Ternary Echo Hiding in Audio Files

1st Rustam Latypov
Department for Computer Sciences
Kazan Federal University
Kazan, Russia
Email:Roustam.Latypov@kpfu.ru

2nd Evgeni Stolov
Department for Computer Sciences
Kazan Federal University
Kazan, Russia
Email:ystolov@list.ru

***Abstract*—We propose a new family of echo hiding procedures designed to work with audio files with ternary watermarks. Most attention is paid to the case when the image is used as a watermark. The possibility of using a melody for this purpose is also mentioned. A human is supposed to be a detector to prove the presence of a watermark in the audio file. The approach employs infinite impulse response (IIR) filters of a particular form that provides a capability insertion a few symbols into a fragment of the container. The suggested method's payload and resistance to various attacks exceed the parameters of the classical implementation of echo hiding.**

***Index Terms*—audio files, echo hiding, ternary watermark**

I. Introduction

Enforcing the author's rights is still an ongoing problem in multimedia. The author, who only starts the career, is induced to upload the created composition in open access. To protect the work from roguery, the creator employs watermarks embedded in the product. The goal is to do this gently without significantly distorting the host signal. Various methods for inserting watermarks into audio files can be found in review [1] as well as in the books [2], [3], [4]. It should be noted that embedded data have not to be sensitive to attacks through general transformations of the encoded audio signal, such as filtering, resampling, or lossy data compression.

Echo hiding is a well-known method for embedding data into an audio signal [5]. The method has various uses, including providing copyright protection and info integrity. The echo hiding watermarking is getting popular last time [6], [7], [8]. The reason for this is the simplicity of the insertion watermark and the lack of a clean file for the extraction of the watermark. Single echo hiding, bipolar echo hiding, backward-forward echo hiding, bipolar backward-forward echo hiding, and time-spread echo hiding methods were developed recently. In echo hiding audio watermarking method, data are embedded into cover audio by adding up delayed versions of the audio signal back to itself. In digital signal processing terms, this process corresponds to finite impulse response (FIR) filtering with an impulse response that consists of a delta impulse at time zero and a time-shifted and weighted delta impulse. To embed multiple echoes, one can use more than one delayed delta impulses.

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

Typically, before embedding a watermark, the watermark signal is converted to a binary sequence [1]. In [9], [10] it is shown, that the ternary form of the watermark is a practical and natural choice in certain cases. These are watermarks that can be recognized by the Human Auditory and Visual Systems since watermarks are music and images fragments converted into ternary sequences. The conversion of a music file into a ternary form is rather simple. Let $Music[k], k = 0, 1, \dots, L-1$ be a fragment of musical file written in wav format. Let us choose a threshold Thr and convert $Music$ into a ternary sequence

$$TMusic[k] = \begin{cases} 0, & |Music[k]| < Thr, \\ \mathbf{sign}(Music[k]), & \text{otherwise} \end{cases} \quad (1)$$

When playing a new sequence, we receive poor quality, but the main tune of the original music can be recognized. There is a problem with optimal setting the value of Thr in (1). This issue is investigated in [9]. It should be pointed out that (1) is not the only method of converting a music file into a ternary form. Another natural example is the image used to construct the watermark. The transformation of the grayscale image into a binary form is standard procedure. The development of an algorithm that converts grayscale pictures into ternary images is the subject of the paper [10]. The idea of the method is as follows. There are two threshold Thr_0, Thr_1. Let $Pict[u, v], u = 0, 1, \dots, L-1, \quad v = 0, 1, \dots, M-1$ be the matrix presentation of a picture, $0 \le Pict[u, v] \le 255$. The first step is calculation Thr_0, Thr_1 providing a kind of suboptimal approximation of the picture by means (2).

$$TPict[u, v] = \begin{cases} 0, & Pict[u, v] < Thr_0, \\ Thr_0, & Thr_0 \le Pict[u, v] < Thr_1, \\ Thr_1, & Pict[u, v] \ge Thr_1. \end{cases} \quad (2)$$

