



MP3 Robust Digital Watermarking

By:

Dobrovinsky Ron

Lavetman Stella

Supervised by:

Ofir Hadas

June 2005

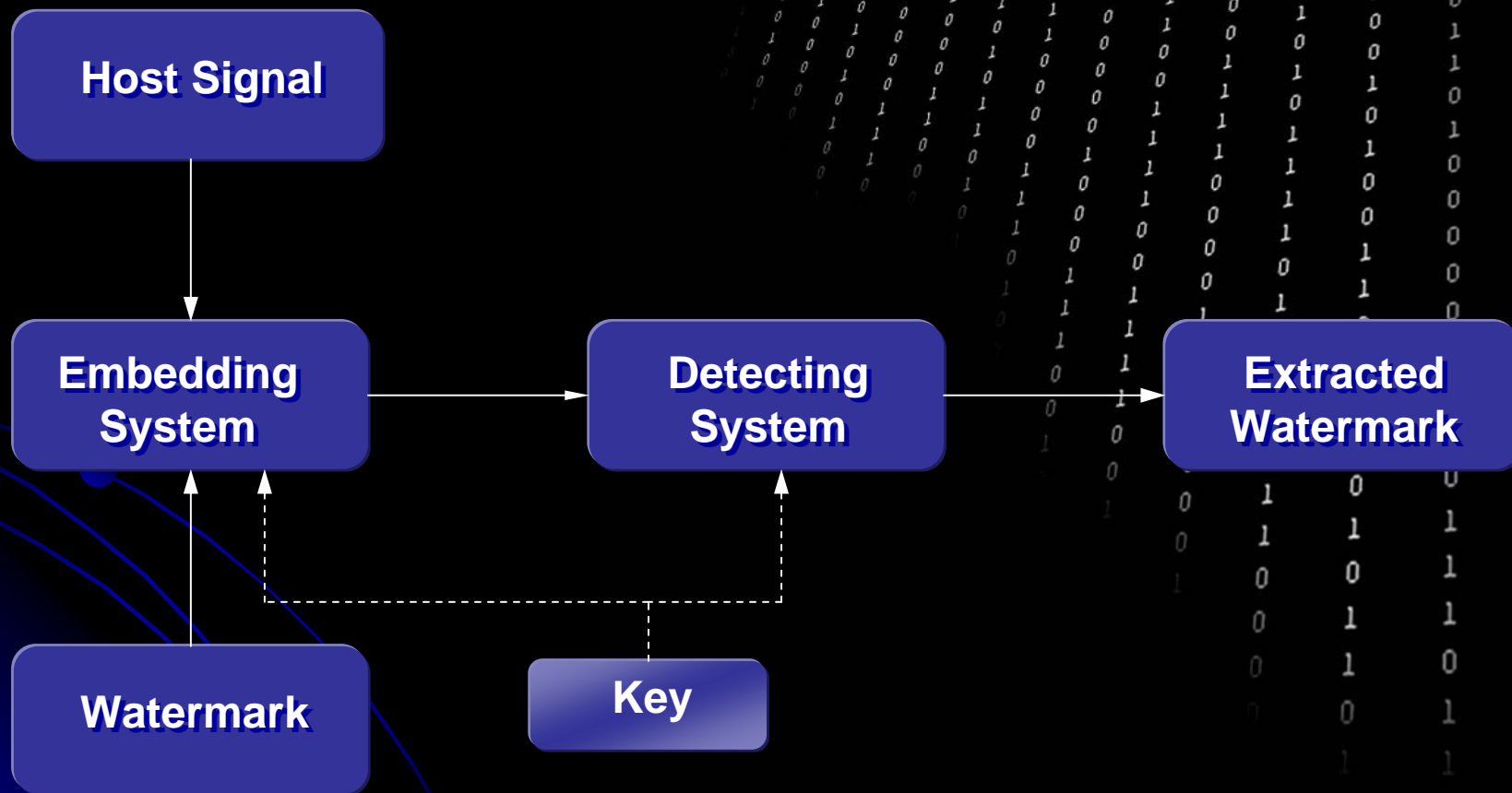
What is Digital Watermarking?

- Digital watermarking/Data Hiding is the practice of embedding data in various digital media types: Images, Audio, Video, Text, etc.
- Provides potential solution to the issue of copyright protection, annotation & identification.

