
		Worst	NA	0.9995	35.78	NA	NA
[34]	M-SW-LSC	Best	0.0035	0.9931	37.8132	NA	-0.53
		Worst	0.0257	0.9557	37.8113	NA	-1.9599
[35]	LPC	Best	0.0000	NA	39	NA	-1.02
		Worst	8.76	NA	33	NA	-3.68

4 Conclusion

Digital audio protection methods have been applied to industry applications since the early development stage in late 1990s [36]. Most of them were the incorporation and modifications of existing techniques from other research areas, e.g., spread spectrum from communication theory [37], and patchwork methods from image watermarking [38]. Most of the proposed methods are based on signal processing techniques. These methods are generally classified whether copyright data is placed in the time domain or the frequency domain. Real time industrial methods differ from academic solutions. They consider more on imperceptibility than robustness. The reason is that each industry solution defines a specific application, in which the attacks may not need to be exhausted [36]. This study implemented an irreversible copyright data using a hash algorithm for audio security. An unlimited

sized copyright data has been reduced to a fixed size and spread from the first to the last bit of the audio signal. Thus, the clipping attack, which is the easiest to apply by attackers, has been rendered ineffective. This feature shows that proposed method can be applicable in the industry.

According to the results of the avalanche effect tests, the fact that even 50% of the copyright data can be extracted has shown that the originality of the sound is guaranteed. This has shown that preserving even half of the LSBs in any interfered sound is sufficient to prove the ownership. The proposed method works independently of the length of the copyright text. The mathematical performance of the method was measured with SSNR, BER, CC, and ODG measurements made on the entire dataset. The results are shown as maximum, minimum, and average values. In addition, all other scores of the sound data that produced a marginal score in the dataset are also shown. In order to see the relationship between the mathematical performance of the method and the duration of the audio, duration-performance analysis graphs are presented. Accordingly, the length of the audio is not a constraint for the method, that is, the method produced consistent results for all lengths of audio in the dataset. The obtained results show the applicability of the proposed method in the real world. The spread embedding ability of the method and the ability to detect the starting point of hashed copyright information show that it is extremely robust against all clipping attacks.

Decleration

Ethics committee approval is not required.

References

[1] Duarte, F. (2024). Amount of data created Daily. Retrieved September 07, 2024 from https://explodingtopics.com/blog/data-generated-per-day

[2] Naqash, K. I., Malik, S. A., & Parah, S. A. (2024). Robust audio watermarking based on iterative filtering. Circuits *Syst Signal Process*, (43), 348–367.

[3] Salah, E., Amine, K., Redouane, K., & Fares, K. (2021). A fourier transform based audio watermarking algorithm. *Appl. Acoust*, (172).

[4] Kwon, O. J., Choi, S., & Lee, B. (2018). A watermark-based scheme for authenticating JPEG image integrity. *IEEE Access*, (6), 46194–46205.

[5] Yalman, Y., & Ertürk, İ. (2009). Gerçek zamanlı video kayıtlarına veri gizleme uygulaması. *XI. Akademik Bilişim Konferansı Bildirileri/Harran University, Şanlıurfa,* (pp. 545-552).

[6] Macit, H. B. (2019). Mobil cihaz görüntüleri için entropi tabanlı kırılgan damgalama metodu geliştirilmesi (Publication No. 566003) [*Doctoral dissertation, Süleyman Demirel University, Isparta*].

[7] Arnold, M., Schmucker, M., & Wolthusen, S. D. (2003). Techniques and applications of digital watermarking and content protection. *Artech House. London*.

[8] Arseven, M. (2019). Turkey provides access to copyright database. *MA Gazette Edition*, (78).

[9] Al-Haj, A. (2014). An imperceptible and robust audio watermarking algorithm. *J Audio Speech Music Proc.*, (37).

[10] Wu, C. P., Su, P. C., & Kuo, C.-C. J. (2000). Robust and efficient digital audio watermarking using audio content analysis. *Proceedings of SPIE 12th International Symposium Electronic Imaging, San Jose, CA*, (pp. 382-392).

[11] Tseng, H. W., & Leng, H. S. (2014). High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion. *EURASIP Journal on Audio, Speech, and Music Processing*, (8)11, 647-654.

[12] Bhattacharyya, S., Kundu, A., & Sanyal, G. (2011). A novel audio steganography technique by M16MA. *International Journal of Computer Applications*, (30)8, 26-34.

[13] Acevedo, A. (2003). Digital watermarking for audio data in techniques and applications of digital watermarking and content protection. *Artech House, USA*. (75–114).

[14] Rachid, R. S. (2014). Binary image watermarking on audio signal using wavelet transform (Publication No. 386090) [*Master's thesis, Çankaya University, Ankara*]

[15] Deshpande, D. (2021). Royalty-Free Audio Dataset. Retrieved September 02, 2024 from https://www.kaggle.com/datasets/darshan1504/royaltyfree-audio-dataset

[16] Blau, Y., & Michaeli, T. (2019). Rethinking lossy compression: The rate-distortion-perception tradeoff. *Proceedings of the International Conference on Machine Learning*, (pp. 675–685).

