A blind audio watermarking based on singular value decomposition and quantization

Chinmay Maiti[1] · **Bibhas Chandra Dhara**[2]

Abstract

In this work, we have proposed a robust audio watermarking technique in which a binary image is embedded into a host audio signal for copyright protection. In the embedding process, blocks of host audio signals are transformed by singular value decomposition method. Then, a watermark bit is embedded in each block by modifying the highest singular value of that block using a user defined quantization parameter. In the extraction process, the watermark is extracted from watermarked audio signal without using the host audio signal. The proposed technique has a high data payload having very good imperceptibility. The experimental result shows that the proposed method is robust under different audio attacks. The performance of the present method is also compared with other existing methods and present method has similar performance (sometimes better) as the existing methods. The novelty of the proposed work is the use of a new quantization method for the quantization of highest singular value. In this method, we scrambled the watermark image before embedding it and this increases the security level of the proposed scheme.

Keywords Audio watermarking · Singular value decomposition (SVD) · Audio attacks · Robustness · Error analysis

1 Introduction

Due to availability of multimedia data in digital form and advancement of multimedia technology, various multimedia data such as image, video, audio can be copied, modified and then, it can be distributed with the help of Internet technology to a distant place very easily. Therefore, it is very simple to access data by any one in the world. It is really a big challenge for the owner of the data to protect it from unauthorized users. The most promising solution to the problem of copyright protection of the digital multimedia data is digital watermarking (Cox et al., 2007), which is a process of embedding the copyright information into a host data. The copyright information is called watermark.

Recently, audio watermarking technique (Ercelebi & Batakci, 2009; Lei et al., 2011; Li & Wu, 2015; Wu et al., 2005) is a hot research topic and many researchers are getting interested in developing watermarking techniques for the copyright protection of audio signals. According to International Federation of the Phonographic Industry (IFPI; Vivekananda Bhat et al., 2010), an effective audio watermarking method must satisfy the following properties or characteristics: (i) Imperceptibility: The quality of the host audio signal should not be degraded after embedding the watermark information. It is the perceptual similarity between the host and the watermarked audio signals. As stated in IFPI, signal-to-noise ratio (SNR) between host and watermarked audio signals should be more than 20 dB. (ii) Robustness: Ability to extract a watermark information from a watermarked audio signal against different audio attacks. (iii) Payload: The amount of watermark information that can be embedded into the host audio signal which satisfies the imperceptibility property. It is defined as the number of watermark bits that can be embedded into a host signal per unit of time and is measured using bits per second (bps). It should be more than 20 bps. (iv) Security: The watermark information can be extracted only by the intended users. Watermarked signals should not reveal any clues about

✉ Chinmay Maiti
chinmay@cemk.ac.in

Bibhas Chandra Dhara
bcdhara@gmail.com

1 Department of Computer Science & Engineering, College of Engineering and Management, Kolaghat, India

2 Department of Information Technology, Jadavpur University, Kolkata, India

watermark information in them. The extraction of watermark from watermarked signal depends on secret keys only rather than the secrecy of the watermarking techniques. Actually, imperceptibility, robustness and payload are mutually conflict to each other. For example, increasing data rate in a watermarking technique results in quality degradation of the watermarked signal and increase of robustness against attacks. For designing and developing of an effective watermarking technique, one has to make a suitable trade-off among those properties.

In the literature, many watermarking techniques have been proposed for images and videos (Muhammad & Bibi, 2015; Rasti et al., 2016). These techniques can be applied for audio watermarking. However, developing audio watermarking technique is difficult compared to image and video watermarking techniques for two reasons (Vivekananda Bhat et al., 2011a). Firstly, the audio signals are represented by much less signals per time interval compared to images and videos. As a result, amount of information that can be added with audio signals is much lower than images and videos. Secondly, the human auditory system (HAS) is much more sensitive than the human visual system (HVS). It is therefore difficult to satisfy the imperceptibility (inaudibility) property of audio watermarking techniques than the imperceptibility (invisibility) property of the watermarking techniques for images and videos

In recent years, many audio watermarking methods are developed. The audio watermarking methods are broadly classified into two groups: (i) time-domain methods (Binny & Koilakuntla, 2014; Lie & Chang, 2006; Subbarayan & Ramanatha, 2009) and transform-domain methods (Dhar & Shimamura, 2015; Elshazly et al., 2017; Vivekananda Bhat et al., 2011b; Zhang, 2015). In time-domain based techniques, watermark information is embedded directly into the audio signals. These techniques are very easy to develop and they require less computation compared to transform domain techniques. In transform domain based techniques, audio signals are transformed into frequency domain using some transformation and then adjust the coefficients of the frequency components. These techniques are more robust than time-domain techniques. There are various types of time-domain methods such as LSB replacement, echo hiding, phase coding, patchwork etc. and transform-domain methods such as FFT, DWT, DCT, SVD etc.

Binny and Koilakuntla (2014) proposed a stenographic technique which embeds the text data into audio signal based on LSB method. To embed the text data, each audio trial is converted into bits and it is shown that the SNR value decreases as text length increases. A time-spread (TS) echo hiding based audio watermarking method is presented in Hu et al. (2016). Each watermark bit is embedded into the corresponding segment of audio signals by adding echos with different delays. In Lie and Chang (2006), an audio

watermarking technique has been presented in time-domain. The algorithm exploits differential average of the absolute amplitude relations to embed a watermark bit in group of audio samples. It maintains high audio quality and also shows robustness against the attacks. In Subbarayan and Ramanatha (2009), an LSB based audio watermarking technique has been described. Before embedding, watermark has been encrypted using RSA algorithm. This technique is not robust against attacks. In Chetan et al. (2021), an audio watermarking method based on modified LSB bit plan of audio signal has been proposed. The watermark is embedded on the least significant bits of samples through the audio signal instead of being localized to a particular set of bits. For embedding the watermark bit, audio signal is quantized to the range [0–255] and convert each sample to 8-bits binary form. A DCT based blind audio watermarking is proposed in Roy et al. (2015). In the embedding process, DCT represented audio signal is partitioned into non-overlapping segments each of same size and then each segment is divided into four non-overlapping frames. The watermark bit is embedded into a segment by modifying the coefficients of the four frames. In the experimental results, they have shown that the technique has a good imperceptibility and robustness, but data payload is not given. In Li and Wu (2015), a blind audio watermarking technique has been proposed based on lifting wavelet transform and QR decomposition. The watermark bits are embedded via quantization of the random correlation coefficients which is computed from a random vector generated by a secret key and the vector obtained by the QR decomposition of the lifting wavelet coefficients. The experimental results show that the technique has a good robustness, imperceptibility and data payload. In Zhang (2015), a semi-fragile audio watermarking scheme is proposed based on DWT and DCT domain for copyright and content authentication.

