HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

HỌC PHẦN: CÁC KỸ THUẬT GIẤU TIN
MÃ HỌC PHẦN: INT14102

Chủ đề: Giấu tin trong âm thanh

Lab: fhss_watermark

Sinh viên thực hiện:	Nguyễn Trường Sơn
Mã sinh viên:	B21DCAT164
Nhóm:	04
Giảng viên hướng dẫn:	Đỗ Xuân Chợ

HÀ NỘI 2025

Bài lab Các kỹ thuật giấu tin: fhss_watermark

1. Mục đích

Giúp sinh viên hiểu được ứng dụng của thủy vân số (watermark) trong thuật toán trải phổ nhảy tần FHSS.

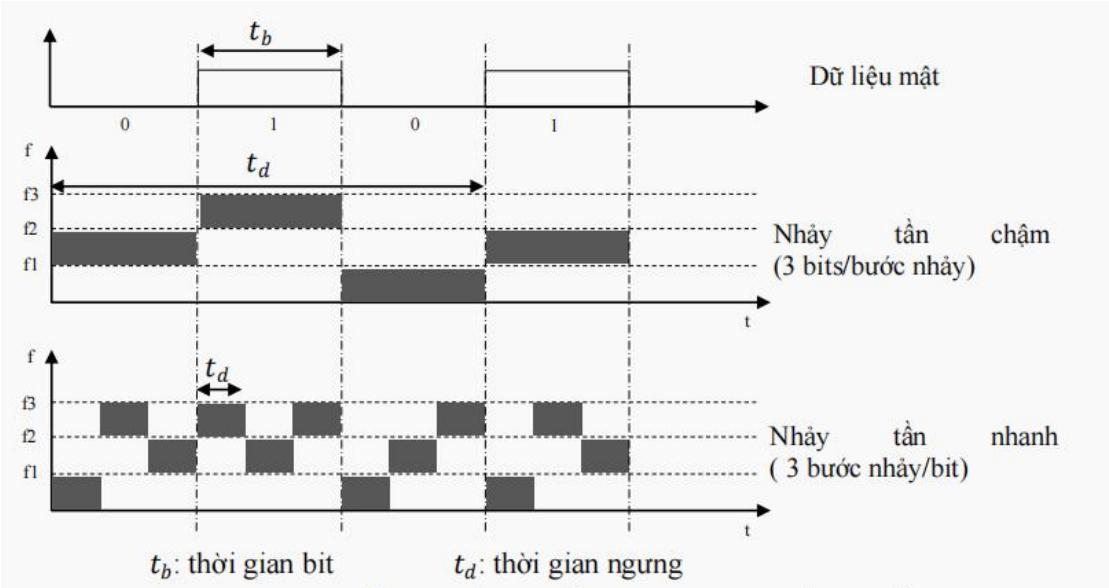
2. Yêu cầu đối với sinh viên

Quen thuộc với hệ điều hành Linux và có kiến thức về kỹ thuật giấu tin.

3. Nội dung lý thuyết

3.1. Nhắc lại lý thuyết về trải phổ nhảy tần

Trải phổ nhảy tần (FHSS) là công nghệ thay đổi tần số truyền đột ngột trong một dãy băng tần. Băng thông được chia thành nhiều khe tần không lấn nhau, và tín hiệu truyền đi có thể chiếm một hoặc nhiều khe tần, việc chọn khe tần được thực hiện theo chuỗi giả ngẫu nhiên. Dựa vào tốc độ nhảy tần, có hai loại: nhảy tần nhanh (tốc độ nhảy nhanh hơn tốc độ dữ liệu) và nhảy tần chậm (tốc độ nhảy chậm hơn tốc độ dữ liệu).



