**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: CÁC KỸ THUẬT GIẤU TIN**

**MÃ HỌC PHẦN: INT14102**

**CHỦ ĐỀ: THỰC HIỆN GIẤU TIN TRONG ÂM THANH BẰNG PHƯƠNG PHÁP FHSS QUA CÔNG CỤ GNU-RADIO**

**Lab: giau\_tin\_fhss\_gnuradio**

| Sinh viên thực hiện | Đặng Minh Hiếu |
| --- | --- |
| Mã sinh viên | B21DCAT087 |

Tên nhóm: Nhóm 10

Tên lớp: Nhóm 3

Giảng viên hướng dẫn: TS. Đỗ Xuân Chợ

**HÀ NỘI 2025**

# Bài lab : Thực hiện giấu tin trong âm thanh bằng phương pháp FHSS qua công cụ GNU-Radio

## Mục đích

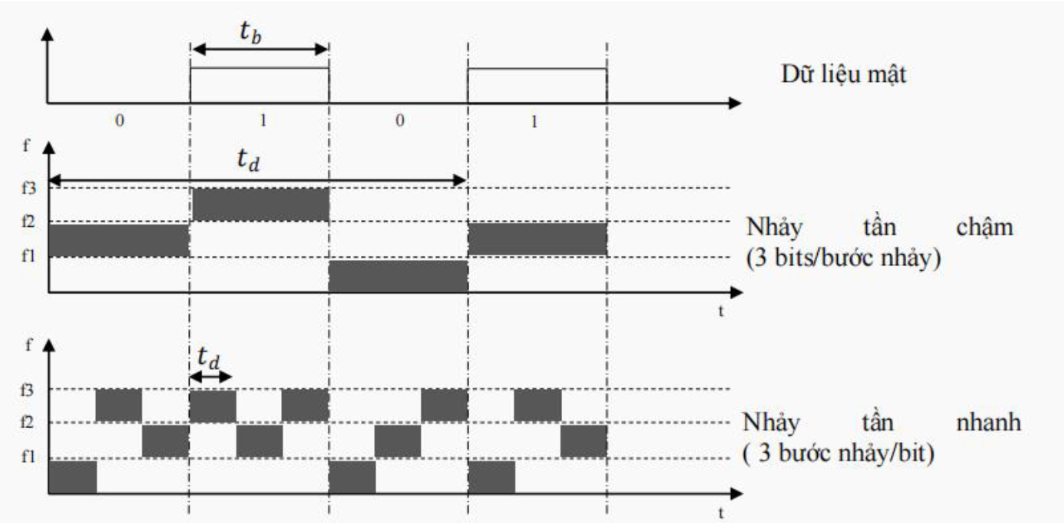
* Giúp sinh viên hiểu được thuật toán giấu tin và tách tin trong âm thanh sử dụng phương pháp FHSS
* Sinh viên làm quen với công cụ GNU-Radio và các block sử dụng trong công cụ

## Yêu cầu đối với sinh viên

* Nắm vững các khái niệm cơ bản về giấu tin (Steganography), đặc biệt là giấu tin trong môi trường âm thanh.
* Hiểu sâu về nguyên lý hoạt động của phương pháp Trải phổ Nhảy tần (FHSS) trong truyền thông và cách nó được ứng dụng để giấu tin trong âm thanh. Điều này bao gồm các thành phần cốt lõi như: chuỗi giả ngẫu nhiên (PN sequence), cơ chế nhảy tần, điều chế FSK (Frequency Shift Keying) cho dữ liệu mật, vai trò của khóa bí mật.
* Hiểu biết về các đặc trưng cơ bản của tín hiệu âm thanh số (tần số lấy mẫu, lượng tử hóa, định dạng file WAV).
* Có kiến thức về cách biểu diễn và xử lý dữ liệu âm thanh trong Python (ví dụ: sử dụng thư viện như scipy.io.wavfile, numpy).
* Thành thạo ngôn ngữ lập trình Python.
* Hiểu rõ kiến trúc và cách hoạt động của GNU Radio và GNU Radio Companion (GRC).
* Nắm được chức năng của các khối xử lý tín hiệu (blocks) cơ bản thường dùng trong GNU Radio (ví dụ: sources, sinks, filters, modulators, math operators, PN sequence generators).