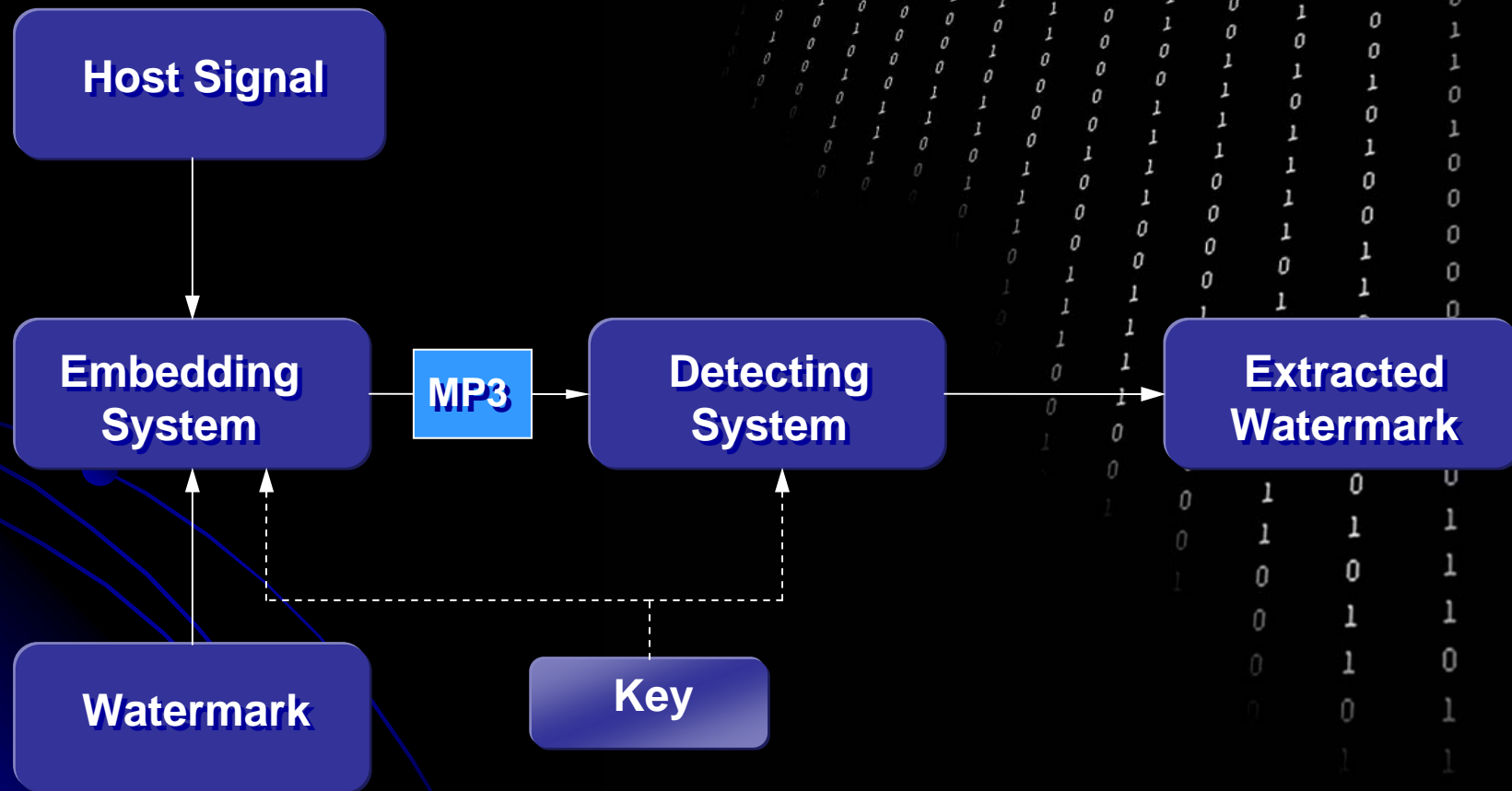
DWM Requirements

- Message embedded within the data itself
- Knowing the watermarking algorithm does not enable to remove the watermark
- Imperceptible, Inaudible watermark
- Undetectable (recommended)
- Resistance to distortions
(A/D, **Lossy Compression (MP3)**, noise, etc.)