But the matrix $TPict[u, v]$ does not fit ternary presentation of a picture as a watermark since the values Thr_0, Thr_1 depend on picture. Instead, we implement standard representation of picture by means of the matrix $SPict[u, v]$ defined by (3) :

$$SPict[u, v] = \begin{cases} 0, & TPict[u, v] = 0 \\ 127, & TPict[u, v] = Thr_0 \\ 255, & Pict[u, v] = Thr_1. \end{cases} \quad (3)$$

While embedding a picture as a watermark, we convert the picture into standard form and then change the values $0, 127, 255$

for $-1, 0, 1$, respectively. An example of an image and its standard form is shown in Fig 1. One can see that there are more than three levels of brightness of the pixels in the standard form of the picture in Fig. 1. That is the result of the work of the viewer exploited for insertion pictures into documents.

In our paper, we present a new kind of echo hiding in audio files that fits the ternary form of watermarks. Recall the basic ideas realized in the echo hiding procedure [5]. Let

$$Fragm = s_0, s_1, \ldots, s_{N-1} \tag{4}$$

be a fragment of the audio file where the watermark is embedded. With the introduction of the echo, some element s_k changes to $\bar{s}_k = s_k + a \cdot s_{k+p}$ where p is an integer number, and a is a small value. Position p in the modified fragment can be revealed through cepstral analysis or autocorrelation of the fragment. That method can be generalized by the leverage of an arbitrary finite impulse response (FIR) filter

$$\bar{s}_k = \sum_{i \in \mathbf{S}} b_i \cdot s_{k-i}. \tag{5}$$

Here b_i are small values. $\mathbf{S}$ is a set of indices that are used for coding a watermark. In our paper, we suggest expanding this approach by replacing the FIR filter in (5) with a special-shape infinite impulse response (IIR) filter. In what follows, we think of trit and symbol in ternary sequence as synonyms. Throughout, we use the following notation: if A is an array, then FA is a result of discrete Fourier transforming (DFT) of A.

(a) Original picture

(b) Picture in standard form

Fig. 1: Example of the original picture and its standard form

II. IIR FILTER IN ECHO HIDING PROCEDURE

Let us suppose that a watermark is represented as a ternary sequence $Watr = \langle a_0, a_1, \ldots, a_M \rangle$ where $a_i \in \{-1, 0, 1\}$. Our goal is embedding of one or more symbols of the watermark into the Fragm (4) of the host file. We start with the case where only one symbol of the watermark is embedded into the fragment.

A. Simple IIR filter in echo hiding procedure

Let modified sequence item

$$\bar{s}_k = a \cdot \bar{s}_{k-p} + s_k - b \cdot s_{k-q}, \tag{6}$$

where p, q are natural numbers. This is a difference equation that defines how the output signal of the IIR filter is related to the input signal. To find the transfer function of the filter, we first take DFT of each part of (6), we get

$$Tr(n) = \frac{1 - b \cdot \exp(-w \cdot q \cdot n/N)}{1 - a \cdot \exp(-w \cdot p \cdot n/N)}, n = 0, 1, \ldots N-1. \tag{7}$$

In the arising formula $w = 2 \cdot \pi \cdot j$. Let $FFragm(n)$, $n = 0, 1, \ldots, N-1$ be the result of DFT of the whole fragment (4). Then the output of $FFragm$ filtering is the modified fragment $MFragm$ that has the form

$$MFragm = IDFT(Tr \cdot FFragm). \tag{8}$$

Here, the operator $\cdot$ denotes the elementwise product of two sequences of the same length, and $IDFT$ means the inverse transform for DFT. The fragment $MFragm$ replaces $Fragm$ in the host file. While extracting the watermark, we use the cepstral transform. The standard cepstral transform applied to the modified fragment $MFragm$ produces

