---

**Revised Report for Exercise 3: Audio Steganography**

**Introduction**

Audio steganography combines the science of signal processing with the art of concealing data in audio files, ensuring secure and imperceptible communication. Exercise 3 focused on three distinct tasks: detecting hidden ultrasonic data in audio files, embedding and extracting secret messages using steganography, and exploring advanced methods in the field. This report provides a detailed account of the implementation, results, and insights gained from these tasks.

---

**Task 3.1: Audio File Analysis**

This task aimed to identify which of four suspicious audio files contained hidden data encoded in the ultrasonic frequency range and make it audible.

**Implementation**:

- The analysis was conducted using Python, leveraging Fourier Transform to convert audio signals into the frequency domain.
- Ultrasonic frequencies (above 20,000 Hz) were isolated for each file, and their total power was calculated. The file with the highest ultrasonic power was marked as suspicious.
- To reveal the hidden data, the ultrasonic frequencies were shifted into an audible range. This was achieved by modifying the frequency indices and reconstructing the time-domain signal using the Inverse Fourier Transform.

**Results**: The analysis successfully identified the file containing the hidden code. The processed audio, when played back, revealed an intelligible secret code. This validated the effectiveness of the ultrasonic frequency detection and shifting techniques.

**Reflections**: This task underscored the importance of frequency domain analysis in signal processing. It also highlighted how seemingly imperceptible frequencies can carry meaningful information, which can be recovered with the right tools.

---

**Task 3.2: Embedding and Extracting Hidden Messages**

In this task, a custom Python application was developed to embed and extract hidden messages using the Least Significant Bit (LSB) steganography technique.

**Implementation**:

1. **Message Embedding**:
    - The message, "An eye for an eye makes the whole world blind," was converted into its binary representation.

- Using a random seed for index selection, the binary data was embedded into the LSBs of randomly chosen audio samples. This non-consecutive embedding increased security and reduced predictability.
- The system allowed flexibility by enabling multiple LSBs to be used for embedding, enhancing capacity while maintaining imperceptibility.

2. **Message Extraction**:
    - The random indices and LSB configuration were reused to retrieve the embedded data.
    - The extracted binary message was then converted back into text, ensuring accurate recovery.

**Results**: The application performed flawlessly. The embedded message was imperceptible to human hearing, and the extraction process accurately retrieved the original message. This demonstrated the reliability and practicality of LSB steganography for secure data hiding.

**Reflections**: While LSB steganography is straightforward and effective, its vulnerability to noise and compression became apparent. Any modifications to the audio signal could compromise the hidden data, which is a critical consideration for real-world applications.

---

**Task 3.3: Investigation into Advanced Audio Steganography**

Beyond the implementation tasks, this investigation explored **Phase Coding**, an advanced audio steganographic technique.

**Overview**: Phase Coding manipulates the phase spectrum of audio signals to embed data. Unlike LSB, which alters amplitude values, Phase Coding exploits the human ear's insensitivity to small phase changes, making it highly imperceptible.

**Key Features**:

- **Imperceptibility**: Changes in the phase spectrum are inaudible, ensuring high audio quality.
- **Robustness**: It is resistant to common audio manipulations like compression (e.g., MP3) and noise.
- **Security**: The statistical detectability of Phase Coding is lower compared to LSB, providing enhanced stealth.

**Comparison with LSB**:

- While LSB is easy to implement and supports higher data capacity, it is vulnerable to noise and compression.
- Phase Coding is more secure and robust but computationally intensive and limited in capacity.

**Conclusion**: Phase Coding emerges as a superior choice for high-security applications where audio quality and resistance to attacks are paramount. However, for scenarios requiring large data capacity or quick implementation, LSB remains a practical option.

---

**Conclusion**

Exercise 3 provided a comprehensive exploration of audio steganography, encompassing practical implementation and theoretical investigation.

- **Task 3.1** highlighted the potential of ultrasonic frequencies to conceal data and demonstrated techniques for detection and recovery.
- **Task 3.2** showcased the effectiveness of LSB steganography for embedding and extracting messages while emphasizing its vulnerabilities.
- **Task 3.3** offered insights into advanced methods like Phase Coding, contrasting their strengths and weaknesses against LSB.

Through these tasks, I gained valuable experience in signal processing, cryptography, and secure communication techniques. This exercise not only deepened my understanding of steganography but also reinforced the importance of balancing imperceptibility, robustness, and capacity in designing secure systems.