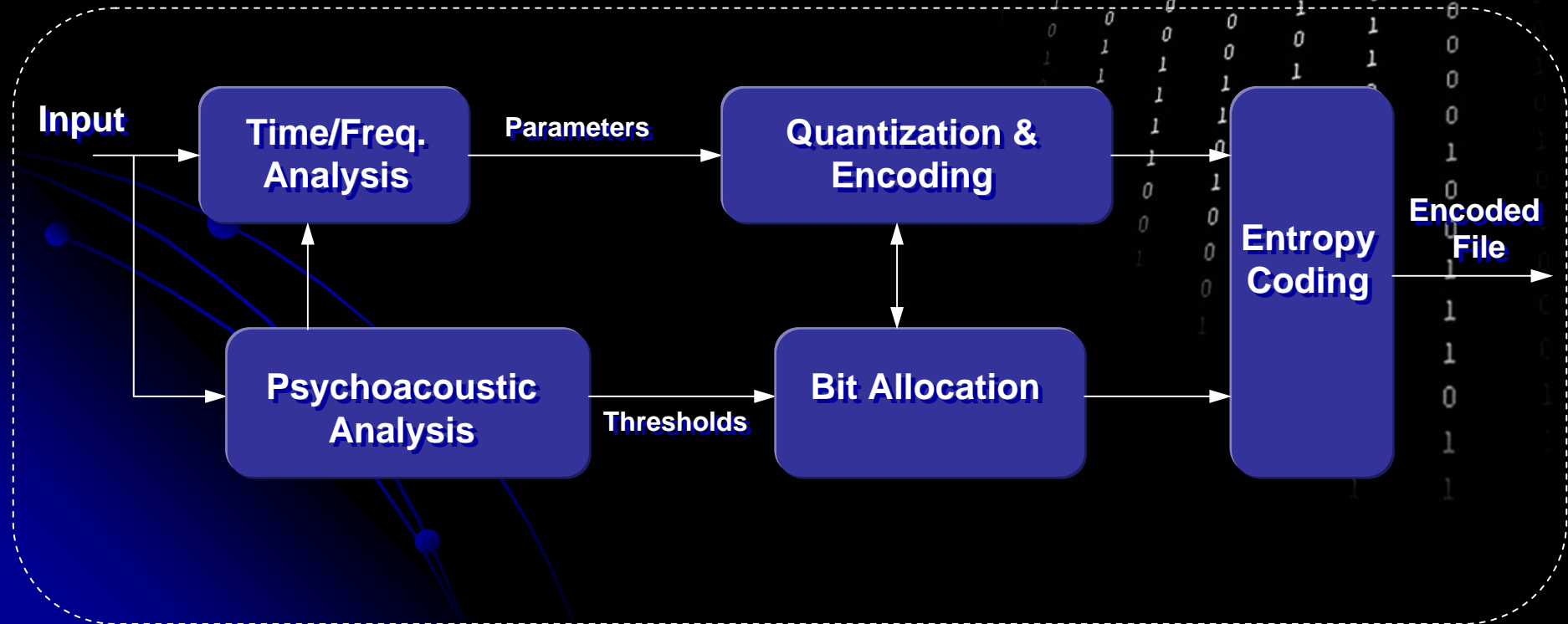
General Watermarking Scheme



Our Goal

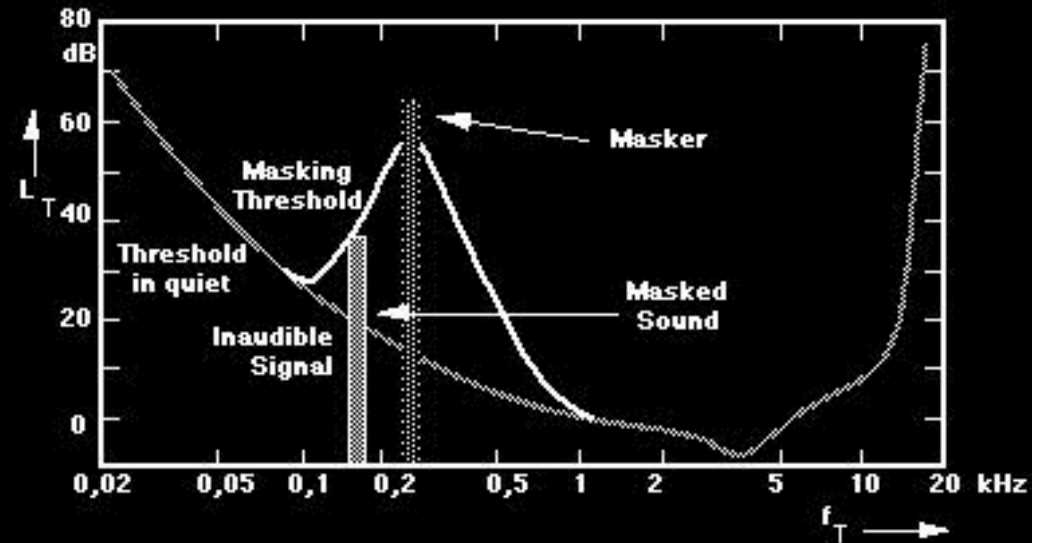


MP3 Coding Scheme



Psychoacoustic Model

- Using the limitations of the hearing system in order to get unperceivable watermarking
- Masking thresholds are calculated & data is being embedded as pseudo-random noise, below hearing threshold.

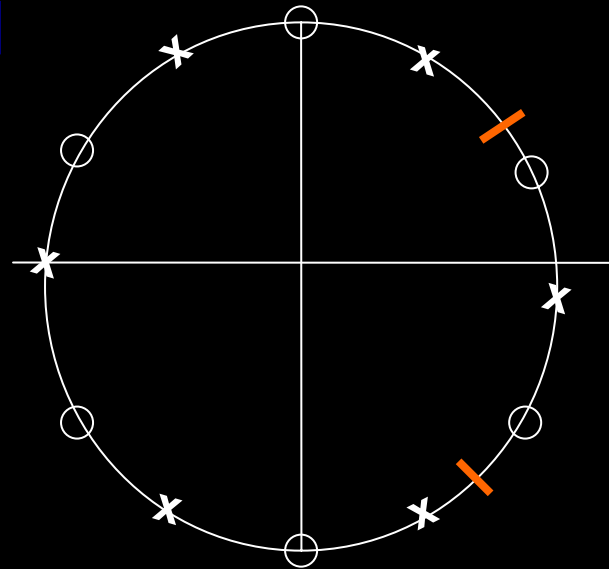


Psychoacoustic Model – Phase

- Phase is thought to be less perceivable than amplitude.
- Humans are more sensitive to phase continuity than to its real values.
- MP3 compression does not affect phase directly – a compression “Hole” we can make use of.

Quantization Index Modulation - QIM

- The signal is divided into segments, proportional to the length used by encoder