To improve the effectiveness of the watermarking techniques, many researchers are using singular value decomposition (SVD; Dhar & Shimamura, 2014; Lei et al., 2012) to embed watermark into the host data. In SVD based audio watermarking system, the obvious preprocessing step is the conversion of 1-D signal into 2-D signal (2-D matrix) and then SVD is applied on the 2-D matrix. In Dhar and Shimamura (2014), a blind audio watermarking technique has been described in fast Fourier transform (FFT) domain based on SVD and Cartesian-polar transformation (CPT). The FFT is applied on each audio frame and then SVD is used for the low frequency components of FFT. The highest singular value is decomposed into two components using Cartesian-polar transform for embedding watermark information. In Vivekananda Bhat et al. (2010), a blind audio watermarking technique has been proposed. The watermark bits are embedded by modifying the singular values of the wavelets coefficients. The method has shown very good imperceptibility and robustness,

but the payload is very low. In Lei et al. (2011), a blind audio watermarking technique has been described based on SVD and DCT. A binary watermark is embedded into the high-frequency coefficients of the SVD–DCT blocks. The method has also shown low payload. In Farzaneh and Toroghi (2020) an audio watermarking scheme has been proposed based on graph-based transform and SVD. This scheme has limited efficiency as it is robust against compression and additive noise. In Novamizanti et al. (2020), authors have proposed an audio watermarking algorithm based on LWT–DCT–SVD. For embedding the watermark, algorithm decomposes the audio signals by LWT and then selected sub-band is transformed by DCT. The output of the DCT is used for SVD process and the singular matrix is modified by quantization to embed the watermark. A SVD based audio watermarking method using dual watermarking for copyright protection has been proposed in Patil and Chitode (2021). DWT process is applied on audio signal followed by SVD process to embed the watermark. The watermark is embedded by linear combination of the SVD component of watermark image and audio signal. Elshazly et al. (2021) proposed a audio watermarking algorithm based on DWT and SVD. They have embedded the color image as watermark. During watermark embedding, watermark bit is embedded by quantization of the largest singular values (LSVs).

In this paper, a blind audio watermarking technique is proposed based on SVD and quantization. The watermark bit is embedded by modifying the LSV of each audio signal block. The advantage of using the LSVs and quantization are: (i) The quantization of the LSV does not affect the signal quality. (ii) Change of LSVs under attacks are very insignificant. (iii) The quantization is simple, easy to implement and less complexity. (iv) The quantization method helps to achieve a good trade-offs among payload, imperceptibility and robustness. In the present work, to enhance the security level the watermark image is scrambled using Fibonacci–Lucas transformation (FLT). The proposed method is simple (less complex) and easy to implement.

This paper is organized as follows. Section 2 provides background information including watermark scrambling using Fibonacci–Lucas transform and SVD. Section 3 demonstrates the embedding and extraction process of the proposed technique. The experimental results are given in Sect. 4. Section 5 provides performance analysis of the proposed technique. Finally, conclusions are given in Sect. 6.

2 Background information

2.1 Fibonacci–Lucas transformation

In this work, we use Fibonacci–Lucas (Mishra et al., 2012) transformation to scramble the binary watermark image for enhancing the confidentiality of the proposed method. The FLT is defined as the mapping $FLT : T^2 \rightarrow T^2$ as given in Eq. (1).

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} F_i & F_{i+1} \\ L_i & L_{i+1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \mod N, \tag{1}$$

where $(x, y) \in \{0, 1, \ldots, N-1\}$, F_i is the $i\,th$ term of the Fibonacci series and L_i is the $i\,th$ term of the Lucas series, $N \times N$ is the size of the image. Using FLT, given in Eq. (1), the pixel (x, y) of the given image is mapped into a new location (x', y'). So, for a particular i, the FLT matrix is defined as

$$FLT_i = \begin{pmatrix} F_i & F_{i+1} \\ L_i & L_{i+1} \end{pmatrix}.$$

This way one can define many Fibonacci–Lucas transforms by changing the value of i. So, one can use Fibonacci–Lucas transform as more secured scrambling method. For example, the first matrix of this series is for $i = 1$ is given by

$$FLT_1 = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}.$$

It may be noted that this FLT_1 is nothing but the Arnold transform.

In this work, we have scrambled the watermark image by applying the FLT. This transformation is adopted to ensure more security.

2.2 Singular value decomposition

There are enormous applications of SVD (Aslantas, 2009; Lai, 2011) in the field of image processing. The image can be viewed as a matrix of non-negative scalar elements. SVD decomposes a matrix A of size $m \times n$ as product of three matrices: U, Σ and V, i.e., $A = U\Sigma V^T$. Here, $U_{m \times m}$ and $V_{n \times n}$ are orthogonal matrices. The matrix $\Sigma_{m \times n}$ whose $i\,th$ diagonal entry is equals to $i\,th$ singular value. The singular values are denoted as α_i, $i = 1, \ldots, n$. When the rank of A is p, elements of Σ satisfy the conditions $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_p > 0$ and $\alpha_{p+1} = \alpha_{p+2} = \cdots = \alpha_n = 0$. The SVD is used frequently by many researchers in designing watermark techniques due its attractive properties (Vivekananda Bhat et al., 2011a, b; Lai, 2011; Lei et al., 2012). In this work, we have considered square matrix of size, say, $n \times n$. Then SVD of $A_{n \times n}$ is formulated as given in Eq. (2).

$$A = U\Sigma V^T = \begin{bmatrix} u_1, \ldots, u_n \end{bmatrix} \times \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_n \end{bmatrix} \times \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}. \tag{2}$$

3 Proposed technique

The proposed method is discussed in this section. In the watermark embedding process, first the watermark is scrambled by FLT and then the host audio signal is decomposed by SVD and watermark bits are embedded by modifying the LSV. The watermark extraction process is simple one, in which the watermarked audio signal is decomposed by SVD and then from LSV watermark bit is determined. The detail of the embedding and extraction process are described in the following subsections.

3.1 Embedding process

The block diagram of the embedding process is shown in Fig. 1. In the proposed method, SVD is applied to embed to watermark bits. We know that SVD can be applied on a 2-D matrix. But, the given audio signal is an 1-D signal. For simplicity, we have considered square matrix. So, the audio signal is converted into 2-D signal of size $M \times M$. Suppose, $X = \{x(i), i = 1, 2, \ldots, L\}$ be a host audio signal of L samples. The audio signal X is converted into 2-D audio signal of size $M \times M$, where $M = \sqrt{L}$.

The 2-D audio is partitioned into non overlapping blocks B_k of size $q \times q$. The number of such blocks is equal to $\frac{M}{q} \times \frac{M}{q}$ which is same as the watermark size, i.e., in each block B_k we embed a bit. So, depending on the size of the
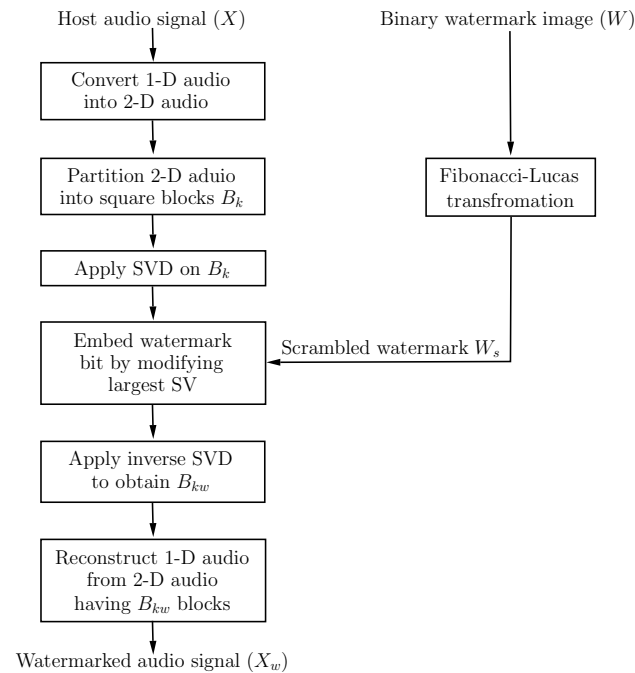
watermark image, the length of host signal is determined or vice versa.

