ECE F434 Project Report: Audio Watermarking

Meghadri Ghosh - 2023A3PS0314P
Pramit Pal - 2023AAPS0765P
Pranav Chandra N. V. - 2023AAPS0013P

1 Approach

The reference paper by Yamni et al. combines three transforms to encrypt and embed an image as a binary watermark into the low frequency components of an audio signal. We replicate their method and analyze its performance under various attacks. (**Github Repo**)

We embed a binary watermark using chaotic encryption, wavelet decomposition and Tchebichef moment modulation. We begin with the chaotic encryption, MLNCML, where we use a map defined by $\epsilon = 0.4, \mu = 3.99$ to generate a chaotic bit-plane H_b. The watermark W is then encrypted using exor operation:

$$W' = W \oplus H_b. \tag{1}$$

The audio is simultaneously divided into $L_{\text{seg}} = \left\lceil \frac{NM}{L_1} \right\rceil = 256$ segments, and each undergoes a 3-level Haar DWT:

$$\text{DWT}^3(\text{segment}) \to (A_3, D_3, D_2, D_1), \tag{2}$$

” ”

with embedding performed in the low-frequency block A_3. This block is split into $L_1 = 4$ sub-blocks, and each is transformed using DTMT with $K = \min(64, L_2/2)$ Tchebichef orders:

$$M = T\,\text{pa3}, \qquad T \in \mathbb{R}^{K \times L_2}. \tag{3}$$

Moments are separated as even and odd components, and a bit b is embedded by adjusting their norms. With

$$\bar{\sigma} = \frac{\|M_1\| + \|M_2\|}{2}, \qquad \delta = 0.05, \tag{4}$$

the modified norms are

$$(\sigma_1', \sigma_2') = \begin{cases} (\bar{\sigma} + \delta,\ \bar{\sigma} - \delta), & b = 1, \\ (\bar{\sigma} - \delta,\ \bar{\sigma} + \delta), & b = 0. \end{cases} \tag{5}$$

Reconstruction uses the inverse DTMT,

$$\text{pa3}' = T^T M', \tag{6}$$

followed by inverse DWT to obtain the watermarked segment. During extraction, the bit is detected by comparing norms:

$$\hat{b} = \mathbf{1}(\|M_1\| > \|M_2\|)\,, \tag{7}$$

and the final watermark is recovered via

$$\hat{W} = \hat{W}' \oplus H_b. \tag{8}$$

Figure 3 in the appendix illustrates the complete enxryption and decryption pipeline.

2 Testing and Results

2.1 Testing

The paper proposes a few different kinds of attacks that they tested their watermarking system against. We used a few of these tests(1-4) as well as proposed some new ones(5-7) in order to extend the robustness analysis of the watermarking system.

1. **Cropping by 20%:** We remove 20% of random audio samples from the watermarked audio to simulate random data loss.

2. **Gaussian Noise of $20dB$:** We add white Gaussian noise to the watermarked audio such that the resulting SNR is 20 dB.

3. **High-Pass Filter at 100Hz:**

4. **Low-Pass Filter at 4kHz:**

5. **MP3 compression at 32, 64 and 128 kbps:** To simulate lossy compression, we encode and decode the watermarked audio using MP3 compression.

6. **Random Time Scaling by ±10%:** We randomly speed up or slow down the watermarked audio by up to 10%. ””

7. **Re-recording:** We simulate re-recording by playing the watermarked audio through speakers and capturing it with a microphone.