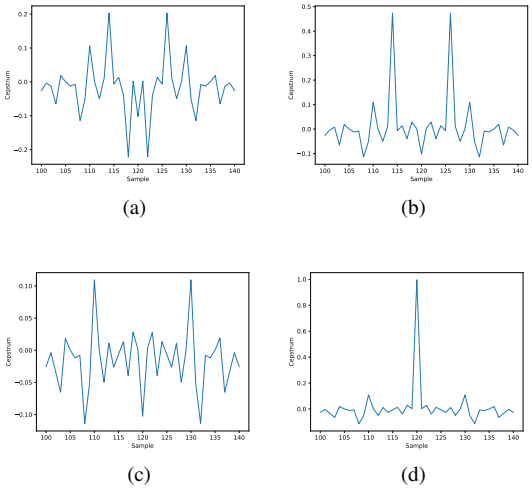
$$\begin{aligned} &Cepstr = IDFT(\log|FMFragm)|) = \\ &IDFT(\log|1 - b \cdot \exp(-w \cdot q \cdot n/N)|) - \\ &IDFT(\log|1 - a \cdot \exp(-w \cdot p \cdot n/N)|) + \\ &IDFT(\log|FFragm|). \end{aligned} \tag{9}$$

For small values of c the meaning $\log(1 + c \cdot \exp(j \cdot t)) \approx c \cdot \exp(j \cdot t)$. Since $\log|Z|^2 = \log(Z) + \log(\bar{Z})$ and $\exp(w \cdot k \cdot n/N) = \exp(-w \cdot (N-k) \cdot n/N)$, the cepstrum of $MFragm$ has four splashes at the points $p, N-p, q, N-q$. Assuming $p = q$ and $b = -a$ in (9), we denote the resulting cepstral function as $\overline{Cepstr}$. We have

$$Splash = \overline{Cepstr}(p) \approx 2 \cdot Cepstr(p), \tag{10}$$

and the sign of $Splash$ coincides with the sign of the parameter a under a small value of $|IDFFT(\log|FFragm|)|$ at this point. The value of $Splash$ is two times more than one in the cepstrum corresponding to the hiding procedure according to (5). A special case of (10) is the situation where $p = q = N/2$ and $b = -a$. In this case, $Splash \approx 4 * Cepstr(N/2)$ and is four times more than the splash in cepstrum related to (5). If $Symbol \in \{-1, 0, 1\}$ then the embedding of the $Symbol$ into a fragment is realized by the IIR filter (6) with $p = q$, $a = c \cdot Symbol$, and $b = -a$. The coefficient c influences the transparency of the embedding procedure.

All these assertions are demonstrated in Fig. 2. All the symbols are embedded into the same fragment of the host.



(a) (b) (c) (d)

Fig. 2: Cepstrum. Length of the fragment $N = 240$; (a) $p = 118$, $q = 114$, $a = b = 0.5$. (b) $p = q = 114$, $a = 0.5$, $b = -0.5$. (c) $p = q = 114$, $a = b = 0$. (d) $p = q = 120$, $a = 0.5$, $b = -0.5$

B. Embedding a few symbols of the watermark into a single fragment

At this point, we display an extension of the method for embedding a few symbols into a single fragment of the host. Suppose that among M symbols of a watermark, which must be inserted into the same fragment, only K symbols are nonzero. For ease, suppose these are K first symbols of the watermark. Let $Positions = \langle p_0, p_1, \ldots, p_{K-1} \rangle$ be the positions in the fragment spectrum where the nonzero symbols will be placed in and $\langle p_K, p_{K+1}, \ldots, p_{M-1} \rangle$ be the positions assigned to zero symbols. These data, corresponding to nonzero items, are used in the development of the IIR. The insertion procedure is presented in Algorithm 1. Here the transfer function is

$$Tr(n, p, c, N) = \frac{1 + c \cdot \exp(-w \cdot p \cdot n/N)}{1 - c \cdot \exp(-w \cdot p \cdot n/N)}. \qquad (11)$$

It follows from (11) that $Tr(n, p, c, N) = 1/Tr(n, p, -c, N)$. The IIR filter used to embed nonzero $Symbols$ is a series of connected simple IIR filters described in the previous section. An example of embedding four symbols $1, 1, 0, -1$ in the positions $106, 108, 110, 112$ of a fragment of length 240 is shown in Fig. 3.