## Nội dung lý thuyết

Trải phổ nhảy tần là một công nghệ sử dụng bộ phát tần số và có thể thay đổi tần số truyền một cách đột ngột trong dãy băng tần sử dụng. Trong trải phổ nhảy tần, độ rộng băng thông sẵn có sẽ được chia thành một số lớn các khe tần không lấn lên nhau. Tại bất kì khoảng thời gian nào, tín hiệu truyền đi đều chiếm một hay nhiều hơn một khe tần số nói trên. Việc chọn một khe hay nhiều khe tần số trong một khoảng thời gian truyền tín hiệu đều được thực hiện một cách giả ngẫu nhiên theo tín hiệu ra của một bộ tạo chuỗi giả ngẫu nhiên. Dựa trên tốc độ nhảy của tần số thì phương pháp trải phổ nhảy tần được chia làm 2 loại đó là trải phổ nhảy tần nhanh và trải phổ nhảy tầm chậm. Trong đó, trải phổ nhảy tần nhanh có đặc điểm là tốc độ nhảy nhanh hơn tốc độ dữ liệu. Còn trải phổ nhảy tần chậm thì có tốc độ nhảy chậm hơn tốc độ dữ liệu. Hình 3.17 mô tả ví dụ của trải phổ nhảy tần nhanh và trải phổ nhảy tần chậm.



#### Quy trình Giấu tin (Phía Máy phát):

***Bước 1: Chuẩn bị thông tin và Mã hóa (Tùy chọn):***

* Thông tin cần giấu (Chuỗi thông tin mật) được đưa vào.
* Thông tin này có thể được đưa qua **Bộ mã hóa**, nơi nó có thể được mã hóa bằng một khóa riêng để tăng tính bảo mật. Bước này là tùy chọn.

***Bước 2: Điều chế FSK:***

* Tín hiệu (đã mã hóa hoặc chưa) được đưa vào **Bộ điều chế FSK (Frequency Shift Keying)**.
* Bộ điều chế FSK chuyển đổi thông tin thành tín hiệu nhị phân x(t), trong đó mỗi bit (0 hoặc 1) tương ứng với một trong hai tần số cụ thể (ví dụ: f′ cho bit 0 và f′+Δf cho bit 1).

***Bước 3: Tạo chuỗi nhảy tần:***

* Một **Bộ tạo chuỗi giả ngẫu nhiên (PN - Pseudorandom Number)** tạo ra một đoạn mã (ví dụ: m bit).
* Đoạn mã này điều khiển một **Bộ tổ hợp tần số (Frequency Synthesizer)** để tạo ra các giá trị tần số nhảy tần y(t) cho sóng mang. Tần số y(t) thay đổi sau mỗi khoảng thời gian Tc​ giây theo giá trị m bit từ bộ tạo chuỗi PN.

***Bước 4: Trộn tín hiệu:***

* Tín hiệu x(t) (từ bộ điều chế FSK) và tín hiệu tần số nhảy y(t) (từ bộ tổ hợp tần số) được đưa vào một **Bộ trộn (Mixer)**.
* Bộ trộn kết hợp hai tín hiệu này, tạo ra các tần số tổng và hiệu. Một trong hai tần số này sẽ được lọc ra (thường bằng bộ lọc băng thông BPF, không thể hiện rõ trong sơ đồ nhưng là một phần của quá trình trộn chuẩn) trước khi đưa vào môi trường chứa (ví dụ: file âm thanh gốc).
* Về bản chất, thông tin mật sẽ được "trải" trên nhiều tần số khác nhau theo một trình tự giả ngẫu nhiên.

#### Quy trình Tách tin (Phía Máy thu):

***Bước 1: Đồng bộ hóa:***

* Đây là bước cực kỳ quan trọng. **Bộ đồng bộ định thời** đảm bảo rằng bộ tạo chuỗi PN ở phía máy thu hoạt động đồng bộ với phía máy phát.

***Bước 2: Tái tạo chuỗi nhảy tần:***

* **Bộ tạo chuỗi PN** ở máy thu (đã được đồng bộ và sử dụng cùng khóa/hạt giống như máy phát) tạo ra lại cùng một chuỗi giả ngẫu nhiên.
* Chuỗi này điều khiển **Bộ tổ hợp tần số** để tạo ra chính xác cùng một chuỗi tần số nhảy y(t) đã được sử dụng ở máy phát.

***Bước 3: Giải trải phổ (De-hopping):***

* Tín hiệu thu được từ môi trường (file âm thanh chứa tin) được đưa vào **Bộ trộn**.
* Tại đây, tín hiệu thu được sẽ được trộn với tín hiệu tần số nhảy y(t) (tái tạo ở bước 2). Nếu đồng bộ chính xác, quá trình này sẽ loại bỏ thành phần nhảy tần, khôi phục lại tín hiệu FSK x(t) ban đầu (sau khi qua bộ lọc BPF thích hợp).