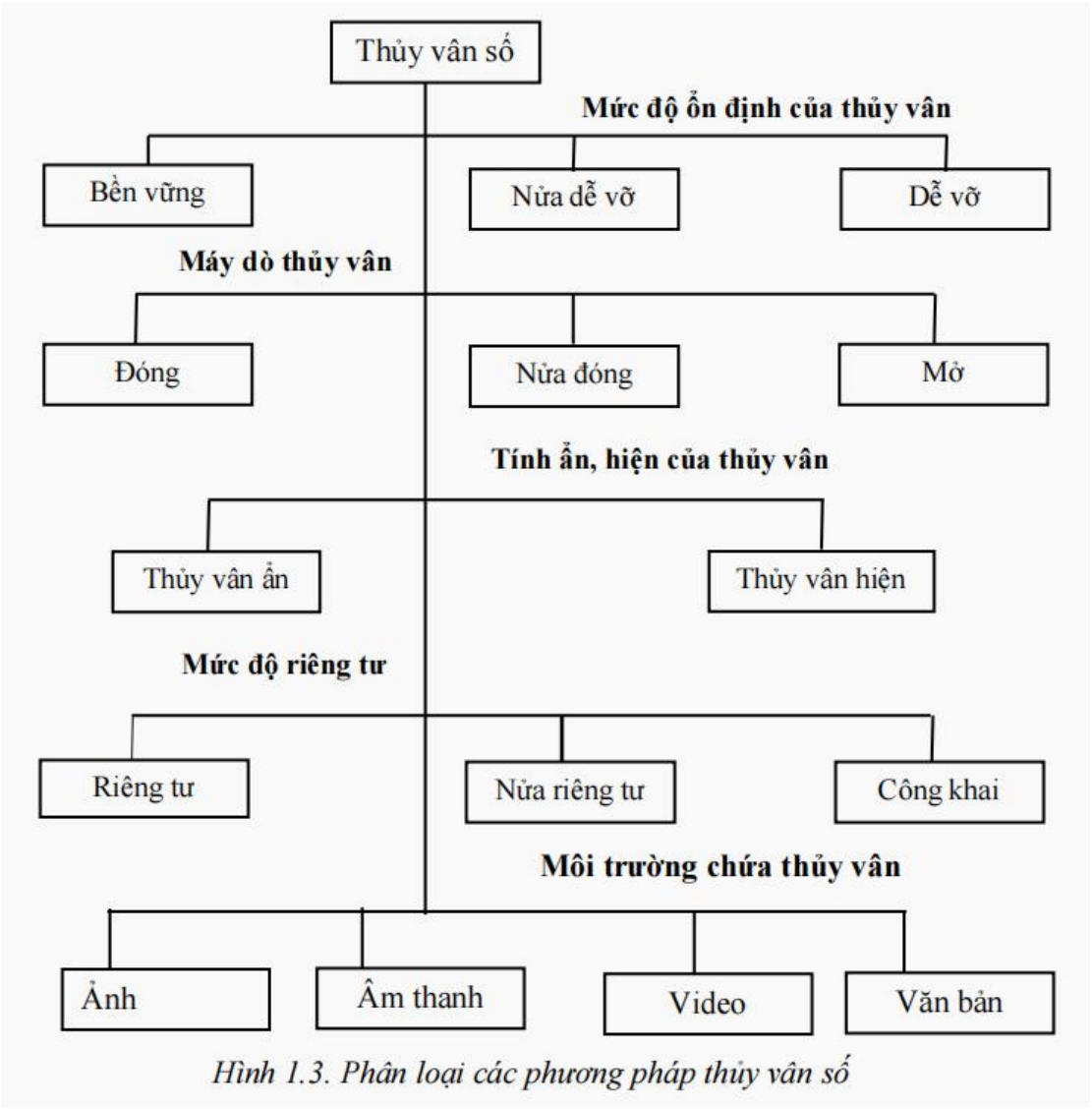
Hình 3.17. Minh hoạ về trải phổ nhảy tần nhanh và trải phổ nhảy tần chậm

3.2. Khái quát về thủy vân số (watermark)

3.2.1 Khái niệm

Thủy vân số là phương pháp giấu thông tin (thủy vân) vào các vật chứa, mang ý nghĩa bảo vệ tính toàn vẹn của vật chứa. Thủy vân là một thông tin nào đó mang ý nghĩa. Yêu cầu đối với thủy vân là một lượng thông tin rất nhỏ nhưng đủ mạnh để có thể bảo vệ vật chứa thủy vân. Ứng dụng của thủy vân số hiện nay rất đa dạng và hầu hết các lĩnh vực như: bảo vệ bản quyền hoặc chống xuyên tạc nội dung,…

3.2.2. Phân loại thủy vân số



Hình 1.3. Phân loại các phương pháp thủy vân số

Hình trên trình bày 4 hướng tiếp cận chính được phân loại theo thủy vân số tương ứng với 4 môi trường: Ảnh, âm thanh, video và văn bản. Bài lab này thực hiện trên môi trường âm thanh với hướng tiếp cận là mức độ ổn định của thủy vân, được mô tả như sau:

- Thủy vân số bền vững (Robust Watermarking): Là dạng thủy vân tồn tại bền vững trước các cuộc tấn công nhằm loại bỏ thủy vân. Trong trường hợp loại bỏ được thủy vân thì vật chứa tin cũng không còn giá trị sử dụng. Một ứng dụng điển hình của thủy vân bền vững chính là bảo vệ bản quyền: thuỷ vân được nhúng trong sản phẩm như một hình thức dán tem bản quyền.
- Thủy vân số nửa dễ vỡ (Semi Fragile Watermarking): Là dạng thủy vân tồn tại bền vững khi vật chứa tin bị sửa đổi vô hại như: nén, làm nhiễu, lọc,… nhưng lại nhạy cảm (dễ vỡ) khi vật chứa tin bị sửa đổi độc hại như: đổi nội

dung, cắt bỏ một phần. Thủy vân nửa dễ vỡ được thiết kế để phát hiện các sửa đổi độc hại trên sản phẩm (nhằm đảm bảo tính toàn vẹn của sản phẩm), đồng thời cho phép một số hoạt động sửa đổi vô hại trên sản phẩm.

- Thủy vân số dễ vỡ (Fragile Watermarking): Là dạng thủy vân nhạy cảm (dễ vỡ) trước mọi thay đổi của vật chứa tin, dù là thay đổi nhỏ nhất. Chính vì đặc điểm nhạy cảm như vậy nên thủy vân dễ vỡ được ứng dụng nhiều vào việc xác thực nội dung. Ví dụ: Khi một tòa soạn sử dụng một bức ảnh để đưa tin, toà soạn phải xác minh bức ảnh này đúng với ảnh gốc và chưa được chỉnh sửa.

3.3 Ứng dụng trong bài lab

Mục đích của bài lab là mô phỏng lại hướng tiếp cận theo mức độ ổn định của thủy vân. Theo đó, sinh viên sẽ được cung cấp file âm thanh gốc và các file code python phục vụ quá trình nhúng, trích xuất thủy vân. Cùng với đó, sinh viên sẽ thực hiện thêm 3 nhiệm vụ: trích xuất thủy vân từ các file âm thanh sau quá trình nhúng, các file này đã được nén, làm nhiễu và cắt. Theo lý thuyết, nếu là thủy vân nửa dễ vỡ thì quá trình nén và làm nhiễu sẽ không làm ảnh hưởng đến thủy vân được nhúng, còn nếu file âm thanh bị cắt đi thì sẽ không thể khôi phục thủy vân ban đầu.

- **Đầu vào:**
  - File âm thanh gốc, tệp tin chứa thủy vân.
  - 1 file code dùng để nhúng, 1 file code dùng để trích xuất thủy vân, 1 file code dùng để thêm nhiễu vào file âm thanh sau khi đã nhúng thủy vân.
  - 2 file âm thanh đã nhúng thủy vân sau khi nén và sau khi cắt.
- **Tổng quan các nhiệm vụ:**
  - Nhúng thủy vân vào 1 file âm thanh.
  - Giải mã thủy vân từ file âm thanh đó.
  - Giải mã thủy vân từ file âm thanh nhúng đã được nén/cắt/làm nhiễu, từ đó rút ra kết luận về mức độ ổn định của thủy vân.

4. Nội dung thực hành

Khởi động bài lab:

Vào terminal, gõ:

rebuild fhss

(Chú ý: Sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)

Quy trình nhúng và trích xuất thủy vân được mô tả ngắn gọn trong ảnh sau

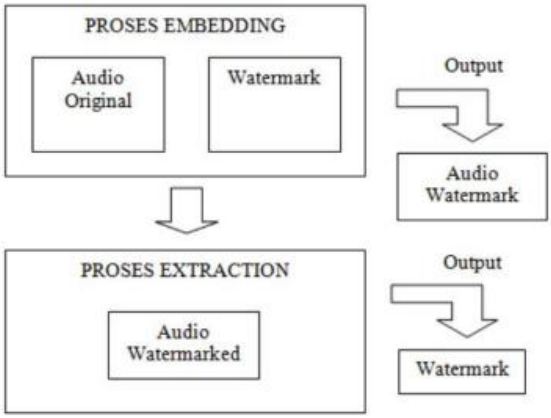


Figure 1. General System (Stefan Katzenbeisser and Fabien A2000)

Quá trình nhúng: Từ file âm thanh gốc, thủy vân được nhúng vào tạo ra file âm thanh mới.

Quá trình trích xuất: Từ file âm thanh sau khi nhúng, giải mã để thu được thủy vân.

Chi tiết cho quá trình này sẽ được giải thích đầy đủ trong các nhiệm vụ dưới.