III. Transparency of Embedding

Let us check the distortion of the original fragment after inserting K watermark characters according to Algorithm 1. To do this, use SNR in form (12)

$$SNR = 10 \cdot \log_{10} \left(\frac{\sigma^2(|FFragm|)}{\sigma^2(|FMFragm - FFragm|)} \right). \qquad (12)$$

Algorithm 1 Embedding M Symbols of Watermark into Single Fragment

Input: $Fragm; C; Symbols; Positions$
Output: $MFragm$ {Modified fragment}

$N \leftarrow Fragm$ {Length of fragment}
$M \leftarrow Symbols$ {Number of symbols}
$FFragm \leftarrow Fragm$ {Implement DFT}
for $I = 0$ to M **do**
  $S \leftarrow Symbols[I]$
  $P \leftarrow Positions[I]$
  if $S = 0$ **then**
    $continue$
  else if $S = 1$ **then**
    for $n = 0$ to $N - 1$ **do**
      $FFragm[n] \leftarrow FFragm[n] \cdot Tr(n, P, C, N)$
    end for
  else
    for $n = 0$ to $N - 1$ **do**
      $FFragm[n] \leftarrow FFragm[n]/Tr(n, P, C, N)$
    end for
  end if
end for
$MFragm \leftarrow FFragm$ {Inverse DFT}

A. Simple IIR filter

If $p = q$ and $b = -a$ then the transfer function has the form (11).We have

$$FMFragm(n) - FFragm(n) = Fragm(n)(Tr(n) - 1) = FFragm(n) \left(\frac{2 \cdot c \cdot \exp(-w \cdot p \cdot n/N)}{1 - c \cdot \exp(-w \cdot p \cdot n/N)} \right)$$

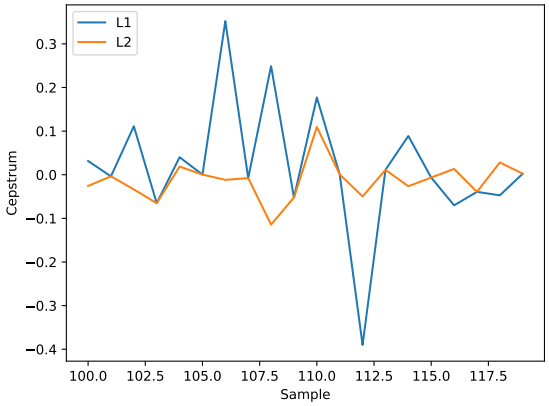


Fig. 3: Example of cepstrum. Length of the fragment $N = 240$; four symbols 1,1,0,-1 are embedded in positions 106,108,110,112. L1 - modified fragment, L2 – original fragment

TABLE I: Compare of Real SNR with Theoretical One in the Case of Single filter

Dependence on parameter c		
Parameter c	**Real SNR** (dB)	**Estimate of SNR** (13) (dB)
0.2	8	7.8
0.15	11	10.5
0.1	14	13.9
0.05	20	20

and

$$|FMFragm(n) - FFragm(n)| = |FFragm(n)| \left(\frac{2 \cdot c}{|1 - c \cdot \exp(-w \cdot p \cdot n/N)|} \right) \approx 2 \cdot |c||FFragm(n)|.$$

Hence

$$SNR \approx -20 \cdot \log_{10}(2|c|). \quad (13)$$

An example of comparing the real value of SNR with the theoretical one is presented in Table I.

From Table I, it follows that the estimate (13) is very close to the real value.

B. Series of simple IIR filters

It can be expected that transparency depends on the type of the embedded symbols. We can not present a simple formula for evaluating SNR for the full file since the symbols inserted in various fragments differ, and the SNR varies. To demonstrate the situation, we present some experimental results, which are collected in Table II. Here $M = 4$ and we use short notations for SNR after embedding symbols: **I** $\rightarrow [1, 1, -1, -1]$, **II** $\rightarrow [1, 1, 1, 1]$, **III** $\rightarrow [1, 0, 0, 1]$. One can see that there is a significant difference in the distortion of a fragment, depending on inserted symbols.

