soundsafe.ai

Department of Research & Development

Payload Splitting, Chunking and Encoding

Scenario: Full Payload in a Full-Length Song

Example of some Metadata Fields:

- **ISRC:** `USRM19901234` (12 characters)
- **Title:** `Symphony No. 5 in C Minor, Op. 67: I. Allegro con brio` (50 characters in this example)
- **Performer(s):** `Ludwig van Beethoven, Vienna Philharmonic` (50 characters)
- **Duration:** `00:07:20` (total 440 seconds or 7 minutes 20 seconds)
- **Ownership Information:** Three rights holders with detailed contact info.

Total Payload Estimate (18 Fields): ~3,000 bytes (optimizing for typical cases, not the worst-case edge).

Objective: Every **1-second segment** of this 440-second track must independently carry the **entire 3,000-byte payload**, making it extractable even after manipulation (e.g., lossy compression or trimming).

How the Payload is Embedded

1. Chunk Creation

- **Core Metadata:** Includes ISRC, Title, and Duration.
    - Prioritized for robustness and smaller size (~500 bytes).
- **Auxiliary Metadata:** Includes Ownership Information, License Terms, and Artist Roles (~2,500 bytes compressed).
    - Compressed and embedded redundantly across chunks.

2. Distribution in 1-Second Segments

- **Payload Splitting:**
    - The **entire 3,000 bytes** is serialized, compressed, and split into micro-chunks (~10-20 ms each) to embed redundantly within the second.
- **Time-Based Encoding:**
    - Each 1-second unit carries overlapping micro-chunks, ensuring redundancy. For example:
        - 0.0–0.5 seconds: Chunks A, B, C, D
        - 0.5–1.0 seconds: Chunks C, D, E, F
- **Frequency-Based Encoding:**
    - Chunks are embedded in multiple frequency bands, e.g., low and mid frequencies for critical fields (ISRC, Title) and high frequencies for auxiliary data.

3. Redundancy for Resilience

- The **entire payload** is encoded redundantly within each second:
  - **Core Metadata:** Encoded multiple times in robust frequency bands.
  - **Auxiliary Metadata:** Distributed across frequency and time redundantly with error correction codes (e.g., Reed-Solomon).

Encoding in the Song

Start of the Song (First Second)

- **ISRC (USRM19901234):** Encoded multiple times in low frequencies (robust to distortion).
- **Title (`Symphony No. 5...`):** Encoded redundantly using Huffman compression.
- **Ownership Information:** Chunked, compressed, and distributed in less robust bands with error correction.

Middle of the Song (e.g., 220th Second)

- **New Payload Instance:** Every second has an independent encoding of the **entire payload**, meaning the 220th second carries all metadata, unaffected by the previous or following seconds.

End of the Song (Last Second)

- The same full payload is embedded as at the start, ensuring consistent extractability.

Decoding and Recovery

1. **Extract Payload from a Single Second:**
   - **Synchronization Markers:** Identify micro-chunks and align them for decoding.
   - **Reassemble Payload:** Combine overlapping chunks to reconstruct the full payload.
   - **Error Correction:** Use redundancy and codes like Reed-Solomon to recover corrupted data.
2. **Handle Manipulations:**
   - **Lossy Compression:** Redundancy ensures sufficient data survives to decode the payload.
   - **Trimming:** Any 1-second audio segment retains the full payload, so trimming doesn’t impact extractability.

Technical Summary for a Full-Length Song

Key Points:

- A **7:20 song (440 seconds)** will have **440 independent copies of the full payload**.
- Each second's embedding:
  - Encodes ~3,000 bytes (compressed and chunked) redundantly across time and frequency.
  - Ensures recovery even after manipulation (compression, re-encoding, or trimming).

Benefits:

- **Robustness:** Full metadata is extractable from any single second of audio.
- **Scalability:** The system works seamlessly across tracks of varying lengths.
- **Standards Compliance:** Metadata remains consistent with DDEX/CWR standards.

Ensuring SoundSafe.ai's watermarking system is robust against the extensive list of **real-world audio attacks** requires adaptive strategies at every stage: encoding, embedding, and decoding. Below is an explanation of how the watermark payload can remain detectable under these expanded real-world conditions.

Robust Watermark Design for Expanded Real-World Audio Attacks

---

1. Common Signal Processing Attacks

Attack Types and Strategies:

1. **Dynamic Range Compression (DRC):**
   - **Impact:** Reduces the intensity of the watermark by flattening amplitude variations.
   - **Solution:**
     - **Multi-band Embedding:** Spread the watermark across both high and low dynamic ranges, making it less dependent on specific amplitude levels.
     - **Redundant Encoding:** Use psychoacoustic masking to embed payload redundantly in perceptually important areas.
2. **Clipping:**
   - **Impact:** Distortion caused by amplitude exceeding limits can destroy parts of the watermark.
   - **Solution:**
     - **Error Correction:** Add robust error correction codes (e.g., Reed-Solomon) to compensate for data loss.
     - **Clipping-Resilient Encoding:** Avoid encoding critical data in amplitude peaks prone to clipping.
3. **De-Essing:**
   - **Impact:** Targets high-frequency sibilant sounds where watermarks may reside.
   - **Solution:** Embed redundant watermark data in mid and low frequencies, less affected by de-essing.

