Audio Steganography: Implementation and Performance Analysis

Nithish Ravikkumar

PHC-303 Course Project

October 31, 2025

Outline

Audio Steganography: An Introduction

- Audio steganography involves hiding information within audio files so that the message is imperceptible to listeners.
- It exploits redundant or perceptually insignificant parts of an audio signal to embed data.
- The goal is to ensure both imperceptibility and robustness against signal processing operations.
- Common applications include secure communication, watermarking, and copyright protection.

Distinguishing Steganography and Cryptography

- **Cryptography** transforms data into an unreadable form using keys; the presence of secret data is obvious.
- **Steganography** hides the very existence of the secret, embedding data covertly within innocuous cover media.
- Combining both enhances confidentiality and secrecy.

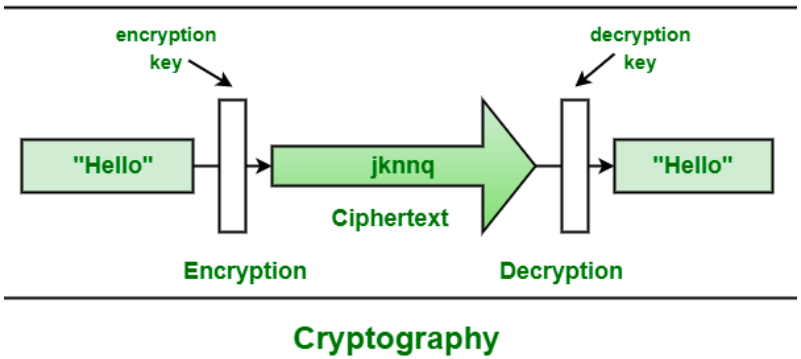

Figure: Cryptography

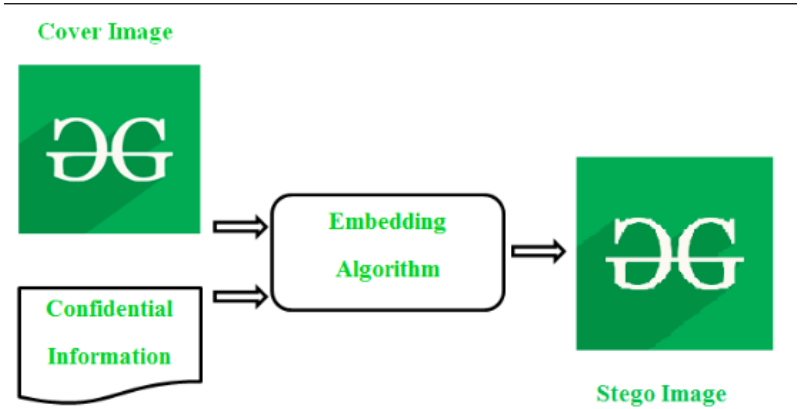

Figure: Steganography

Digital Image and Audio Representation

- **Images:** Represented as a grid of pixels, each pixel carrying intensity values in one or multiple color channels (e.g., RGB).

