**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_decode

| | |
|---|---|
| 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_decode

## 1. Mục đích

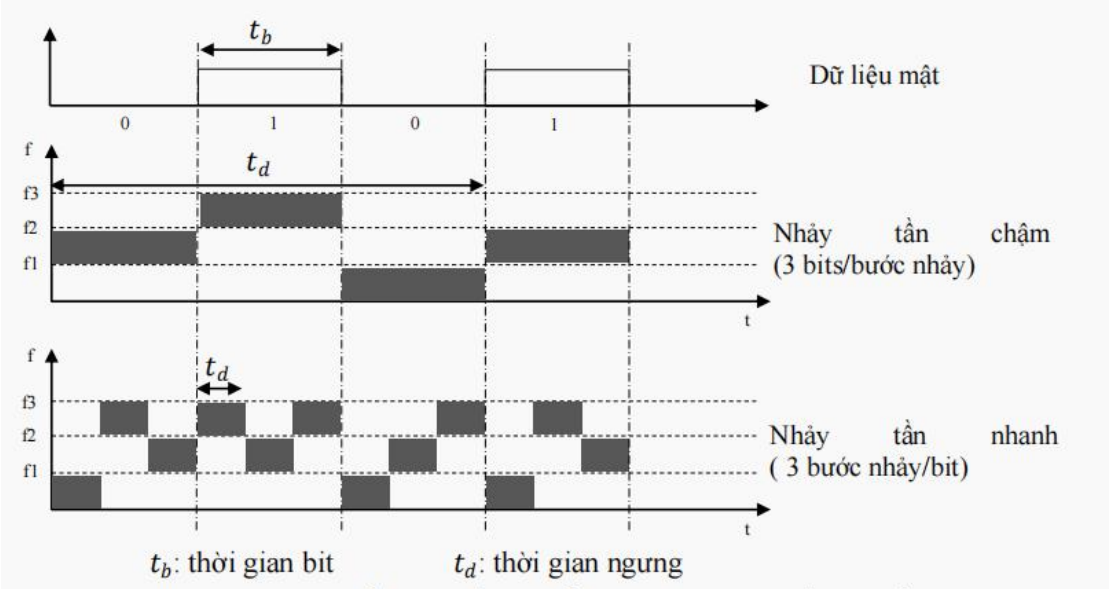
Giúp sinh viên hiểu được thuật toán giấu tin trong âm thanh sử dụng phương pháp trải phổ nhảy tần FHSS, cụ thể là quy trình tách tin của phương pháp này.

## 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

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). Bài thực hành này thực hiện theo phương pháp nhảy tần chậm.



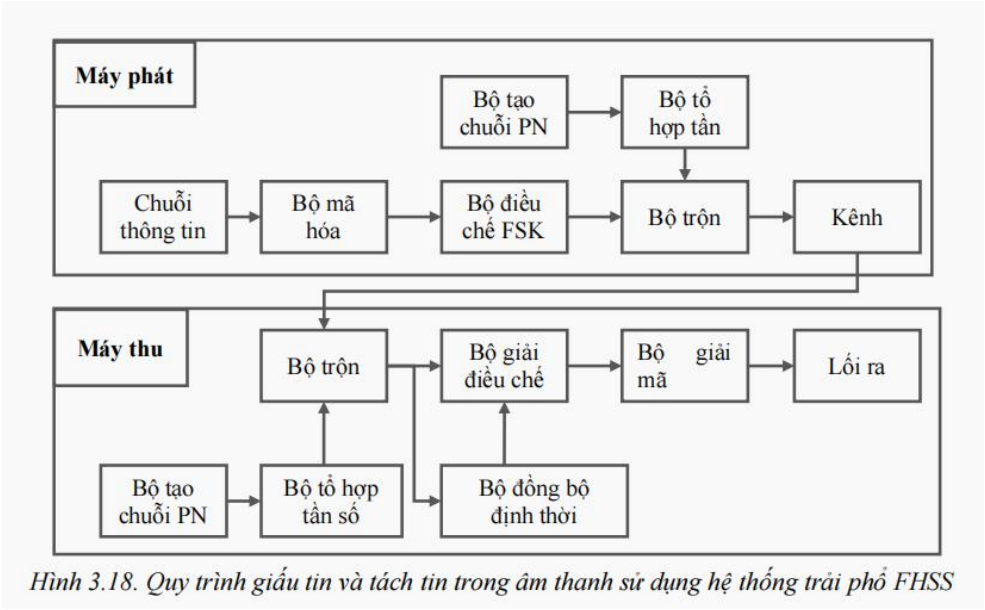

*Hình 3.17. Minh hoạ về trải phổ nhảy tần nhanh và trải phổ nhảy tần chậm*

Quy trình tách tin:

Hình dưới mô tả quá trình giấu tin ở phía máy phát và tách tin ở phía máy thu. Quá trình tách tin bao gồm các bước sau:

- Thu tín hiệu từ kênh truyền => đưa vào bộ trộn.
- Bộ tạo chuỗi ngẫu nhiên tạo chuỗi giả ngẫu nhiên (PN) đồng bộ với chuỗi tới.
- Chuỗi giả ngẫu nhiên được gửi đến bộ tổ hợp tần số để tạo giá trị nhảy tần.