Before embedding the watermark, it is scrambled by FLT. The Fibonacci–Lucas transform matrix is used as the secret key to enhance the security level of the watermark.

The block B_k is decomposed by SVD method and the watermark bit is embedded into the block B_k by modifying the LSV $\alpha_{k,1}$. Here, a user defined quantization interval Δ is used in the embedding process. The $\alpha_{k,1}$ is quantized as given in Eq. (3).

$$h_k = \left\lceil \frac{\alpha_{k,1}}{\Delta} \right\rceil, \tag{3}$$

where value of $h_k \in [0, H_{max}]$ and H_{max} is computed as $H_{max} = \frac{||A||_2}{\Delta}$, where $||A||_2$ represents the induced 2-norm of the matrix A. In the embedding process, we follow a simple logic. Here, we discuss the process when watermark bit is '0' and exactly symmetric method can be applied when watermark bit is '1'.

If watermark bit is '0', we modify the $\alpha_{k,1}$ to $\alpha'_{k,1}$, as the middle value of the previous/next quantization level (which is appropriate), so that corresponding h_k (i.e., h'_k) will be even. So, when watermark bit is '0' we have four different cases, which are as follows:

(1) If h_k is even then do nothing, i.e., $\alpha'_{k,1} = \alpha_{k,1}$.
(2) If $h_k = 1$, then $\alpha'_{k,1} = \frac{3\Delta}{2}$, i.e., $\alpha'_{k,1} = h_k * \Delta + \frac{\Delta}{2}$ (see Fig. 2a).
(3) If $h_k = h_{max}$ and h_k is odd, then $\alpha'_{k,1} = h_k * \Delta - \frac{3\Delta}{2}$ (see Fig. 2b).
(4) If h_k is odd, between the middle value of the previous quantization interval $\left(P_m = h_k * \Delta - \frac{3\Delta}{2}\right)$ and the middle value of the next quantization interval

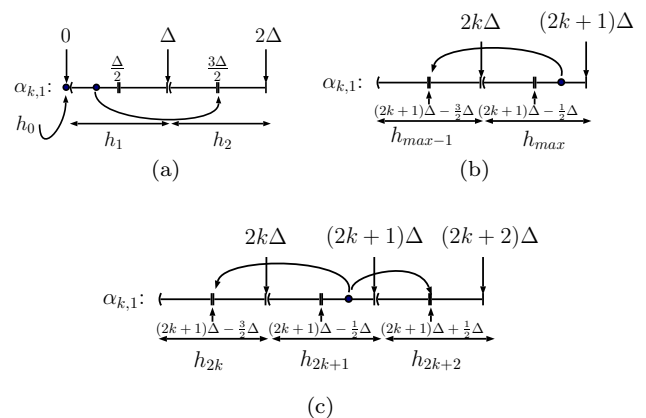**Fig. 1** Block diagram of embedding process

Fig. 2 Watermark bit embedding process when watermark bit is '0' and k is odd for h_k. Three cases are illustrated: **a** extreme left, i.e., h_1, **b** extreme right, i.e., h_{max} and **c** middle case, i.e., $1 < k < max$

$\left(N_m = h_k * \Delta + \frac{\Delta}{2}\right)$ which one is closest to $\alpha_{k,1}$ is consider as $\alpha'_{k,1}$ (see Fig. 2c).

The algorithmic sketch of the proposed embedding process is given in **Algorithm: Embedding Process()**.

Algorithm: Embedding Process(X, W, FL, Δ, X_w)

Input: Host audio(X), Binary watermark(W), FLT matrix (FL), Quantization interval(Δ)

Output: Watermarked audio(X_w)

---

(1) Convert the 1-D host audio signal X of L samples into 2-D audio signal S.

(2) Partition the 2-D audio signal S into non-overlapping blocks B_k, $k = 1, 2, \ldots, N \times N$, each of size $q \times q$.

(3) The watermark W of size $N \times N$ is scrambled by FLT which gives scrambled watermark W_s.

(4) SVD is applied on each block B_k and $B_k = U_k \Sigma_k V_k^T$, where the LSV is $\alpha_{k,1}$.

(5) Quantize $\alpha_{k,1}$ and obtain $h_k = \left\lceil \frac{\alpha_{k,1}}{\Delta} \right\rceil$.

(6) The watermark bits are embedded by modifying $\alpha_{k,1}$ as $\alpha'_{k,1}$

   (a) $P_m = h_k * \Delta - 3 * \frac{\Delta}{2}$

   (b) $N_m = h_k * \Delta + \frac{\Delta}{2}$

   (c) If watermark bit $W_{s,k} = 0$ //make h_k even

$$\alpha'_{k,1} = \begin{cases} 0, & \text{if } h_k = 0 \\ N_m, & \text{if } h_k = 1 \\ \alpha_{k,1}, & \text{if } h_k \text{ is even} \\ P_m, & \text{if } h_k \text{ is odd and } h_k = H_{max} \\ P_m, & \text{if } h_k \text{ is odd and} \\ & |\alpha_{k,1} - P_m| < |\alpha_{k,1} - N_m| \\ N_m, & \text{if } h_k \text{ is odd and} \\ & |\alpha_{k,1} - P_m| \geq |\alpha_{k,1} - N_m| \end{cases}$$

   (d) When watermark bit $W_{s,k} = 1$ //make h_k odd

$$\alpha'_{k,1} = \begin{cases} N_m, & \text{if } h_k = 0 \\ \alpha_{k,1}, & \text{if } h_k \text{ is odd} \\ P_m, & \text{if } h_k \text{ is even and } h_k = H_{max} \\ P_m, & \text{if } h_k \text{ is even and} \\ & |\alpha_{k,1} - P_m| < |\alpha_{k,1} - N_m| \\ N_m, & \text{if } h_k \text{ is even and} \\ & |\alpha_{k,1} - P_m| \geq |\alpha_{k,1} - N_m| \end{cases}$$

(7) Apply inverse SVD to obtain watermarked block B_{kw}

   (a) $\Sigma_{kw} = \Sigma_k - \{\alpha_{k,1}\} \cup \{\alpha'_{k,1}\}$

   (b) $B_{kw} = U_k \Sigma_{kw} V_k^T$

(8) The watermarked audio signal X_w is reconstructed from all blocks B_{kw}.