IV. Extraction of watermark

Let a be a symbol that is inserted at the position p of the spectrum. From (10) we obtain $|Cepstr(p) - \overline{Cepstr(p)}| \approx 2 \cdot c$, where $Cepstr(p), \overline{Cepstr(p)}$ denote the cepstra associated with $a \neq 0$ and $a = 0$ respectively. Since the positions $p_0, p_1, \ldots p_{K-1}$ are arbitrary, we select them with step 2. We assume that $Cepstr(p-1)$ and $Cepstr(p+1)$ are close to $\overline{Cepstr(p)}$. That is the basic idea realized in Algorithm 2. The values of cepstrum are randomly distributed, and the choice $c/2$ as $Bound$ value provides better results. Using Algorithm 2, one can extract only a single symbol

TABLE II: SNR Calculated in the Case Insertion Four Symbols in Fragment

Dependence on parameter c			
Parameter c	**I SNR** (dB)	**II SNR** (dB)	**III SNR** (dB)
0.2	12	-5	2
0.15	14	-2	5
0.1	18	2	8.5
0.05	24	8	14

Algorithm 2 Extraction of Watermark Symbol at the Given Position from the Modified Fragment

Input: $MFragm; c; p.$
Output: a {Symbol of watermark in position p}
1: $Cepstr \leftarrow MFragm$ {Cepstrum}
2: $Bound \leftarrow c/2$
3: $Diff \leftarrow Cepstr[p] - 0.5 \cdot (Cepstr[p+1] + Cepstr[p-1])$
4: **if** $|Diff| < Bound$ **then**
5: $a = 0$
6: **else**
7: $a = \mathbf{sign}(Diff)$
8: **end if**

from the fragment at a given position of the spectrum. This algorithm can be extended to the case where a few trits are inserted into fragment, the algorithm must be implemented for each position. An alternative approach to the problem is implemented in Algorithm 3 where the value of c is excluded from calculation. Here $AllMFragms$ is the list containing all modified fragments of the container and Pos – the positions in spectrum utilized for insertion M symbols. We use notation

Algorithm 3 Extraction of Watermark Symbols by Means of K-means Procedure

Input: $AllMFragms; Pos$
Output: $Watermark$
1: $M \leftarrow Pos$ {Number trits inserted in $MFragm$}
2: $Collect \leftarrow 0$ {Empty list}
3: **for** $MFragm$ in $AllMFragms$ **do**
4: $Cepstr \leftarrow MFragm$ {Cepstrum}
5: $Block \leftarrow M$ {Zero block of size M}
6: **for** $I = 1$ to M **do**
7: $P \leftarrow Pos[I]$
8: $Block[I] \leftarrow Cepstr[P] - 0.5 * (Cepstr[P-1] + Cepstr[P+1])$
9: **end for**
10: $Collect \leftarrow Block$ {Append to Collect $Block$}
11: **end for**
12: $Centroids \leftarrow kmeans(Collect, 3)$ {$kmeans$ creates centroids for 3 clusters}
13: $Watermark \leftarrow vq(Centroids, Collect)$ {vq distributes all collected values among 3 clusters}

$kmeans$ and vq for the function from [11], which realize the mentioned procedures. Since all hiding procedures are based on the modification of random spectrum values, one can not hope that all watermarks are restored accurately. We have to have a criterion for evaluating the quality of the algorithm intended to extract the watermark. The most natural digital value for evaluating an extraction procedure is the trit error rate (TER). That is the ratio of the number of incorrectly extracted trits to the total symbols in the watermark. Let us measure the quality of the two presented above algorithms. To this end, we leverage the image in standard form in Fig. 1. Length of fragment = 300, the number of trits embedded in

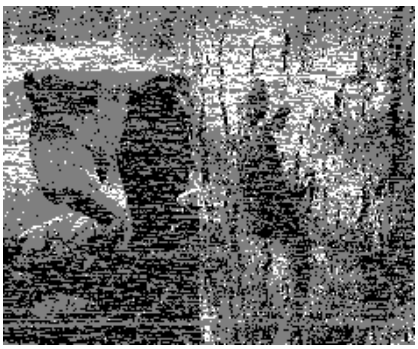


Fig. 4: Extracted watermark with TER=27%

fragment $M = 3$, container – a song written with sample frequency 44100Hz. The results of the experiment are placed in Table III. One can see that both algorithms have the same quality.

V. Attacks

The advantage of applying the picture as a watermark is shown in Fig. 4. Although TER is 27% (more than the quarter of the trits is restored incorrectly), the image is recognized without problems. We continue our experiments with the same container, figure, $M = 3, c = 0.1$, and length of fragment equals 300.