- Bộ tổ hợp tần số điều khiển tần số sóng mang và gửi tín hiệu ra bộ trộn.
- Bộ trộn kết hợp tín hiệu thu với tín hiệu từ bộ tổ hợp tần số, qua bộ lọc BPF tạo x(t).
- Tín hiệu x(t) được gửi đến bộ giải điều chế FSK và bộ đồng bộ định thời để đồng bộ về thời gian.
- Bộ giải điều chế FSK giải điều chế tín hiệu để tái tạo dữ liệu.
- Bộ giải mã khôi phục dữ liệu gốc ban đầu.




*Hình 3.18. Quy trình giấu tin và tách tin trong âm thanh sử dụng hệ thống trải phổ FHSS*

Trong bài lab này, thuật toán được mô tả như sau:

- **Đầu vào:**
  - File âm thanh gốc và file âm thanh đã qua quá trình giấu tin, mục đích của 2 file là để tách lấy tín hiệu FHSS đã được gửi đi.
  - Chuỗi đồng bộ PN đã sử dụng ở phía máy phát.
  - Tập hợp các tần số sóng mang được dùng trong quá trình giấu tin.
- **Các bước thực hiện tách tin**
  - Bước 1: Tiền xử lý dữ liệu bằng cách chuyển các file âm thanh thành định dạng mono để dễ cho quá trình xử lý. Sau đó lưu ra các file làm đầu vào cho quá trình tách tin.
  - Bước 2: Đọc dữ liệu tiền xử lý thu được, tách lấy tín hiệu FHSS bằng cách lấy hiệu của âm thanh đã giấu tin và âm thanh ban đầu.
  - Bước 3: Thực hiện giải điều chế tín hiệu, đồng thời sử dụng chuỗi đồng bộ hóa đã dùng ở quá trình giấu tin làm để tìm vị trí bắt đầu của thông điệp.

- Bước 4: Chuyển các bit thông điệp về dạng plaintext, thu được thông điệp ban đầu.

- **Đầu ra:** Thông điệp ban đầu.

**4. Nội dung thực hành**

Khởi động bài lab:

git clone https://github.com/SonNTB21DCAT164/fhss_decode

Sau đó giải nén file *fhss_decode.tar.xz* rồi chuyển thư mục đã giải nén vào trunk/labs

Vào terminal, gõ:

*rebuild fhss_decode*

(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)

**Nhiệm vụ 1: So sánh file âm thanh gốc với file âm thanh đã giấu tin bằng FHSS**

Mục đích của nhiệm vụ này là tìm hiểu sự khác biệt giữa 2 file wav trước và sau giấu tin. Trước tiên, sử dụng công cụ soxi để xem thông tin kỹ thuật của 2 file âm thanh

*soxi sample_29s.wav*

*soxi fhss_output_29s.wav*

Kết quả cho thấy 2 file có cùng thông số kỹ thuật. Tuy nhiên, điều đó không có nghĩa là nội dung âm thanh giống nhau, do đó cần so sánh raw data của 2 file âm thanh. Sử dụng lệnh sau

***diff <(sox sample_29s.wav -t .raw -) <(sox fhss_output_29s.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_29s.wav -t .raw -) <(sox fhss_output_29s.wav -t .raw -)*

Kết quả cho thấy ký tự thứ 5 của dòng 1 là nơi đầu tiên xảy ra sự khác biệt. Như vậy, mặc dù 2 file có cùng định dạng và thông số kỹ thuật nhưng nội dung 2 file khác nhau hoàn toàn.

**Nhiệm vụ 2: Tiền xử lý**

Tiền xử lý dữ liệu bằng cách chuyển các file âm thanh thành định dạng mono để dễ cho quá trình xử lý. Sau đó lưu ra các file làm đầu vào cho quá trình tách tin. Thực hiện câu lệnh sau

***python3 tienxuly.py***

Sử dụng lệnh *ls* để liệt kê các file mới xuất hiện, đây là các file dùng làm đầu vào/xử lý cho quá trình tách tin.

**Nhiệm vụ 3: Giấu tin**

- Nhiệm vụ này thực hiện các bước
  - Bước 1: Đọc dữ liệu tiền xử lý thu được, tách lấy tín hiệu FHSS bằng cách lấy hiệu của âm thanh đã giấu tin và âm thanh ban đầu.
  - Bước 2: Thực hiện giải điều chế tín hiệu, đồng thời sử dụng chuỗi đồng bộ hóa đã dùng ở quá trình giấu tin làm để tìm vị trí bắt đầu của thông điệp.
  - Bước 3: Chuyển các bit thông điệp về dạng plaintext, thu được thông điệp ban đầu.

Toàn bộ quá trình này được thực hiện trong file *tachtin.py,* sinh viên thực hiện câu lệnh sau:

***python3 tachtin.py***

Kết quả cho thấy thông điệp ‘hidden message’ đã được giải mã thành công.

**Kết thúc bài lab:**

Kiểm tra checkwork:

*checkwork*

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

*stoplab*