- Each segment is transformed to the DFT domain
- Phase and amplitude are extracted
- Phase is being quantized according to the following scheme:



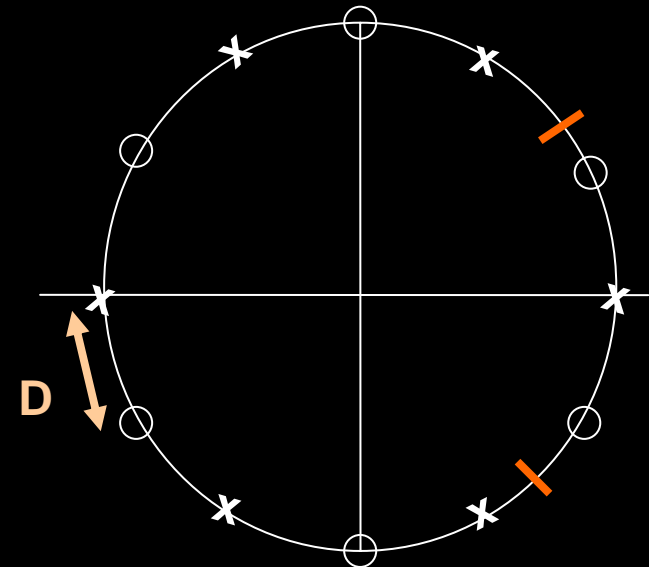
QIM

- Information is being represented by using quantization patterns :

(xoxo) \equiv bit 0

(oxox) \equiv bit 1

- $D \equiv$ distance (x point ,o point)
 - Fine quantization for better sound quality
 - Rough quantization for better watermark preservation



QIM

- Decoding:

→ Find the nearest $(x,0)$ points (a_i, b_i) to given phase φ_i

→ Construct two patterns

$$u = (a_0, b_1, a_2, b_3)$$

$$v = (b_0, a_1, b_2, a_3)$$

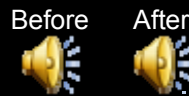
→ If $\sum r_i \cdot (u_i - \varphi_i)^2 < \sum r_i \cdot (v_i - \varphi_i)^2$ the bit is 1, else the bit is 0.