Nhiệm vụ 1: Nhúng thủy vân

Quy trình chi tiết như sau

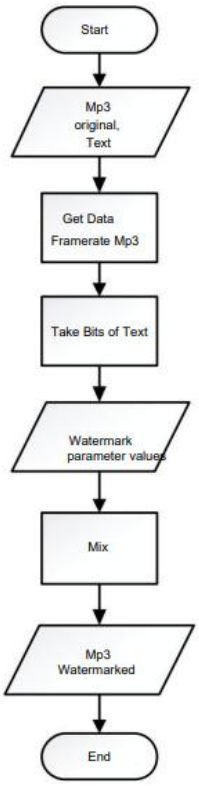


Figure 2. Embedding Process Flowchart

Các bước chính:

- Bước 1: Lấy đầu vào là file âm thanh (trong bài lab sử dụng file .wav) và 1 file .txt chứa thủy vân (trong bài lab, giá trị của thủy vân là "Cảnh sát chính tả").
- Bước 2: Lấy dữ liệu frame từ file âm thanh gốc.
- Bước 3: Chuyển đổi thủy vân thành dạng nhị phân và phân phối thành tín hiệu âm thanh.
- Bước 4: Sử dụng 1 giá trị alpha làm giới hạn độ cao của biên độ.
- Bước 5:
  - Lấy dữ liệu từ thủy vân (1) và dữ liệu frame từ file âm thanh gốc (2), truyền các bit giá trị của (1) vào (2) bằng cách sử dụng chuỗi giả ngẫu nhiên PN. Sơ đồ cho quá trình này như ảnh dưới

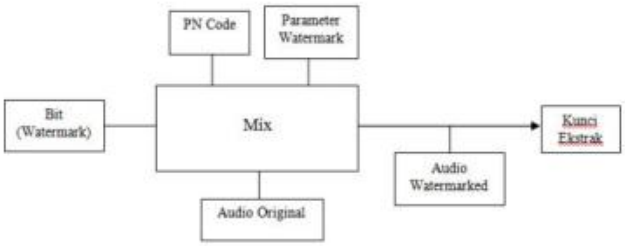


Figure 3. Mix Process Block (Gordy 2000).

From the explanation above, the following calculations are explained:

*Original Audio Coefficient Data + (Parameters Watermark * Bit (Watermark) * PN Code) = Watermarked Audio Coefficient Data*

  - Sau đó, áp dụng công thức như trong ảnh cùng biến đổi Fourier để tính ra dữ liệu của âm thanh chứa thủy vân.

Toàn bộ quá trình này đã được mô phỏng lại bằng file *embed_watermark.py*, sinh viên chạy lệnh sau để thực hiện nhúng thủy vân:

python3 embed_watermark.py sample_29s.wav

Kết quả thu được file âm thanh chứa thủy vân.

Nhiệm vụ 2: Trích xuất thủy vân

Quy trình chi tiết như sau

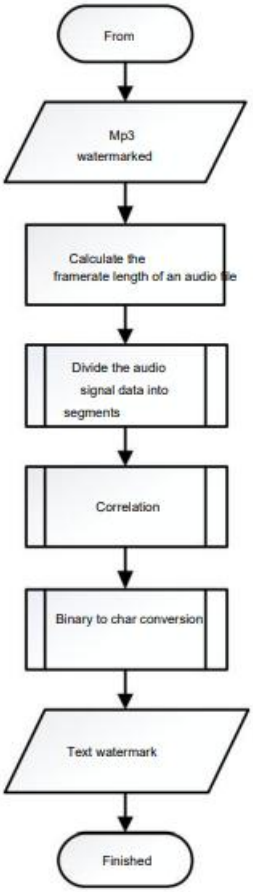


Figure 5. Extracting Process Flowchart

- Bước 1: Lấy đầu vào là file âm thanh sau khi đã nhúng thủy vân.
- Bước 2: Tính tổng dữ liệu frame đã được đưa vào trong quá trình nhúng.
- Bước 3: Chia dữ liệu âm thanh thành các segment, sau đó áp dụng biến đổi Fourier để thu được hệ số của thủy vân (Watermark Coefficent Data).
- Bước 4: Áp dụng công thức cho quá trình tương quan (Correlation Process)

The formula for the Correlation Process is

Audio Watermark Coefficient Data – Data Original Coefficient = Watermark Data

  - Từ dữ liệu hệ số trong file âm thanh chứa thủy vân, ta sẽ trừ đi hệ số trong file âm thanh gốc, từ đó thu được dữ liệu thủy vân dạng bit.
- Bước 5: Chuyển đổi giá trị bit sang ký tự, thu được thủy vân ban đầu.

Toàn bộ quá trình này đã được mô phỏng trong file *extract_watermark.py*, sinh viên chạy lệnh sau:

python3 extract_watermark.py sample_29s.wav sample_output.wav

Kết quả thu được thủy vân ban đầu.