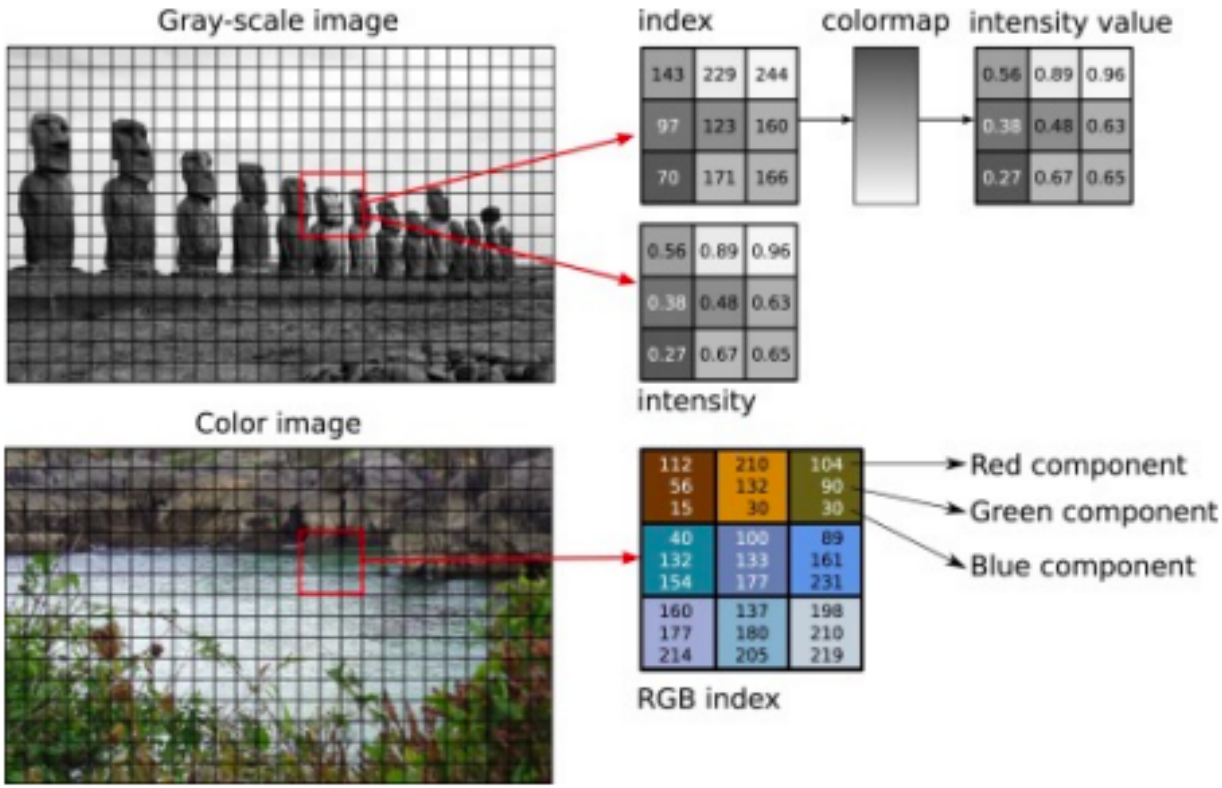

Figure: Pixelated Image representation

- **Audio:** Discrete time signals sampled at frequency f_s with quantized amplitude $x[n]$.

$$x[n] = x(nT), \quad T = \frac{1}{f_s}$$

- Bit-depth defines the resolution of the amplitude values, typically 16 bits per sample.

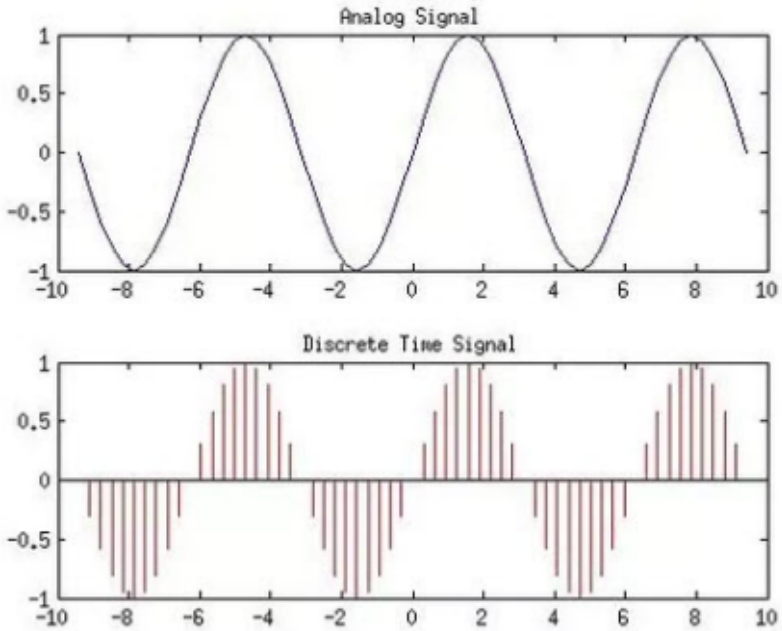

Figure: Pulse modulation

Signal Domains in Audio Processing

- **Time Domain:** Signal amplitude variation over time.

$$x[n]$$

- **Frequency Domain:** Decomposition into sinusoids via Fourier transform.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}}$$

- Frequency domain analysis reveals characteristics like pitch, timbre.
- Transforms used in steganography exploit these domains for embedding data with improved invisibility and robustness.