(9) Return X_w

3.2 Extraction process

The watermark extraction process is very simple. Here, the watermarked audio signal, X_w, is process like the embedding method and from the LSV $\alpha_{w_{k,1}}$ we compute the value h_{w_k}. The parity (even/odd) of h_{w_k} determines the watermark bit (Wt_k), i.e.,

$$Wt_k = \begin{cases} 0, & \text{if } h_{w_k} \bmod 2 = 0, \\ 1, & \text{if } h_{w_k} \bmod 2 = 1. \end{cases}$$

Finally, this watermark is unscrambled using FLT and we obtain the binary watermark W'. The proposed method is blind as no other information is required other than watermarked audio signal X_w. The block diagram of the proposed extraction process is shown in Fig. 3 and the algorithmic structure is given in **Algorithm: Extraction Process()**.

Algorithm: Extraction Process(X_w, FL, Δ, W')

Input: watermarked audio(X_w), FLT matrix(FL), Quantization interval(Δ)
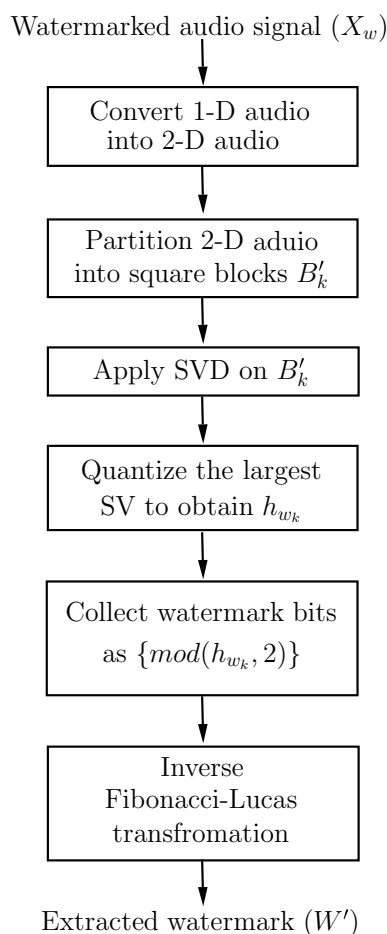
Output: Extracted binary watermark(W')

---

(1) Convert the 1-D watermarked audio signal X_w of L samples into 2-D audio signal S'.

(2) Partition the 2-D audio signal S' into non-overlapping blocks B'_k, $k = 1, 2, \ldots, N \times N$, each of size $q \times q$.

(3) SVD is applied on each block B'_k and $B'_k = U'_k \Sigma'_k V'^T_k$, where, the LSV is $\alpha_{w_{k,1}}$.

(4) Quantize $\alpha_{w_{k,1}}$ and obtain $h_{w_k} = \left\lceil \frac{\alpha_{w_{k,1}}}{\Delta} \right\rceil$.

(5) If $\bmod(h_{w_k}, 2) = 0$, then $Wt_k = 0$ Else $Wt_k = 1$

(6) Assemble all extracted watermark bits and then apply inverse FLT using FL and obtain extracted watermark W'

(7) Return W'

4 Experimental results

In this section we demonstrate the experimental result of the proposed watermarking method. We have implemented the proposed method using MATLAB 7.1.

Four different host audio signals namely, 'Classical', 'Jazz', 'Piano' and 'Tabla', are used as test signal in the experiment. These test signals are provided by the corresponding author of Ph.D. Thesis (Ghosal, 2014). Each audio signal has sampling rate 44.1 kHz with 16 bits/sample in the WAVE format. Each audio signal contains 230,400 samples

Watermarked audio signal (X_w)

↓

Convert 1-D audio into 2-D audio

↓

Partition 2-D aduio into square blocks B'_k

↓

Apply SVD on B'_k

↓

Quantize the largest SV to obtain h_{w_k}

↓

Collect watermark bits as $\{mod(h_{w_k}, 2)\}$

↓

Inverse Fibonacci-Lucas transfromation

↓

Extracted watermark (W')

Fig. 3 Block diagram of extraction process

(duration 5.224 s). These audio signals are transformed into 2-D signals with size 480×480 and each audio signal is further divided into blocks of size 15×15. Number of blocks is $\frac{480}{15} \times \frac{480}{15} = 32 \times 32$. Since, we assume that a bit will be embedded into each block then a binary image of size 32×32 is used as the watermark in our proposed method. If the blocks size is bigger than 15×15 we can use the current watermark for embedding purpose but in that case some bits will not be used. On the other hand, when we have smaller size blocks, then for some blocks there will no bit for embedding.

The watermark image is scrambled using FLT as described in Sect. 2. The original and the scrambled version of the watermark are shown in Fig. 4a and 4b, respectively. The quantization threshold Δ is set as 0.19 to achieve a good trade-off between conflicting requirements of imperceptibility and robustness and detail of this is discussed in Sect. 5.2. Since, the embedding and extraction process of the proposed technique depend on Δ, the Δ value is used as one secret key of the method. Another secret key is the FLT matrix. Without appropriate transform matrix, no one can identify

Fig. 4 Binary watermark image of size 32×32: **a** original and **b** scrambled

(a) (b)
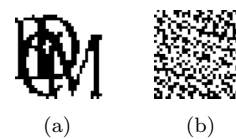
the actual sequence of the watermark by reverse scrambling process. The host and watermarked audio signal 'Jazz' are shown in Fig. 5.

4.1 Imperceptibility test

In our experiment, the imperceptibility test is performed using SNR which is used to measure the quality of watermarked audio signal. It is defined as

$$SNR = 10 * \log_{10} \frac{\sum_{i=1}^{L} X^2(i)}{\sum_{i=1}^{L} [X(i) - X_w(i)]^2} \text{ dB}, \quad (4)$$

where X and X_w are host and watermarked audio signals. The SNR is more when the difference between the host signal and the watermarked signal is less, i.e., imperceptibility is less (which is desired property of any watermarking technique). In Fig. 5c, the difference signal between the original and watermarked 'Jazz' is shown. From this figure we may note that the magnitude of the difference signal is very close to zero, so the imperceptibility is less. The SNR of the watermarked audio signals of 'Classical', 'Jazz', 'Piano' and 'Tabla' are 29.14 dB, 22.16 dB, 26.91 dB and 21.10 dB respectively. According to the IFPI standard, the SNR value of all the watermarked audio signals should be above 20 dB (Vivekananda Bhat et al., 2010). Our proposed technique satisfies the IFPI requirement.

4.2 Robustness test

Robustness means the ability to extract the watermark under the attacks, i.e., we need to measure the similarity/dissimilarity between the original watermark nd extracted watermark. In this experiment, we use two parameters: (i) normalized coefficient (NC) and (ii) bit error rate (BER).

The NC is the measure of similarity between the watermarks and it is defined as follows:

$$NC = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} W(i,j)W'(i,j)}{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} W(i,j)} \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} W'(i,j)}}. \quad (5)$$

If $NC(W, W')$ is close to 1, then the similarity between W and W' is very high else the similarity is very low.