***Bước 4: Giải điều chế FSK:***

* Tín hiệu x(t) được đưa vào **Bộ giải điều chế FSK**.
* Bộ giải điều chế FSK sẽ chuyển đổi tín hiệu tần số trở lại thành chuỗi bit dữ liệu (đã mã hóa nếu có).

***Bước 5: Giải mã (Tùy chọn):***

* Nếu thông tin đã được mã hóa ở phía máy phát, chuỗi bit dữ liệu thu được sẽ được đưa qua **Bộ giải mã** (sử dụng cùng khóa riêng) để khôi phục lại thông tin mật ban đầu.

#### Xử lý Tín hiệu số Âm thanh và Vai trò của Công cụ GNU Radio:

* Xử lý Tín hiệu số Âm thanh cơ bản: Đề tài yêu cầu làm việc với tín hiệu âm thanh dưới dạng số (chuỗi các mẫu biên độ). Các thao tác cơ bản bao gồm đọc/ghi file âm thanh, hiểu về tần số lấy mẫu, và các biểu diễn tín hiệu cơ bản.
* Công cụ GNU Radio:
  + Là một bộ công cụ phát triển phần mềm miễn phí, mã nguồn mở, cung cấp các khối xử lý tín hiệu (blocks) để xây dựng các hệ thống vô tuyến được định nghĩa bằng phần mềm (SDR) và các ứng dụng xử lý tín hiệu khác.
  + Người dùng thiết kế hệ thống bằng cách kết nối các khối này trong một sơ đồ khối (flowgraph) thông qua GNU Radio Companion (GRC).
* Vai trò của GNU Radio trong đề tài:
  + Cho phép thiết kế, mô phỏng và thực thi hệ thống giấu tin và tách tin FHSS bằng cách sử dụng các khối xử lý tín hiệu có sẵn hoặc tự tạo.
  + Cung cấp môi trường để tạo ra các thành phần của FHSS như bộ tạo chuỗi PN, bộ điều chế FSK, bộ trộn tín hiệu, bộ lọc, và các khối xử lý âm thanh (đọc/ghi file).
  + Hỗ trợ việc kiểm tra và đánh giá hệ thống thông qua các công cụ trực quan hóa tín hiệu.

## Nội dung thực hành

Khởi động bài lab:

* Vào terminal, gõ lệnh:

rebuild giau\_tin\_fhss\_gnuradio

### Nhiệm vụ 1: Đọc tin nhắn cần giấu vào âm thanh

Trước khi thực hiện giấu tin, ta cần đọc trước thông tin sẽ giấu. Thông tin được để trong file message.txt:

***cat message.txt***

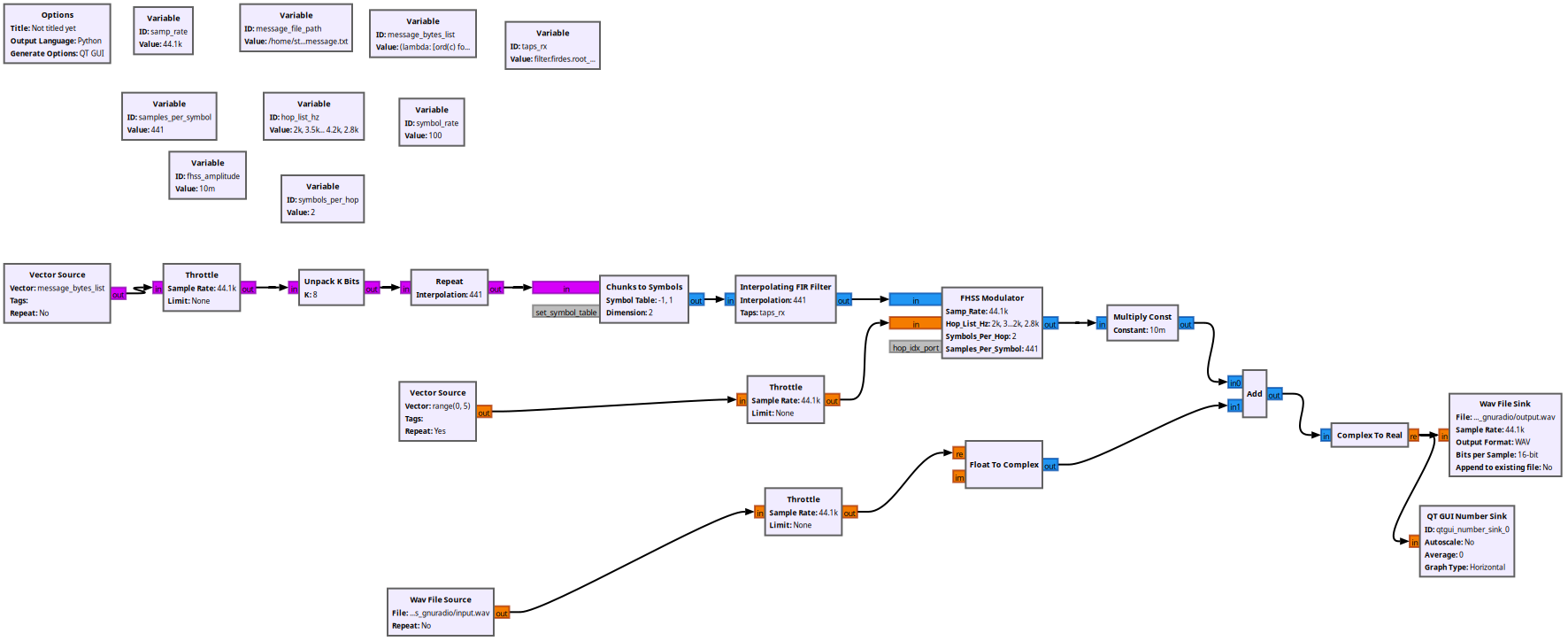
### Nhiệm vụ 2: Thực hiện chạycông cụ GNU-Radio và mở file fhss\_giau.grc