4. **Phasing/Flanging and Chorus/Reverb:**
    - **Impact:** Smearing or delaying signals can desynchronize the watermark.
    - **Solution:**
        - **Time-Frequency Localization:** Encode in short, localized time frames, and spread redundantly across time and frequency to survive smearing effects.
5. **Pitch Shifting and Time Stretching:**
    - **Impact:** Alters the frequency and temporal properties of the audio, misaligning watermark elements.
    - **Solution:**
        - **Frequency-Invariant Features:** Use features that remain consistent across pitch and time changes, such as mel-frequency cepstral coefficients (MFCCs).
        - **Robust Synchronization Markers:** Add adaptive markers that adjust dynamically to pitch/time changes.
6. **Equalization (EQ):**
    - **Impact:** Alters frequency balance, potentially masking or attenuating watermarked frequencies.
    - **Solution:**
        - Embed the payload across a wide range of frequencies to ensure detectability under varied EQ curves.
7. **Noise Gates:**
    - **Impact:** Removes low-amplitude segments, potentially erasing the watermark.
    - **Solution:** Encode in both high and low amplitude regions and ensure critical data resides in higher amplitudes.

---

2. Format and Encoding Attacks

Attack Types and Strategies:

1. **Lossy Codecs (AAC, Ogg Vorbis, WMA):**
    - **Impact:** Compression algorithms remove inaudible or less critical parts of the audio, potentially removing watermarks.
    - **Solution:**
        - **Psychoacoustic Embedding:** Ensure the watermark resides in perceptually critical areas that codecs preserve.
2. **Lossless to Lossy Conversion:**
    - **Impact:** Introduces subtle compression artifacts that may degrade the watermark.
    - **Solution:** Use **robust error correction** and **redundant embedding** across multiple bands to withstand artifact introduction.
3. **Bit Depth Conversion and Dithering:**
    - **Impact:** Quantization noise or reduced bit depth may affect precision in watermark recovery.
    - **Solution:**
        - Use **quantization-resilient embedding** methods, ensuring critical data survives rounding errors.
        - Add redundancy to spread the payload across unaffected bits.

---

3. Environmental and Real-World Attacks

Attack Types and Strategies:

1. **Background Noise and Acoustic Transmission:**
    - **Impact:** Overlapping noises mask the watermark; transmission adds distortions.
    - **Solution:**
        - Encode using **noise-robust features** like spectral subbands less affected by ambient noise.
        - Simulate microphone distortions during training for real-world robustness.
2. **Over-the-Air Transmission:**
    - **Impact:** Wireless distortions introduce packet loss and frequency response alterations.
    - **Solution:** Spread redundancy across time and frequency to mitigate random losses.
3. **Acoustic Interference and EMI:**
    - **Impact:** Overlapping or external signals may obscure the watermark.
    - **Solution:** Use **multi-band redundancy** and **time-frequency masking** to ensure critical parts of the watermark remain detectable.

---

4. Malicious Attacks

Attack Types and Strategies:

1. **Watermark Removal Techniques and Signal Inversion:**
    - **Impact:** Sophisticated attempts to bypass or cancel the watermark.
    - **Solution:**
        - Use cryptographic signatures within the payload to ensure integrity detection.
        - Design watermarks that adapt dynamically to signal inversion.
2. **Copy and Paste/Collusion Attacks:**
    - **Impact:** Replaces segments or combines watermarked copies to remove the watermark.
    - **Solution:** Encode watermarks with **content-specific hashes** to ensure they're tied to the original host audio.
3. **Stretching and Cutting:**
    - **Impact:** Removes parts of the audio, affecting complete payload recovery.
    - **Solution:** Ensure the **entire payload is independently embedded** in every second of audio for guaranteed recovery.

---

5. Combined Attacks

Attack Types and Strategies:

1. **Sequential Attacks:**
   - **Impact:** Compounding effects of multiple distortions.
   - **Solution:** Train models with **randomized attack sequences** to simulate real-world conditions.
2. **Randomized Attack Chains:**
   - **Impact:** Simultaneous or sequential attacks with varied intensity levels.
   - **Solution:** Use **dynamic loss functions** during training to penalize decoding errors under such conditions.

Implementation Workflow

1. **Attack Simulation:**
   - Implement an **expanded attack simulator** to include these real-world conditions.
   - Randomize attack order and intensity during training.
2. **Model Training:**
   - Train the watermarking model on these simulated conditions, optimizing for:
     - **Bit Error Rate (BER):** Reduce errors under extreme distortions.
     - **Robust Synchronization:** Ensure watermark alignment remains intact.
3. **Testing and Validation:**
   - Validate watermark robustness on real-world devices, codecs, and environments.
   - Conduct **sequential and combined attack testing** to identify weaknesses.

Summary:

The SoundSafe.ai watermarking system must incorporate **redundant embedding**, **error correction**, and **attack simulation during training** to ensure payload detectability under all listed real-world manipulations. By spreading the payload redundantly across time and frequency domains, embedding with psychoacoustic precision, and simulating attack chains during development, the watermark can achieve resilience even in hostile audio conditions.