Esteganografía de Audio — Práctica

Audio original: 10 Mulatada Audio.wav

Audio con mensaje oculto: 10 Mulatada Audio - con mensaje.wav

Objetivo

Ocultar el mensaje “que lindo es el tango” dentro de un archivo de audio .wav utilizando esteganografía por modificación de los bits menos significativos (LSB).

Herramientas utilizadas

- Python 3
- Módulo estándar wave
- Editor: VSCode / Notepad++ (o similar)
- Archivo de audio .wav en formato PCM sin compresión (16 bits)

Proceso realizado

1. Preparación del audio

- Se convirtió el archivo original a formato WAV PCM firmado 16 bits, necesario para manipular los datos con Python.
- Esto se hizo usando Audacity (Archivo > Exportar > WAV > PCM firmado 16 bits).

2. Codificación del mensaje

- El mensaje "que lindo es el tango" se convirtió a binario (8 bits por carácter).
- Se añadió un delimitador especial (###) para marcar el final del mensaje.

3. Inserción del mensaje

- Se recorrieron los bytes del archivo WAV y se modificó el bit menos significativo de cada byte de audio para insertar el mensaje binario.

- Se utilizó este código en Python:

```python
import wave

def ocultar_mensaje(audio_entrada, mensaje, audio_salida):
    audio = wave.open(audio_entrada, mode='rb')
    frames = bytearray(audio.readframes(audio.getnframes()))
    mensaje += '###'
    bin_mensaje = ''.join(format(ord(c), '08b') for c in mensaje)
    for i in range(len(bin_mensaje)):
        frames[i] = (frames[i] & 254) | int(bin_mensaje[i])
    audio_mod = wave.open(audio_salida, mode='wb')
    audio_mod.setparams(audio.getparams())
    audio_mod.writeframes(frames)
    audio_mod.close()
    audio.close()
```

4. Archivo generado

- El archivo final 10 Mulatada Audio - con mensaje.wav contiene el mensaje oculto de forma imperceptible para el oído humano.

Resultado

El mensaje fue ocultado exitosamente dentro del archivo de audio sin afectar su reproducción. El archivo puede ser compartido sin levantar sospechas, y el mensaje se puede recuperar con un script de extracción similar.

```python
import wave

def extraer_mensaje(audio_modificado):
    # Abrir el archivo con el mensaje oculto
    audio = wave.open(audio_modificado, mode='rb')

    # Leer todos los frames como bytearray
    frames = bytearray(audio.readframes(audio.getnframes()))

    # Extraer el bit menos significativo de cada byte
    bits = [str(frames[i] & 1) for i in range(len(frames))]

    # Agrupar de a 8 bits para formar caracteres
    caracteres = [chr(int(''.join(bits[i:i+8]), 2)) for i in range(0, len(bits), 8)]

    # Unir en un string y cortar donde aparezca el delimitador
    mensaje = ''.join(caracteres)

    audio.close()

    # Retornar solo el texto antes del delimitador
    return mensaje.split('###')[0]

# USO:
ruta_audio = "10 Mulatada Audio - con mensaje.wav"
mensaje_oculto = extraer_mensaje(ruta_audio)
print("Mensaje extraído:", mensaje_oculto)
```