A. Filtering attack

Since the filtering of containers significantly changes the spectrum of the signal, the inserted watermark can not be recognized because $TER > 55\%$ in this case.

B. Additional noise attack

The random noise with uniform distribution, produced by the $random$ function from [11], is added to the watermarked container. The extraction of the watermark is performed through Algorithm 2 and Algorithm 3. The results are assembled in Table IV.

C. Compression of container attack

That is the most straightforward attack. The container written in wav format is converted to mp3 with various bitrates and converted to wav format again. The initial bitrate of the container is 705 kilobit per second (kbps). We use package PyDub for manipulation with a container [12]. The results are placed in Table V. That is the first case where Algorithm 3 shows its advantage against Algorithm 2.

TABLE III: Compare of Quality of Algorithm 2 and Algorithm 3

TER depending on parameter c		
Parameter c	**Alg2, TER** (%)	**Alg3, TER** (%)
0.2	8	7
0.15	3	4
0.1	4	4
0.05	17	16

TABLE IV: TER after Additional Noise Attack

Dependence on the level of noise		
SNR (dB)	**Alg2, TER** (%)	**Alg3, TER** (%)
42	6.1	6.2
35	8.5	8.4
32	10.7	10.6
29	12.8	12.9

TABLE V: TER after Compression

Dependence on bitrate		
Bitrate (kbps)	**Alg2, TER** (%)	**Alg3, TER** (%)
500	35	24
400	36	25
200	38	27
100	60	49

Conclusion

Implementation ternary picture as watermark has perspectives for hiding watermark in audio files because the payload is more than one compared with binary watermarks implemented with the same length of fragments. Utilizing human as a detector for recognition of the watermark is very effective since the watermark can be recognized even for a high level of TER. Usage of IIR filters for embedding increases splash of cepstrum at chosen points and provides better resistance to attacks compared to standard echo hiding.

References

[1] G. Hua, J. Huang, Y. Q. Shi, J. Goh, V. L. Thing,"Twenty years of digital audio watermarking. A comprehensive review, " *Signal Processing*,vol. 128, no. 11, pp. 222–242, 2016.

[2] N. Cvejic N, D. Drajic, T. Seppanen,"Audio watermarking: more than meets the ear, " *Recent Advances in Multimedia Signal Processing and Communications*, M. Grgic, K. Delac, M. Ghanbari, Eds. Berlin and Heidelberg: Springer Verlag, 2016, pp. 523–550.

[3] Y. Xiang , G. Hua , B. Yan, *Digital Audio Watermarking : Fundamentals, Techniques, and Challenges*, Singapore: Springer Verlag, 2017.

[4] R. Thanki, *Advanced Techniques for Audio Watermarking*, Switzerland: Springer Verlag, 2019.

[5] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding," *IBM Systems J.*, vol. 35, nos. 3&4, pp. 313–336, 1996.

[6] G. Hua, J. Goh, V. L. Thing, "Cepstral Analysis for the application of echo-based audio watermark detection, " *IEEE Trans. Inform Forensics Secur.* vol. 10, no 9, pp. 1850–1861, Sep., 2015.

[7] G. Hua, J. Goh, V. L. Thing, "Time-Spread Echo-Based Audio Watermarking With Optimized Imperceptibility and Robustness," *IEEE/ACM Trans. on Audio, Speech, and Language Processing.* vol. 23, no 2, pp. 227–239, Feb., 2015.

[8] S. Wang, W. Yuan, J. Wang, M. Unoki, "Inaudible Speech Watermarking Based on Self-compensated Echo-hiding and Sparse Subspace Clustering," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 2632–2636.

[9] K. Absalyamova, R. Latypov, E. Stolov, "Ternary Code of Melody and Reliable Audio Watermarking," *Proc. 27th Int. Telecommunication Forum(TELFOR)*, Belgrade, Serbia, 2019.

[10] R. Latypov, E. Stolov, "Ternary Picture as Watermark for Audio Files," *Proc.3rd Int. Conf. on Computer Applications & Information Security (ICCAIS)*, Er-Riyadh, Saudi Arabia, 2020.

[11] SciPy [Online]. Available: https://scipy.org/

[12] Pydub [Online]. Available: http://pydub.com/