Discrete Cosine Transform (DCT)

DCT Equation

$$X_k = \alpha(k) \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

where

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & k > 0 \end{cases}$$

- Basis for many compression and watermarking algorithms.
- Compacts signal energy into few coefficients, simplifying embedding.

Fast Fourier Transform (FFT)

FFT Equation

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1$$

- Efficient algorithm for Discrete Fourier Transform (DFT).
- Provides magnitude $|X_k|$ and phase $\angle X_k$ components.
- Phase information used in phase coding steganography.

LSB Method (Text, Audio, Image)

- Least Significant Bit (LSB) modification replaces insignificant bits with payload bits.

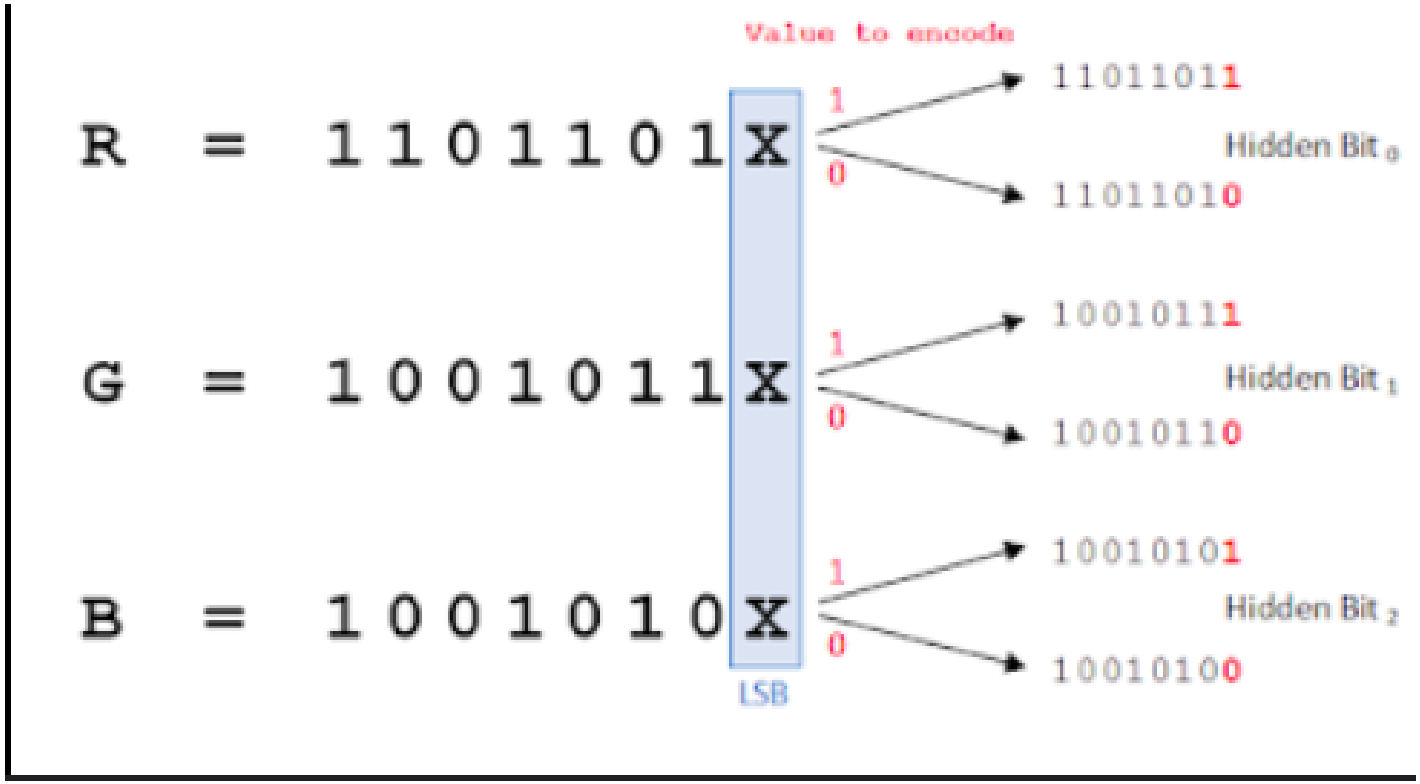

Figure: LSB encoding

Example - Img2Img Encoding

Figure: Secret Image

Figure: Base Image

Figure: Resultant image with Hidden message

DCT-Based Method

- Audio divided into frames, DCT applied to each.
- Embedding by changing sign of select DCT coefficient:

$$X'_k = \begin{cases} |X_k| & \text{if } m_i = 1 \\ -|X_k| & \text{if } m_i = 0 \end{cases}$$

- Signal reconstructed using inverse DCT.
- More robust embedding with moderate payload capacity.

Example: Txt2Audio LSB, DCT Encoding

Input Audio - 12s, 441.1 KHz Sampling Rate, 2.1MB WAV format file.
Input Text - "This is the secret message : King kong is actually alive"
.txt file, 49B

- **2-bit and 8-bit LSB Encoding**
  - Hides text bits in the least significant bits of audio samples
  - 2-bit: Modifies 2 LSBs/sample
  - 8-bit: full byte modification
- **DCT-Based Encoding**
  - Embeds bits by modifying sign of Discrete Cosine Transform coefficients
- The demo showcases both approaches for effective data hiding in audio.

Phase Coding Method

- Frame-wise FFT of signal performed.
- Payload bits embedded by setting phase of frequency component:

$$\angle X'_k = \begin{cases} 0 & \text{if } m_i = 1 \\ \pi & \text{if } m_i = 0 \end{cases}$$

- Signal reconstructed via inverse FFT.
- Maintains perceptual transparency but sensitive to noise.

BER and Signal Attacks

Bit Error Rate (BER)

- BER measures the percentage of bits that were incorrectly recovered after hiding and extracting the payload.
- $$\text{BER} = \frac{\sum_{i=1}^{N}(m_i \oplus \hat{m}_i)}{N} \times 100\%$$

  where m_i is original bit, $\hat{m}_i$ is recovered bit.
- Lower BER means better recovery accuracy.

Gaussian Noise Attack

- Random noise is added to the audio, similar to static or hiss.
- Makes the signal noisier, causing some bits to flip and increase BER.
- $$y[n] = x[n] + w[n], \quad w[n] \sim \mathcal{N}(0, \sigma^2)$$

Low-pass Filtering Attack

- Removes high-frequency components from the audio.
- Can distort embedded data by smoothing abrupt changes.
- $$y[n] = \sum_{k=0}^{M-1} b_k x[n-k]$$

  where b_k are filter coefficients.

Bit Depth Reduction Attack

- Reduces precision of audio samples by lowering number of bits used.
- Causes detail loss by rounding amplitude values to fewer levels, similar to compressing an image.
- $$y[n] = \left\lfloor \frac{x[n]}{Q} \right\rfloor \times Q$$

  where $Q = 2^{16-b}$ and b is bits retained.

Performance Comparison

Method	BER (%)
LSB	Original: 0.0 Gaussian Noise: 49-50 Lowpass Filter: 48-50 Bit Depth Reduction: 40-50
DCT	Original: 0.21 Gaussian Noise: 13.12 Lowpass Filter: 13.54 Bit Depth Reduction: 4.79
Phase Coding	Original: 40.0 Gaussian Noise: 42.92 Lowpass Filter: 41.67 Bit Depth Reduction: 42.29

Conclusions and Future Work

- DCT method yields best robustness with low BER under attacks.
- LSB provides ideal invisibility but low robustness.
- Phase coding requires advanced error handling to reduce BER.
- Future directions include:
  - Error correction in Decoding to prevent data loss.
  - Integration with Cryptography for Security
  - Hybrid domain approaches