Table 1 summarizes the results of these attacks on the audio and watermark. The SNR and BER

Attack	**SNR (dB)**	**SI-SDR**	**BER**	**NC**	**LMSE**	**SSIM**
No Attack	7.734	6.933	0	1	0	1
20% Crop	4.749	2.977	0.104	0.792	1.808	0.263
Gaussian 20dB	7.525	6.683	0.004	0.993	0.043	0.989
HPF 100Hz	1.742	-2.868	0.322	0.356	3.749	0.159
LPF 4kHz	6.808	5.794	0.0199	0.960	0.380	0.598
MP3 32kbps	6.669	5.643	0.180	0.640	2.811	0.181
MP3 64kbps	7.379	6.538	0.075	0.850	1.317	0.357
MP3 128kbps	7.662	6.909	0.001	0.998	0.015	0.973
Time Scale ±10%	-1.457	-38.571	0.499	0.002	5.313	0.002
Re-recording	1.842	-0.274	0.483	0.035	5.262	0.008

Table 1: Attack robustness metrics: SNR (audio quality), BER (bit errors), NC (pattern correlation)

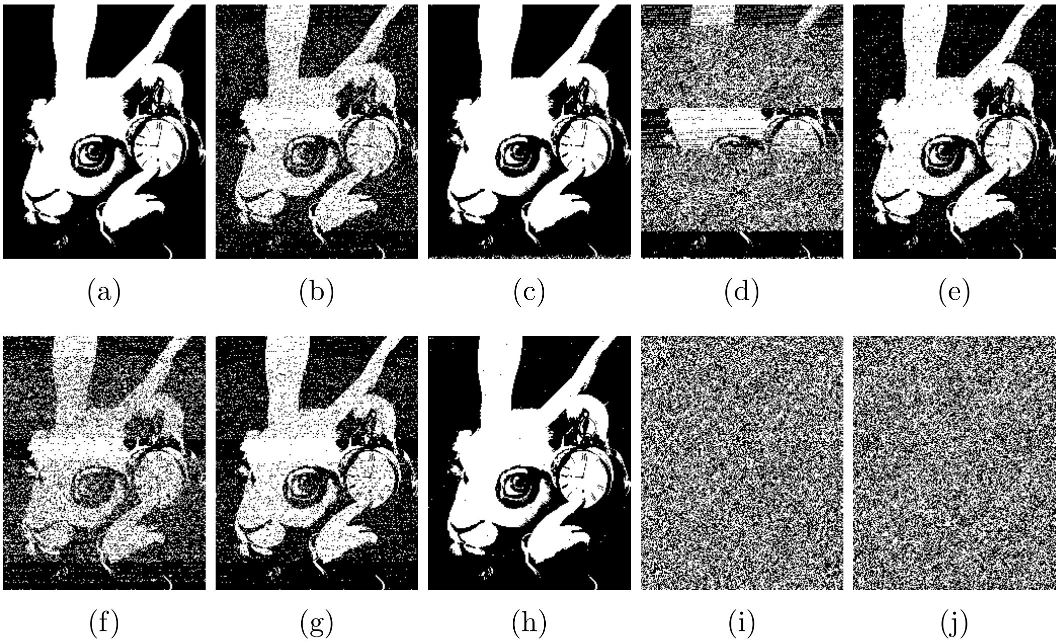
2.2 Results

In order to effectively demonstrate the watermarking, the original and encrypted watermarks are shown in Figure 1.

Figure 1: Original watermark (left) and extracted watermark with no attack (right)

You would note that the structure of the extracted image is very similar, with even fine details like the numbers on the dial being preserved.

Now, the attacked images are shown in figure 2.



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

(f) (g) (h) (i) (j)

Figure 2: (a) No attack extraction (b) 20% Crop (c) Gaussian Noise 20dB (d) HPF 100Hz (e) LPF 4kHz (f) MP3 32kbps (g) MP3 64kbps (h) MP3 128kbps (i) Random Time Scaling ±10% (j) Re-recording

From the results, we see that this watermarking method is very effective against nearly all attacks, save for re-recording and time scaling. However, an important note is that the audio quality is significantly degraded in these two cases, as can be seen from the very low SNR and SI-SDR values in Table 1. This means that the low performance of the watermarking method is not the primary issue, as pirates or other opposition parties would be unlikely to use such low-quality audio.