The BER is the measure of dissimilarity and it is defined as

(a) Host signal (b) Watermarked signal (c) Difference signal
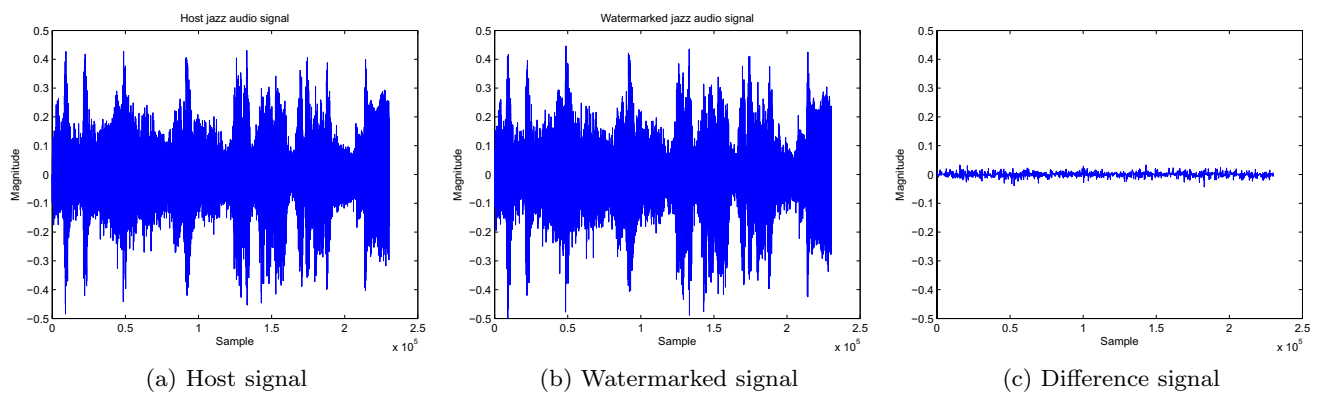
Fig. 5 Performance of the proposed watermark embedding method on the host signal 'Jazz'

$$BER = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} W(i,j) \oplus W'(i,j)}{N \times N} \times 100\%, \quad (6)$$

where $\oplus$ is the exclusive or (XOR) operator and W and W' are the original and extracted watermarks, respectively. So, when BER value is close to zero then we may infer that both the watermarks (original and extracted) are close enough.

So, a method is robust when the *NC* between the original watermark and extracted watermark is high and BER is less. To measure the robustness of the proposed method, we have considered different audio attacks. The audio editing and attacking tools used in this experiment are Adobe Audition 1.0 (for echo addition and inverse attacks), GoldWave 5.18 (for denoising, smoothing, MP3 compression and resampling attacks) and MATLAB (for low-pass filtering, additive white Gaussian noise, cropping and requantization attacks).

(i) *Low-pass filtering* low-pass filtering using second order Butterworth filter with 11.025 kHz cut-off frequency is performed on the watermarked audio signals.

(ii) *Additive noise* white Gaussian noise is added to the watermarked audio signals until the SNR of the resulting signal below 20 dB.

(iii) *Cropping* segments of 500 samples of the watermarked audio signals are replaced by the segments of the watermarked signal attacked with additive white Gaussian noise at five positions.

(iv) *Echo addition* an echo signal with a delay of 98 ms and a decay of 41% is added to the watermarked audio signal.

(v) *Denoising* the "Hiss removal" function of GoldWave is used to denoise the watermarked audio signal.

(vi) *Reverse amplitude* reverse the sign of the sample amplitudes.

(vii) *Smoothing* the smoothing operation of GoldWave is used to produce slow changes to the watermarked audio signal.

(viii) *Re-quantization* the 16-bit watermarked audio signals is quantized down to 8 bits/sample and then back to 16 bits/sample.

(ix) *MP3 compression* the MPEG-1 layer-3 compression is applied using GoldWave on the watermarked audio signals. The watermarked audio signal is compressed at different bit rates (128 kbps, 64 kbps and 48 kbps) and then decompressed back to the WAVE format.

(x) *Re-sampling* as the host audio signal is sampled at 44.1 kHz, thus, watermarked audio signal is down-sampled at 22.05 kHz, 11.025 kHz, 8.0 kHz and then up-sampled back to 44.1 kHz.

The extracted watermarks along with the NC and BER values under different attacks on the watermarked audio signal 'Jazz' are presented in Table 1. The NC values are all above 0.9837 and the BER values are all below 1.9%. The extracted watermark images are visually similar to the original watermark image. This establishes high robustness of the proposed technique for the 'Jazz' audio signal. In the first column of this, the numerals are used to represent the different attacks and these numerals are used in the next tables to refer the attacks.

Similar results for the audio signals 'Piano', 'Tabla' and 'Classical' are shown in Table 2 (due to space, we ignore the extracted watermark images). The NC values are above 0.8879, so 90% or more accurate. On the other hand, the performance under the attack 'echo addition' is comparatively poor in terms of BER, which is around 12.5%. As a whole, the proposed method is robust against almost all attacks except 'echo addition'.

4.3 Data payload

The data payload refers to the number of bits that can be embedded into the audio signal per unit of time. The data payload for watermarking techniques can be measured by

Table 1 Extracted watermark from watermarked 'Jazz' audio signal under different audio attacks with NC and BER

Sr. nos.	Attacks	NC	BER (%)	Extracted watermark
1	No attack	1.0000	0.0000	
2	Low-pass filtering	0.9837	1.8555	
3	Additive noise	1.0000	0.0000	
4	Cropping	1.0000	0.0000	
5	Echo addition	0.9896	1.1719	
6	Denoising	1.0000	0.0000	
7	Reverse amplitude	1.0000	0.0000	
8	Smoothing	1.0000	0.0000	
9	Re-quantization	1.0000	0.0000	
10	MP3 (128 kbps)	1.0000	0.0000	
11	MP3 (64 kbps)	1.0000	0.0000	
12	MP3 (48 kbps)	1.0000	0.0000	
13	Re-sampling (44.1–22.05–44.1)	1.0000	0.0000	
14	Re-sampling (44.1–11.025–44.1)	1.0000	0.0000	
15	Re-sampling (44.1–8.0–44.1)	1.0000	0.0000	

various methods. In the proposed method data payload is denoted as D_p and it is defined as

$$D_p = \frac{F_s}{N_B} \text{ bps} , \tag{7}$$

where F_s is sampling rate of the audio signal in Hz, N_B is the number of samples in each block. From Eq. (7), it has been seen that the data payload can be increased by decreasing the number of samples in each block. The data payload can effect the imperceptibility and robustness of any audio watermarking technique. In our proposed technique, we have performed our experiment with varying number of samples in each block (i.e., varying data payload) and we have observed that SNR is increasing when the data payload

is decreasing. In our method, if we increase the size of the block then payload will be decreasing. In Table 3, the SNR values with varying data payload on different audio signals are given. For our proposed method, F_s is 44,100 Hz and $N_B = 225$, thus data payload of our method is 196 bps. This is a very high payload as typical payload is 20–50 bps (Lei et al., 2012).

