from scipy.io.wavfile import read

import matplotlib.pyplot as plt

rate, data = read("encrypted.wav")

print(f"Sample Rate: {rate}")

print(f"Number of Samples: {len(data)}")

print(f"First 10 Samples: {data[:10]}") # Check a few samples

plt.figure(figsize=(10, 6))

plt.plot(data[:10000]) # Plot the first 10,000 samples for visualization

plt.title("Encrypted Waveform")

plt.xlabel("Sample Index")

plt.ylabel("Amplitude")

plt.show()A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

import numpy as np

from scipy.fftpack import fft

import matplotlib.pyplot as plt

frequencies = np.abs(fft(data))

plt.figure(figsize=(10, 6))

plt.plot(frequencies[:5000]) # Plot the first 5000 frequency components

plt.title("Frequency Spectrum")

plt.xlabel("Frequency Index")

plt.ylabel("Amplitude")

plt.show()

dominant\_indices = np.argsort(-frequencies)[:10] # Get the 10 most significant frequencies

print("Dominant Frequencies (Indices):", dominant\_indices)

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

import numpy as np

from scipy.fftpack import fft

from scipy.io.wavfile import read

# Step 1: Load the encrypted WAV file

rate, data = read("encrypted.wav")

# Step 2: Perform FFT on the data

frequencies = np.abs(fft(data))

# Step 3: Extract dominant frequencies

# Find the indices of the 10 most significant frequencies

dominant\_indices = np.argsort(-frequencies)[:10] # Top 10 frequencies

dominant\_amplitudes = frequencies[dominant\_indices] # Get their amplitudes

# Convert indices to frequencies

dominant\_frequencies = dominant\_indices \* rate / len(data) # Frequency in Hz

# Print results

print("Dominant Frequencies (Hz):", dominant\_frequencies)

print("Corresponding Amplitudes:", dominant\_amplitudes)

**Map Frequencies to Characters**

The encryption script (encrypt.py) uses a sine wave with the frequency proportional to (i + 1) for each character in the flag. Use the dominant frequencies to reverse this process.

A white paper with black text

Description automatically generated

A computer screen shot of a number

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

HTB{mY\_f4v0UR1t3\_tr4nSF0rM\_-\_f0urIEr!!}