Nhiệm vụ 3: Trích xuất thủy vân từ file chứa thủy vân đã được nén

Bây giờ chúng ta sẽ kiểm tra độ bền vững của thủy vân với điều kiện file âm thanh chứa nó bị nén. Ở đây bài lab đã cung cấp file *sample_output_compress.wav*, trước tiên chúng ta sẽ kiểm tra file này và so sánh nó với file âm thanh chứa thủy vân không bị nén:

soxi sample_output.wav

soxi sample_output_compress.wav

Kết quả cho thấy file size và bit rate của file nén chỉ bằng 1 nửa file ban đầu. Sinh viên chạy lệnh sau:

python3 extract_watermark.py sample_29s.wav sample_output_compress.wav

Kết quả cho thấy watermark vẫn extract được.

Nhiệm vụ 4: Trích xuất thủy vân từ file chứa thủy vân đã làm nhiễu

Bây giờ chúng ta sẽ kiểm tra độ bền vững của thủy vân với điều kiện file âm thanh chứa nó bị làm nhiễu. Bài lab đã cung cấp file *add_noise.py* dùng để thêm nhiễu với giá trị SNR là 50dB vào file âm thanh đã nhúng thủy vân. Sinh viên chạy lệnh sau

python3 add_noise.py sample_output.wav

Kết quả cho thấy 1 file sample_output_noisy.wav đã được tạo ra, bây giờ thử so sánh với file đã nhúng ban đầu

soxi sample_output.wav

soxi sample_output_noisy.wav

Kết quả cho thấy 2 file có vẻ giống nhau, tuy nhiên, để chắc chắn thì cần so sánh raw data của 2 file âm thanh. Sử dụng lệnh sau

diff <(sox sample_output.wav -t .raw -) <(sox sample_output_noisy.wav -t .raw -)

Giải thích:

- sox: Đây là lệnh gọi công cụ sox để xử lý tệp âm thanh, ở đây công cụ đang nhận dữ liệu đầu vào là các file wav.
  - -t .raw: Tùy chọn này chỉ định loại tệp đầu ra, là tệp nhị phân thô (RAW), không có header.
  - -: Dấu gạch ngang ở cuối chỉ ra rằng tệp đầu ra sẽ được gửi đến đầu ra chuẩn (stdout), thay vì lưu vào tệp.
- diff <(...) <(...):
  - Lệnh diff được dùng để so sánh nội dung của hai tệp.

- diff so sánh kết quả đầu ra của hai lệnh con (sox chuyển đổi tệp WAV thành tệp thô trong trường hợp này).
- Dấu <(...) là cú pháp để thực thi một lệnh trong bash và so sánh kết quả đầu ra của chúng mà không cần lưu vào tệp.

Nếu kết quả của lệnh trên có dạng "Binary files /dev/fd/63 and /dev/fd/62 differ", điều đó cho thấy rằng 2 tệp wav có raw data khác nhau (dạng nhị phân).

Tiếp theo chúng ta thử xác định xem raw data của 2 file wav này khác nhau bắt đầu từ byte nào. Sử dụng lệnh

cmp <(sox sample_output.wav -t .raw -) <(sox sample_output_noisy.wav -t .raw -)

Kết quả cho thấy ký tự thứ 2 của dòng 1 là nơi đầu tiên xảy ra sự khác biệt. Như vậy, nội dung 2 file khác nhau hoàn toàn. Bây giờ hãy thử trích xuất thủy vân từ file *sample_output_noisy.wav* này.

python3 extract_watermark.py sample_29s.wav sample_output_noisy.wav

Kết quả cho thấy trích xuất thủy vân thành công.

Nhiệm vụ 5: Trích xuất thủy vân từ file chứa thủy vân đã bị cắt 1 phần

Tương tự 2 nhiệm vụ trên, chúng ta sẽ kiểm tra độ bền vững của thủy vân với điều kiện file chứa nó bị cắt bỏ 1 phần. Trong bài lab đã cung cấp file sample_output_cut.wav, trước hết ta so sánh nó với file sample_output ban đầu:

soxi sample_output.wav

soxi sample_output_cut.wav

Ta thấy rằng duration chỉ còn 1/3, bây giờ chạy lệnh sau để thử trích xuất thủy vân từ file đã bị cắt:

python3 extract_watermark.py sample_29s.wav sample_output_cut.wav

Kết quả thu được là 1 chuỗi ký tự vô nghĩa.

Như vậy, từ kết quả của nhiệm vụ 3, 4, 5, ta có thể kết luận thủy vân số trong bài lab thuộc loại nửa dễ vỡ.

Kết thúc bài lab:

Kiểm tra checkwork:

checkwork

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

stoplab