The data payload also effect the robustness of the watermarking method. It is obvious that robustness (under any attack) will be decreasing when data payload is increasing. The results in Table 4 demonstrate that our proposed technique is robust even with high payload on 'Classical' audio signal under different attacks.

From Tables 3 and 4, we may note that if we increase the window size ($N = q \times q$) both the imperceptibility and the robustness increases at the cost of payload (which decreases very fast). Therefore, other window size cannot be considered.

4.4 Security

In the proposed method, we have used a quantization interval Δ during watermark embedding and watermark extraction. For unauthorized users, it is impossible to extract the watermark without actual value of Δ. This quantization interval is used as secret key in our method. Another secret key is FLT matrix which is used to scramble the watermark image. Without correct matrix, it is not possible to identify the actual sequence of the watermark. These two secret keys actually enhance the security of the watermark.

5 Performance analysis

5.1 Error analysis

The error analysis of a watermarking technique is characterized by two types of error: (i) false positive error and (ii) false negative error. It is difficult to give an exact model of these errors. Here, we adopt a simplified model based on binomial probability distribution to provide an error analysis to compute the probability of two types of errors for the proposed technique.

5.1.1 False positive error analysis

The false positive error is the probability that an unwatermarked audio signal is considered to be watermarked during extraction process. Let r be the total number of watermark bits and s be the number of matching bits computed while extracting watermark. The extracted watermark bits are assumed to be independent random variables. The probability P_e is defined

as the probability that extracted watermark bits are matched with the original watermark bits. According to Bernoulli trials, we denote the probability P_s as the probability that exactly s number of matching bits are found out of r watermark bits and it is defined as

$$P_s = \binom{r}{s} P_e^s (1 - P_e)^{r-s}, \tag{8}$$

where $\binom{r}{s}$ is the binomial coefficient. Since watermark values are either 0 or 1, thus P_e is $\frac{1}{2}$. The false positive error will occur when the number of matching bits are greater than or equal to some threshold T_p. Then, the probability of false positive error is denoted as P_{fp} and it is defined as

$$P_{fp} = \sum_{s=T_p}^{r} P_s = \sum_{s=T_p}^{r} \frac{\binom{r}{s}}{2^r}. \tag{9}$$

Table 2 NC and BER of extracted watermark from watermarked audio signal under different attacks for 'Piano', 'Tabla' and 'Classical'
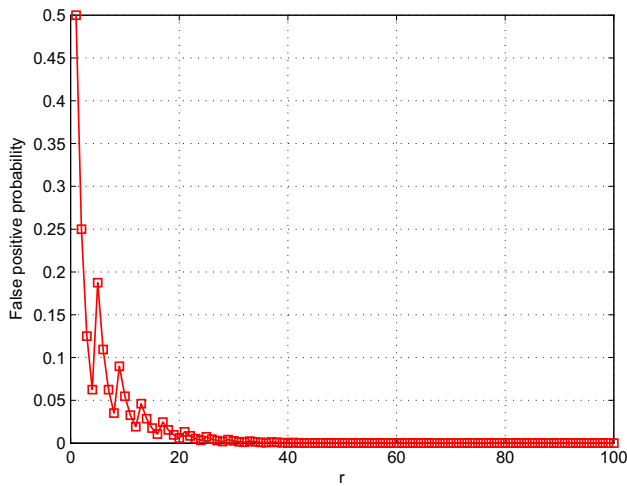
Attacks	Piano		Tabla		Classical	
	NC	BER (%)	NC	BER (%)	NC	BER (%)
(1)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(2)	1.0000	0.0000	0.9957	0.4883	0.9430	6.3477
(3)	1.0000	0.0000	0.9983	0.1953	0.9974	0.2930
(4)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(5)	0.8879	12.5506	0.9957	0.4883	0.8917	12.1094
(6)	1.0000	0.0000	0.9948	0.5859	0.9991	0.0977
(7)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(8)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(9)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(10)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(11)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(12)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(13)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(14)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(15)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

Table 3 SNR versus payload

Audio signal	Payload					
	N = 225	N = 256	N = 400	N = 576	N = 784	N = 1024
	(196 bps)	(172 bps)	(110 bps)	(76 bps)	(56 bps)	(43 bps)
Classical	29.14	29.47	30.12	31.77	33.04	34.48
Jazz	22.16	22.88	25.00	26.24	28.10	28.69
Piano	26.91	27.38	28.12	28.66	30.93	31.94
Tabla	21.10	21.73	23.33	24.52	25.67	26.09

Table 4 BER (%) versus payload (under attacks) with 'Classical' audio signal

Attacks	BER					
	N = 225	N = 256	N = 400	N = 576	N = 784	N = 1024
	(196 bps)	(172 bps)	(110 bps)	(76 bps)	(56 bps)	(43 bps)
(2)	6.3447	4.3984	1.9973	0.7923	0.0000	0.0000
(3)	0.2930	0.1613	0.0000	0.0000	0.0000	0.0000
(6)	0.0977	0.0000	0.0000	0.0000	0.0000	0.0000
(9)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
(11)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
(14)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Fig. 6 Probability of false of positive error

Table 5 NC, BER (%) and false negative error probability (P_{fn}) against different audio attacks

| Attacks	Average NC	Average BER (%)	P_{fn}
(1)	1.0000	0.0000	0.0000
(2)	0.9806	2.1729	3.52×10^{-247}
(3)	0.9989	0.1221	7.89×10^{-291}
(4)	1.0000	0.0000	0.0000
(5)	0.9412	6.5880	1.23×10^{-164}
(6)	0.9985	0.1709	8.32×10^{-290}
(7)	1.0000	0.0000	0.0000
(8)	1.0000	0.0000	0.0000
(9)	1.0000	0.0000	0.0000
(11)	1.0000	0.0000	0.0000
(14)	1.0000	0.0000	0.0000

In our proposed scheme, we assume that an unwatermarked audio signal is claimed to be watermarked when 75% or more bits are matched. Thus, $T_p = \lceil 0.75 \times r \rceil$ and then P_{fp} is given as

$$P_{fp} = 2^{-r} \sum_{s=\lceil 0.75 \times r \rceil}^{r} \binom{r}{s}, \tag{10}$$

where r is the number of bits in the watermark (here $r = 32 \times 32 = 1024$). A plot of the false positive error probability for $r \in (0, 100]$ is shown in Fig. 6. It is observed that probability of false positive error approaches to zero when $r \geq 25$. In this experiment, the false positive error probability P_{fp} is close to 0. We obtain $P_{fp} = 9.12 \times 10^{-318}$ by putting r = 1024 in Eq. (10).