Trên Terminal, nhập lệnh để mở GNU-Radio:

***gnuradio-companion***

Ở trong công cụ, ta sẽ chọn “Open” → “fhss\_giau.grc” hoặc ta có thể dùng lệnh:

***gnuradio-companion fhss\_giau.grc***



Đây là sơ đồ của gru-radio khi ta thực hiện giấu tin, nếu sinh viên thấy những block nào có tên “Missing Block” thì cần thay block đó bằng đúng block như trên. Hãy thêm phù hợp với tên biến có sẵn ở bên trên (không bị thay đổi). Hãy chọn file wav được giấu tin ở Wav File Sink đúng với đường dẫn trong bài lab với tên: “output.wav”

Có thể thêm các QT GUI để theo dõi giá trị nhảy tần như QT GUI Waterfall Sink và QT GUI Frequency Sink ở sau block Add.

### Nhiệm vụ 3: Thực hiện giấu tin

Để thực hiện giấu tin, ta chỉ cần nhấn Execute the flowgraph. Khi đó, ta sẽ thấy QT Number Sink, QT GUI Frequency Sink và QT GUI Waterfall Sink sẽ hiển thị sự thay đổi của tần số và lượng data đang truyền đi.

Sau khi flowgraph chạy hết, sẽ xuất ra một file output.wav tong thư mục vủa bài lab.

### Nhiệm vụ 3: So sánh kích thước của file âm thanh gốc và file âm thanh sau khi giấu tin

Để thực hiện so sánh kích thước của 2 file âm thanh, ta sẽ sử dụng file python có sẵn để so sánh sự khác biệt giữa kích thước

***python3 compare\_file\_size.py input.wav output.wav***

Sau khi chạy python để so sánh, hệ thống sẽ in ra kích thước của 2 file và so sánh 2 file lớn hơn/bé hơn bao nhiêu so với file còn lại

### Nhiệm vụ 4: So sánh BER (Bit Error Rate) của file âm thanh gốc và file âm thanh sau khi giấu tin

Để thực hiện so sánh BER của 2 file âm thanh, ta sẽ sử dụng file python có sẵn để so sánh sự khác biệt bit error rate

***python3 compare\_file\_ber.py input.wav output.wav***

Sau khi chạy python, hệ thống sẽ in ra thông tin frequency samples, kiểu dữ liệu và số kênh của 2 file âm thanh

Sau đó hệ thống sẽ chỉ ra số bit lỗi giữa âm thanh gốc và âm thanh đã giấu tin. Và cuối cùng là tỷ lệ lỗi bit của dữ liệu âm thanh (bit)

## Kết quả cần đạt được

* Chạy được tất cả các bước như yêu cầu.

## Kết thúc bài lab:

Sử dụng lệnh: “checkwork” để kiểm tra số checkwork đã làm đúng.

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

***stoplab giau\_tin\_fhss\_tool***

Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh:

***checkwork giau\_tin\_fhss\_tool***

Khởi động lại bài lab: Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

***labtainer –r giau\_tin\_fhss\_tool***

Link bài lab: <https://github.com/hieubabe123/Ky-thuat-giau-tin-trong-am-thanh.git>