Appendix: System Architecture

Figure 3 shows how the three transforms work together. The watermark flows through encryption, gets split across audio segments, and embeds into frequency-domain moments. Extraction reverses the process.

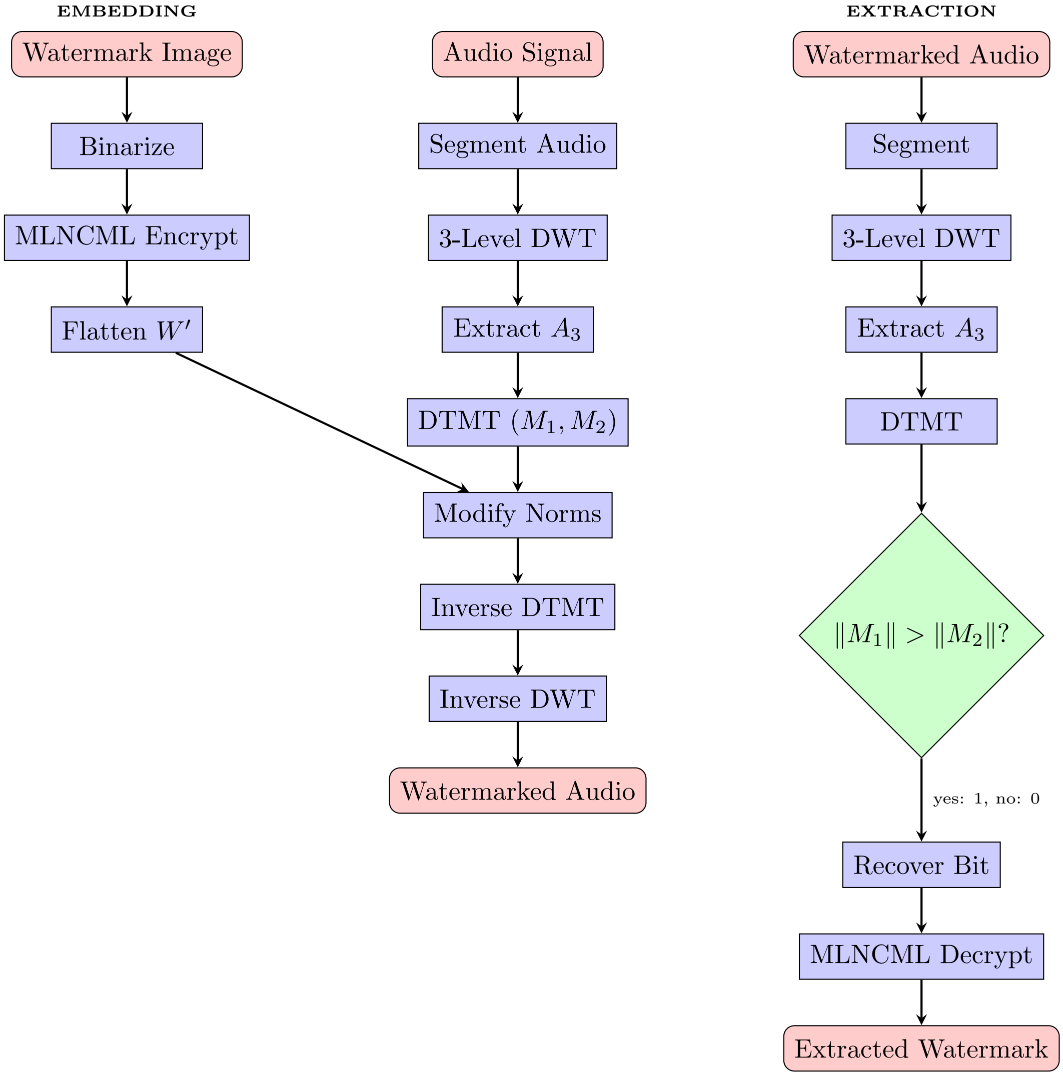


Figure 3: Watermarking system flowchart showing embedding (left/center) and extraction (right) pipelines