5.1.2 False negative error analysis

False negative error is the probability that a watermarked audio declared as unwatermarked one. Let r be the total number of watermark bits and s is the number of matching bits. An watermarked audio signal is considered to be unwatermarked when the number of matching bits is less than or equal to a threshold T_n. Similarly, the probability of false negative error is denoted as P_{fn} and it is defined as

$$P_{fn} = \sum_{s=0}^{T_n} \binom{r}{s} P_e^r (1 - P_e)^{r-s}. \tag{11}$$

Here, we consider 75%, i.e., if number of matching bits is less than 75% we claim that there is no watermark in the

Fig. 7 Probability of false of negative error
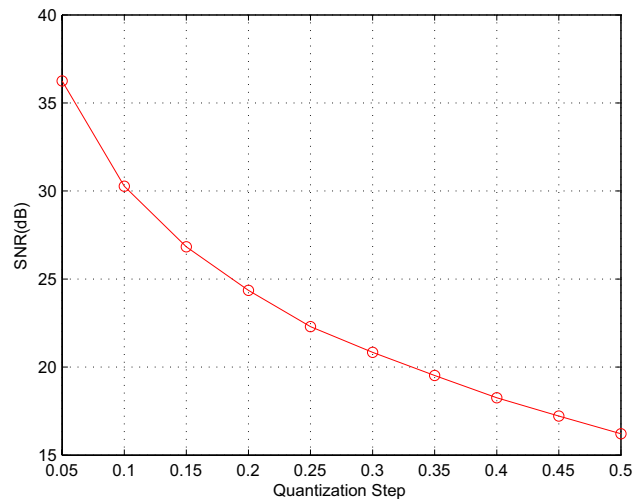
Fig. 8 SNR (dB) values under varying quantization steps (measures imperceptibility)
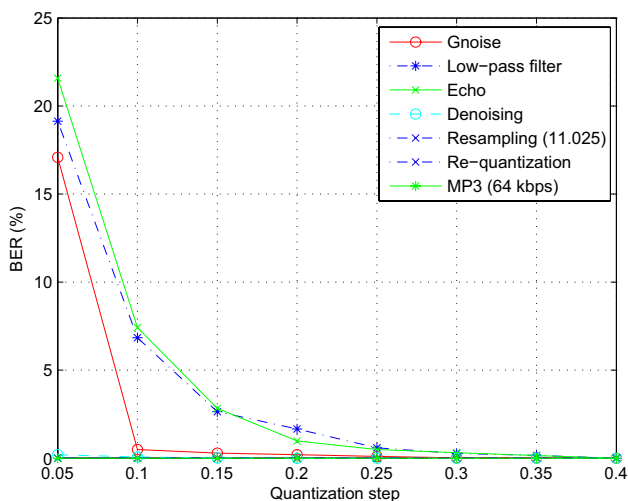
Fig. 9 BER under varying quantization steps (measures robustness)

given audio signal. So, $T_n = \lceil 0.75 \times r \rceil - 1$. In our experiment, we set BER = 25%. Then, the P_{fn} is written as

$$P_{fn} = \sum_{s=0}^{\lceil 0.75 \times r \rceil - 1} \binom{r}{s} P_e^s (1 - P_e)^{r-s}. \qquad (12)$$

Here, P_e is depend on the attacks. For different attacks, P_e has different values. From Tables 1 and 2, we have obtained that the BERs are all less than 0.1255 and hence P_e is taken as 0.8745 in our experiment. We obtain P_{fn} is equal to 1.67×10^{-88} from the Eq. 12 by putting $P_e = 0.8745$ and r = 1024. Figure 7 plots the false negative error probability for $r \in (0, 100]$. It is observed that the false negative error probability approaches to 0 when r is greater than 50. In Table 5, average NC values, average BER values (averages are computed over the data available in Tables 1, 2) and false negative error probability (P_{fn}) under audio attacks are shown. It is observed that P_{fn} values of our proposed technique are very close to 0 against different attacks.

Table 6 Performance of audio watermarking scheme sorted by data payload

Reference	Method	Payload (bps)	SNR (dB)	Blind	Robustness to MP3 compression (tested) (kbps)
Proposed	SVD-Quantization	196	24.83	Yes	48
Vivekananda Bhat et al. (2011b)	SVD–QIM	196	24.76	Yes	32
Lei et al. (2012)	LWT–SVD	170.67	24.17	Yes	64
Li et al. (2017)	DWT–QIM	172.39	24.26	Yes	64
Novamizanti et al. (2020)	LWT–DCT–SVD	172.26	24.65	Yes	32

Table 7 Result of robustness of different audio watermarking techniques on the 'Classical' audio signal

Attacks	Vivekananda Bhat et al. (2011b)		Lei et al. (2012)		Li et al. (2017)		Novamizanti et al. (2020)		Proposed method	
	NC	BER (%)	NC	BER (%)	NC	BER (%)	NC	BER (%)	NC	BER (%)
(1)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(2)	0.9965	0.3906	1.0000	0.0000	0.9356	7.1762	1.0000	0.0000	0.9430	6.3477
(3)	0.9957	0.4883	0.7892	22.9492	0.9947	0.5992	0.9651	5.3677	0.9974	0.2930
(4)	1.0000	0.0000	0.9998	0.0244	1.0000	0.0000	0.9988	0.1237	1.0000	0.0000
(5)	0.8688	14.6484	1.0000	0.0000	0.8648	15.0897	0.9195	9.2763	0.8917	12.1094
(6)	1.0000	0.0000	0.9991	0.0977	1.0000	0.0000	1.0000	0.0000	0.9991	0.0977
(7)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(8)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(9)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(10)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(11)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(12)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(13)	1.0000	0.0000	0.9954	0.5195	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(14)	1.0000	0.0000	0.9868	1.4687	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(15)	1.0000	0.0000	0.9803	2.1816	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

5.2 Quantization parameter

The quantization parameter Δ plays an important role in the performance of any quantization based audio watermarking technique. The two conflicting requirements of watermarking, i.e., imperceptibility and robustness are very much depend on the quantization parameter. As the value of Δ increases, the imperceptibility decreases. On the other hand, the robustness of the technique increases with the increased value of Δ. For effective audio watermarking technique, suitable Δ value to be selected experimentally. Figures 8 and 9 show the imperceptibility and robustness under different Δ respectively. From the experimental evaluation, we have obtained that the optimal range of Δ is [0.1, 0.3] and within this range the SNR is greater than 20 dB and BER is less than 0.1. For $\Delta \in [0.1, 0.3]$, from the above two figures, we may note that (i) in case of lower Δ, imperceptibility is high and robustness is low, (ii) when Δ is high, the imperceptibility is low and robustness is high. So, we consider the middle of the range as the trade-off and in this implementation we set $\Delta = 0.19$.

5.3 Performance comparison and discussion

In this section, the performance of the proposed watermarking technique is compared with state-of-the-art techniques. Here, we have compared the proposed method with four methods described in Vivekananda Bhat et al. (2011b), Lei et al. (2012), Li et al. (2017) and Novamizanti et al. (2020) with respect to robustness under different audio attacks, imperceptibility, data payload. The details of the experimental results for comparison among different methods are given in Table 6 and in Table 7. From Table 6, it is clear that proposed method achieve highest SNR with maximum payload. The robustness of five algorithms have tested against MP3 compression and results are shown in Table 6. The performance of the proposed against MP3 compression is better compare to two existing methods and slight less compare to other two methods. So, we may note that the performance of the proposed method is better than the existing methods. In Table 7, the robustness of different methods have compared and we observe that performance of the proposed method is equal or better than first three methods and has similar performance as the fourth method in terms of NC. The robustness respect to the BER, we note that a method is performs well compare to other methods against a particular attack whereas against other attack another method performs well. So, very close performance. Therefore, we may note that performance of the proposed method is comparable with the state-of-the-art methods. Here, we may also note that the proposed method used a very simple approach, SVD followed by quantization, to embed the watermark bit. Other existing methods have used SVD with some other complex

steps like QIM, LWT, DWT, LWT–DCT, etc. and that sense our method is less complex.