[17] Stankowski, J., Korzeniewski, C., Domański M., & Grajek, T. (2015). Rate-distortion optimized quantization in HEVC: Performance limitations. *2015 Picture Coding Symposium (PCS)/Cairns, QLD, Australia*, (pp. 85-89).

[18] Al-Darrat, K., & Abushaala, A. (2024). Copyright protection based on hybrid image watermark (DCT) with audio watermark (EMD). *The International Journal of Engineering & Information Technology (IJEIT)*, 12(1), (pp. 84–89).

[19] Aggarwal, K., & Verma, H. K. (2015). Hash_RC6 — Variable length Hash algorithm using RC6. *Proceedings of 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India*, (pp. 450-456).

[20] Pittalia, P. P. (2019). A comparative study of hash algorithms in cryptography. *International Journal of Computer Science and Mobile Computing*, (8)6, 147-152.

[21] Barreto, P., & Rijmen, V. (2003). The Whirlpool hashing function, *First open NESSIE Workshop, Leuven, Belgium*, (pp. 13- 14).

[22] Stallings, W. (2006). The Whirlpool secure hash function. *Cryptologia*, (30), (pp. 55–67).

[23] Shannon, C. (1949). Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28 (4), (pp. 656–715).

[24] Kashifa, S., Tangeda, S., Sree, U. K., & Manikandan, V. M. (2023). Digital image watermarking and its applications: A detailed review. *Proceedings of IEEE Int. Students' Conf. Electr., Electron. Comput. Sci. (SCEECS)*, (pp. 1–7).

[25] Ramyashree, Venugopala, P. S., Raghavendra, S., & Ashwini, B. (2024). CrypticCare: A strategic approach to telemedicine security using LSB and DCT steganography for enhancing the patient data protection. *IEEE Access*, (12), (pp. 101166 – 101183).

[26] Chi, L., & Zhu, X. (2018). Hashing techniques: A survey and taxonomy. *ACM Comput. Surveys*, (50)1, (pp. 1–36).

[27] Upadhyay, D., Gaikwad, N., Zaman, M., & Sampalli, S. (2022). Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications. *IEEE Access*, (10), (pp. 112472-112486).

[28] Prodeus, A. (2015). Reducing sensitivity of segmental signal-to-noise ratio estimator to time-alignment error. *International Journal of Electrical and Electronic Science*, 2(2), (pp. 31-36).

[29] Alsaad, S., & Hashim, E. (2013). A speech scrambler algorithm based on chaotic system. *Al- Mustansiriyah J. Sci.*, (24), 357-372.

[30] Salovarda, M., Bolkovac, I., & Domitrovic, H. (2005). Estimating perceptual audio system quality using PEAQ algorithm, *18th International Conference on Applied Electromagnetics and Communications*, Dubrovnik, Croatia, 1-4

[31] Thiede, T., Treurniet, W. C., Bitto, R., Schmidmer, C., Sporer, T., Beerends, J. G., Colomes, C., Keyhl, M., Stoll, G., Brandenburg, K., & Feiten, B. (2000). PEAQ the ITU standard for objective measurement of perceived audio quality. *J.Audio Eng.Soc.*, (48)1/2, 3-29.

[32] Zeng, Y., Mao, H., & Peng, D. (2019). Spectrogram based multi-task audio classification. *Multimed Tools Appl*, (78), 3705–3722

[33] Kalantarian, H., Alshurafa, N., Pourhomayoun, M., Sarin, S., Le, T., & Sarrafzadeh, M. (2014). Spectrogram-based audio classification of nutrition intake, *2014 Health Innovations and Point-of-Care Technologies Conference*, Seattle, Washington USA, 161-164.

[34] Zhang, G., Zheng, L. Su, Z., Zeng Y., & Wang, G. (2023). M-Sequences and sliding window-based audio watermarking robust against large-scale cropping attacks. *IEEE Transactions on Information Forensics and Security*, (18), 1182-1195.

[35] Korany, N. O., Elboghdadly, N. M. & Elabdein, M. Z. (2024). High capacity, secure audio watermarking technique integrating spread spectrum and linear predictive coding. *Multimed Tools Appl*, (83), 50645–50668.

[36] Hua, G. Huang, J. Shi, Y.Q., Goh, J., Thing, V.L.L. (2016). Twenty years of digital audio watermarking-a comprehensive review, *Signal Processing*, (128), 222-242.

[37] Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia, *IEEE Trans. Image Process.*, 6(12), 1673-1687.

[38] Yeo, I.K., Kim, H.J. (2003). Modified patchwork algorithm: a novel audio watermarking scheme, *IEEE Speech Audio Process.*, 11(4), 381–386.