→ Extending pattern size yields more robust watermark

ODG and Error

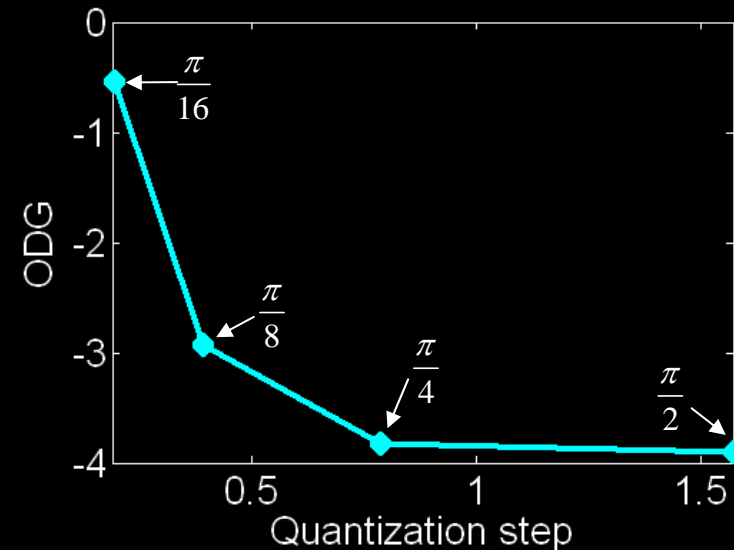
Example Parameters:

- Input Wav: Kern.wav
- Encoder: MP3 (LAME)
- Mp3 bitrate: 128 kbps
- Pattern size: 4
- Watermark bitrate: 0.4 Kbps

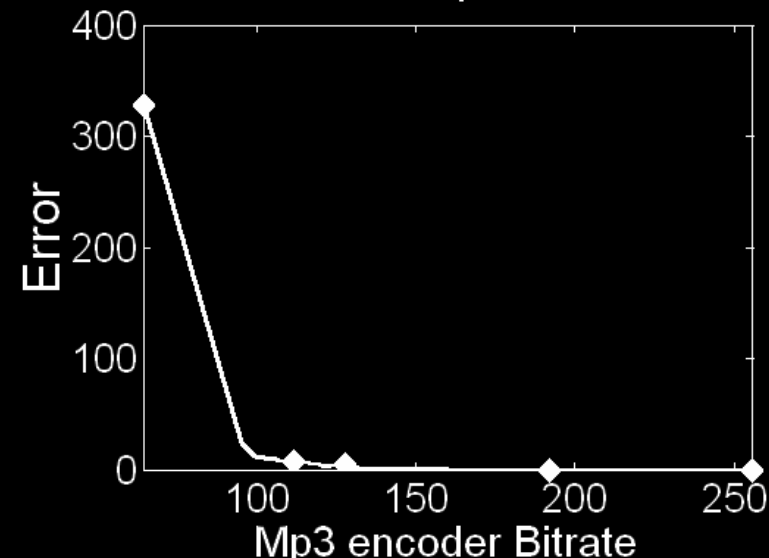


- Objective Difference Grade increases with finer quantization
- Error bitrate increases with finer quantization