6 Conclusions

In this work, we have proposed a blind audio watermarking technique using SVD and quantization. The watermark is embedded by quantization of the LSV of the audio blocks. The proposed method provides a good trade-off among the imperceptibility, robustness and payload. The quantization parameter Δ and the FLT provide more security level. The experimental results demonstrate that proposed technique is robust against different audio attacks. The performance of the proposed method is similar or even better than the state-of-the-art methods. In addition, the proposed technique also provides low error probability rates. These results prove that the proposed watermarking technique can be used for copyright protection of audio signals. To further improve the performance, synchronization and error correcting codes may be considered within the proposed architecture.

References

Aslantas, V. (2009). Algorithm. *Optics Communications, 282*, 769–777.

Binny, A., & Koilakuntla, M. (2014). Hiding secret information using LSB based audio steganography. In *Proceedings of IEEE international conferences on soft computing and machine intelligence* (pp. 56–59).

Chetan, M., Bhat, P. P., Shet, V., Husenbhai, S. B., & Bhat, A. (2021). Audio watermarking using modified least significant bit technique. In *2021 International conference on circuits, controls and communications (CCUBE)* (pp. 1–5). IEEE.

Cox, I., Miller, M., Bloom, J., Fridrich, J., & Kalke, T. (2007). *Digital watermarking and steganography*. Morgan Kaufmann.

Dhar, P. K., & Shimamura, T. (2014). Audio watermarking in transform domain based on singular value decomposition and Cartesian-polar transformation. *International Journal of Speech Technology, 17*, 133–144.

Dhar, P. K., & Shimamura, T. (2015). Blind SVD-based audio watermarking using entropy and log-polar transformation. *Information Security and Application, 20*, 74–83.

Elshazly, A. R., Nasr, M. E., Fouad, M. M., & Abdel-Samie, F. S. (2017). High payload multi-channel dual audio watermarking algorithm based on discrete wavelet transform and singular value decomposition. *International Journal of Speech Technology, 20*, 951–958.

Elshazly, A. R., Nasr, M. E., Fouad, M., & Abdel-Samie, F. E. (2021). Intelligent high payload audio watermarking algorithm using colour image in DWT–SVD domain. *Journal of Physics: Conference Series, 2128*, 012019.

Ercelebi, E., & Batakci, L. (2009). Audio watermarking scheme based on embedding strategy in low frequency components with a binary image. *Digital Signal Processing, 19*(2), 265–277.

Farzaneh, M., & Toroghi, R. M. (2020). Robust audio watermarking using graph-based transform and singular value decomposition. In *2020 10th International symposium on telecommunications (IST)* (pp. 137–141). IEEE.

Ghosal, A. (2014). A hierarchical approach for content based audio classification. PhD Thesis, Jadavpur University.

Hu, P., Peng, D., Yi, Z., & Xiang, Y. (2016). Robust time-spread echo watermarking using characteristics of host signals. *Electronics Letters, 52*(1), 5–6.

Lai, C. C. (2011). An improved SVD-based watermarking scheme using human visual characteristics. *Optics Communications, 284*, 938–944.

Lei, B. Y., Soon, I. Y., & Li, Z. (2011). Blind and robust audio watermarking scheme based on SVD–DCT. *Signal Processing, 91*(8), 1973–1984.

Lei, B. Y., Soon, I. Y., Zhou, F., Li, Z., & Lei, H. (2012). A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition. *Signal Processing, 92*, 1985–2001.

Li, J., Wang, H. X., Wu, T., Sun, X., & Qian, Q. (2017). Norm ratio-based audio watermarking scheme in DWT domain. *Multimedia Tools and Applications, 77*, 1–17.

Li, J., & Wu, T. (2015). Robust audio watermarking scheme via QIM of correlation coefficients using LWT and QR decomposition. In *Proceedings of IEEE international conferences on informative and cybernetics for computational social systems* (pp. 1–6).

Lie, W. N., & Chang, L. C. (2006). Robust and high-quality time-domain audio watermarking on low-frequency amplitude modification. *IEEE Transaction on Multimedia, 8*(1), 46–59.

Mishra, M., Mishra, P., Adhikary, M. C., & Kumar, S. (2012). Image encryption using Fibonacci–Lucas transformation. *International Journal on Cryptography and Information Security, 2*(3), 131–141.

Muhammad, N., & Bibi, N. (2015). Digital image watermarking using partial pivoting lower and upper triangular decomposition into the wavelet domain. *IET Image Processing, 9*(9), 795–803.

Novamizanti, L., Budiman, G., & Astuti, E. N. F. (2020). Robust audio watermarking based on transform domain and SVD with compressive sampling framework. *TELKOMNIKA Telecommunication Computing Electronics and Control, 18*(2), 1079–1088.

Patil, M., & Chitode, J. (2021). SVD based audio watermarking algorithm using dual watermark for copyright protection. *New Visions in Science and Technology, 5*, 107–120.

Rasti, P., Samiei, S., Agoyi, M., Escalera, S., & Anbarjafari, G. (2016). Robust non-blind color video watermarking using QR decomposition and entropy analysis. *Visual Communication and Image Representation, 38*, 838–847.

Roy, S., Sarkar, N., Chowdhury, A. K., & Iqbal, S. M. A. (2015). An efficient and blind audio watermarking technique in DCT domain. In *Proceedings of IEEE international conferences on computer and information technology* (pp. 362–367).

Subbarayan, S., & Ramanatha, S. K. (2009). Effective watermarking of digital audio and image using MATLAB technique. In *Proceedings of IEEE international conferences on machine vision* (pp. 317–319).

Vivekananda Bhat, K., Sengupta, I., & Das, A. (2010). An adaptive audio watermarking based on the singular value decomposition in the wavelet domain. *Digital Signal Processing, 20*, 1547–1558.

Vivekananda Bhat, K., Sengupta, I., & Das, A. (2011a). An audio watermarking scheme using singular value decomposition and dither-modulation quantization. *Multimedia Tools and Applications, 52*, 369–383.

Vivekananda Bhat, K., Sengupta, I., & Das, A. (2011b). A new audio watermarking scheme based on singular value decomposition quantization. *Circuits, Systems and Signal Processing, 30*, 915–927.

Wu, S., Huang, J., Huang, D., & Huang, Y. Q. (2005). Efficiently self-synchronized audio watermarking for assured audio data transmission. *IEEE Transactions on Broadcasting, 51*(1), 69–76.

Zhang, J. (2015). Audio dual watermarking scheme for copyright protection and content authentication. *International Journal of Speech Technology, 18*, 443–448.