Bài Thực Hành: Giấu Tin Trong Âm Thanh Bằng Phương Pháp Phase Coding

1. Mục tiêu

- Hiểu về kỹ thuật giấu tin (steganography) trong âm thanh sử dụng **Phase Coding**.
- Tìm hiểu cách nhúng thông điệp bí mật vào file âm thanh WAV mà không làm thay đổi rõ rệt chất lượng âm thanh.
- Thực hành giấu tin vào âm thanh bằng mã Python.
- Thực hành trích xuất thông điệp ẩn từ file âm thanh đã xử lý.

2. Yêu cầu thực hành

Phase Coding là một phương pháp giấu tin trong âm thanh, tận dụng việc thay đổi pha của các thành phần tần số mà tai người ít nhạy cảm. Kỹ thuật này sử dụng biến đổi Fourier nhanh (FFT) để nhúng thông điệp vào phổ pha của tín hiệu âm thanh, sau đó tái tạo lại âm thanh với thông điệp ẩn.

Kịch bản

- Kẻ tấn công có thể giấu thông tin bí mật (ví dụ: mã độc hoặc tin nhắn) vào file âm thanh và phát tán mà không bị phát hiện.
- Nhà nghiên cứu bảo mật cần nắm kỹ thuật này để phát hiện và trích xuất dữ liệu ẩn từ âm thanh.

Yêu cầu

- Hiểu khái niệm cơ bản về **Steganography**.
- Có kiến thức cơ bản về xử lý tín hiệu âm thanh và biến đổi Fourier.
- **Công cụ sử dụng**:
    - Python 3.x.
    - Thư viện: wave, numpy, scipy.
    - File âm thanh WAV (không nén).

3. Thực hành

Khởi động bài lab: Giả sử môi trường thực hành đã được thiết lập với Python và các thư viện cần thiết.

labtainer –r stego_code_audio_phase

Bước 1: Chuẩn bị file âm thanh và thông điệp

- **File âm thanh**: Sử dụng một file WAV không nén (ví dụ: sample.wav).
- **Thông điệp bí mật**: Tạo file secret.txt chứa thông điệp (ví dụ: "root").

nano secrect.txt

sau đó nhập root và lưu

Bước 2: Giấu tin vào âm thanh bằng Phase Coding

Sử dụng lệnh sau để nhúng thông điệp vào file âm thanh:

python3 hide.py

Giải thích:

1. **Đọc thông điệp**: Đọc nội dung từ file secret.txt (trong trường hợp này là "root")
2. **Chuyển đổi thông điệp**: Chuyển text thành chuỗi bit nhị phân
3. **Xử lý âm thanh**:
   - Chia file âm thanh thành các đoạn nhỏ (mỗi đoạn 1024 mẫu)
   - Mỗi bit của thông điệp được giấu vào một đoạn âm thanh
4. **Giấu thông điệp**:
   - Sử dụng FFT để chuyển đổi mỗi đoạn âm thanh sang miền tần số
   - Thay đổi pha của tín hiệu để biểu diễn bit 0 hoặc 1
   - Chuyển đổi ngược lại sang miền thời gian
5. **Lưu kết quả**: Lưu file âm thanh đã giấu thông điệp thành stego_audio.wav

Bước 3: Trích xuất thông điệp từ âm thanh

Sử dụng lệnh sau để đọc thông điệp từ file âm thanh chưa thông điệp giấu:

python3 extract.py

Giải thích:

1. **Đọc file âm thanh**: Đọc file stego_audio.wav đã giấu thông điệp
2. **Trích xuất bit**:
   - Chia âm thanh thành các đoạn giống như khi nhúng
   - Với mỗi đoạn:
   - Chuyển đổi sang miền tần số bằng FFT

- Đọc giá trị pha để xác định bit 0 hay 1

3. **Chuyển đổi kết quả**:
   - Ghép các bit lại thành chuỗi nhị phân
   - Chuyển đổi chuỗi nhị phân thành text
4. **Hiển thị kết quả**: In ra thông điệp đã trích xuất được

Bước 4: So sánh file video gốc và file đã giấu tin

4.1. So sánh kích thước file bằng lệnh ls -l

- **Mục đích**: Kiểm tra xem quá trình giấu tin có làm thay đổi kích thước của file âm thanh có thay đổi hay không.
- **Lệnh**:

ls -l sample.mav stego_audio.wav

- **Giải thích**:
  - Lệnh này hiển thị thông tin chi tiết về hai file, bao gồm kích thước (tính bằng byte).
  - Nếu kích thước của hai file giống nhau, điều này cho thấy quá trình giấu tin không làm thay đổi độ dài tổng thể của file âm thanh.

4.2. So sánh nội dung file bằng lệnh cmp

- **Mục đích**: Kiểm tra sự khác biệt về nội dung giữa file gốc và file đã giấu tin.
- **Lệnh**:

cmp sample.mav stego_audio.wav

- **Giải thích**:
  - Nếu hai file hoàn toàn giống nhau, lệnh cmp sẽ không hiển thị gì.
  - Nếu có sự khác biệt, lệnh sẽ báo vị trí byte đầu tiên mà hai file khác nhau. Điều này xác nhận rằng nội dung file đã thay đổi do quá trình giấu tin.

4. Kết quả cần đạt được

- Chạy thành công các bước: giấu tin và trích xuất thông điệp.

- **Kết thúc bài lab**:
    - Dùng lệnh: stoplab stego_code_audio_phase.
    - File kết quả sẽ được tạo và lưu tại vị trí hiển thị sau lệnh stoplab.
- **Kiểm tra kết quả**: checkwork stego_code_audio_phase
- **Khởi động lại lab nếu cần**: labtainer –r stego_code_audio_phase