ODG Vs. Quantization step



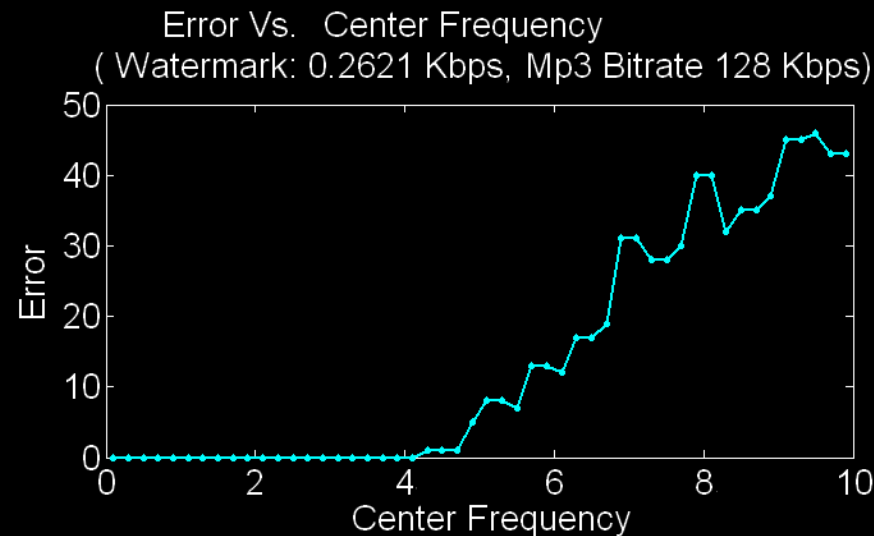
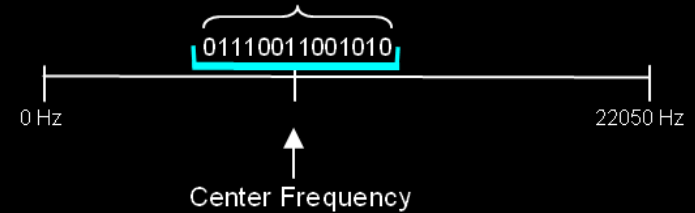
Error for 0.38 Kbps data hidden



More Results Evaluation

Influence of embedding frequencies:

Embedding bandwidth : 1.225 kHz



- Behavior suits the hearing threshold model
- MP3 damages the watermark & the quality according to the threshold

Robustness to New Encoders

- Embedding system shown good results with some more complex & improved perceptual coders:
 - Mpeg 4 AAC
 - Ogg Vorbis
 - MPC – Musepack
- Similar error bitrate performance to MP3 although the AAC, Ogg and MPC use more advanced encoding techniques

GUI Application

Input Parameters:

- Input File
- Input Watermark
- Start Frequency
- Frame Length
- Quantization Step
- Pattern Size

Output Parameters:

- Watermarked File
- Extracted Watermark
- ODG
- Bit Errors
- DWM Bitrate

The screenshot displays the Cryptic application interface. At the top, the title bar reads 'Cryptic' and the menu bar includes 'File' and 'Edit'. The main window features a spectrogram on the left with the text 'Embedding Range: 1.8375 kHz out of 44.1 kHz'. To the right of the spectrogram are four dropdown menus for input parameters: 'Start Frequency: 4 kHz', 'Frame Length: 576', 'Quantization Step: pi/8', and 'Pattern Size: 4'. Below these are two buttons: 'Embed Watermark' and 'Extract Watermark'. The interface is divided into two columns of text, each containing a watermarking example. The right column shows the output parameters: 'Channel 1' (selected), 'Channel 2', 'Stereo', and 'Mono Output' (unchecked). Below these are 'in' and 'out' buttons. The 'ODG' section shows '-2.92' and a 'Calculate' button. The 'Bit Errors' section shows '0 / 4088' and a 'Calculate' button. The 'DWM Bitrate' section shows '~ 0.40807 kbps'. A small 'SIPL' logo is visible in the bottom right corner. A modal dialog box is open in the center, titled 'Extracting Watermark. Please Wait...' with a progress bar at 59%.

GUI Application

- Embedding any text information
- Stereo Files - Embedding different DWM in each channel
- Different Codecs available
- Performance analysis easily calculated

Examples

Before



After



Kevin_Kern.Wav

- Encoder: MP3 (LAME)
- Sample Length : 10 sec
- Mp3 bitrate: $160 \text{ kbps} \frac{\pi}{8}$
- Quantization Step :
- Watermark bitrate: 0.48 Kbps

ODG : -0.62

Errors : 9/4088

Before



After



Stevie Wonder_As.Wav

- Encoder: MP3 (LAME)
- Sample Length : 8 sec
- Mp3 bitrate: $128 \text{ kbps} \frac{\pi}{8}$
- Quantization Step :
- Watermark bitrate: 0.48 Kbps

ODG : -1.57

Errors : 0/4088

The End

