**Kỹ thuật thủy vân số với dãy trực tiếp (DSSS)**

1. **Mục tiêu**

Trong bài lab này, chúng ta sẽ tìm hiểu và thực hành giấu và tách thủy vân số để bảo vệ bản quyền bằng kỹ thuật DSSS vào file âm thanh.

1. **yêu cầu đối với sinh viên**

Có kiến thức cơ bản về hệ điều hành linux, ngôn ngữ Python.

Có kiến thức cơ bản về phổ, trải phổ và DSSS, truyền thông

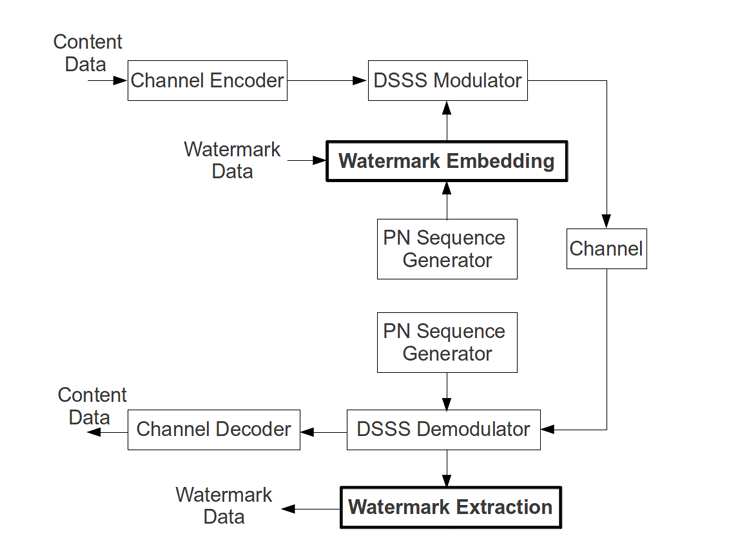
Có kiến thử về thủy vân số.

1. **Giới thiệu chung về bài thực hành**

**Thủy vân số (Digital Watermarking)** là phương pháp nhúng thông tin ẩn (watermark data) vào tín hiệu gốc (content data) để bảo vệ bản quyền hoặc xác thực. Trong bài lab, thủy vân được tích hợp vào tín hiệu DSSS, sử dụng âm thanh làm sóng mang, và sau đó trích xuất để khôi phục dữ liệu.

**Nguyên lý hoạt động:**

* **Phía truyền**: Dữ liệu nội dung được mã hóa kênh, nhúng thủy vân bằng mã PN, sau đó modul hóa DSSS để truyền qua kênh.
* **Phía nhận**: Tín hiệu nhận được được demodul hóa DSSS, trích xuất thủy vân, và giải mã để khôi phục dữ liệu.



**Máy phát (Phía Alice)**

* **Channel Encoder (Mã hóa kênh)**: Chuyển đổi dữ liệu nội dung (content data) từ file âm thanh thành định dạng phù hợp để modul hóa, lưu vào tệp wav\_output.json.
* **Watermark Embedding (Nhúng thủy vân)**: Nhúng dữ liệu thủy vân (watermark data) từ file watermark.txt vào tín hiệu, sử dụng mã PN từ hai bộ **PN Sequence Generator** để tạo tín hiệu ẩn.
* **DSSS Modulator (Modul hóa DSSS)**: Kết hợp tín hiệu nội dung và thủy vân, modul hóa bằng kỹ thuật DSSS để tạo tín hiệu truyền qua kênh.
* **Channel (Kênh truyền)**: Tín hiệu được truyền qua kênh (ở đây là file âm thanh).

**Máy nhận (Phía Bob)**

* **DSSS Demodulator (Demodul hóa DSSS)**: Nhận tín hiệu từ kênh, demodul hóa để tách tín hiệu DSSS, lưu kết quả vào dsss\_demodule\_output.txt.
* **Watermark Extraction (Trích xuất thủy vân)**: Sử dụng mã PN để trích xuất dữ liệu thủy vân từ tín hiệu nhận, tạo hình ảnh watermark\_output.png.
* **Channel Decoder (Giải mã kênh)**: Giải mã tín hiệu nội dung để khôi phục dữ liệu gốc từ file âm thanh.

1. **Nội dung thực hành**

**Tại máy Alice**

1. **Mã hóa kênh**

python3 channel\_encoder.py -l input1.wav

* + Kết quả: Tệp wav\_output.json chứa dữ liệu âm thanh đã mã hóa.

cat wav\_output.json

1. **Nhúng thủy vân**

python3 watermark\_embedding.py -w watermark.txt

* + Kết quả: Tệp watermark\_data.json chứa dữ liệu thủy vân.

cat watermark\_data.json

1. **Modul hóa DSSS**

python3 dsss\_module.py -e wav\_output.json watermark\_data.json

* + Kết quả: Tệp dsss\_module\_output.wav chứa tín hiệu DSSS, hình ảnh audio\_comparison.png so sánh tín hiệu.
  + Kiểm tra:

feh audio\_comparison.png

tar czf file.tar.gz dsss\_module\_output.wav params.json

ncat --send-only 172.10.0.11 4444 < file.tar.gz

* + Kết quả: File file.tar.gz được gửi qua mạng.

**Tại máy Bob**

ncat -l 4444 > file.tar.gz

tar -xzf file.tar.gz

1. **Demodul hóa DSSS**

python3 dsss\_demodule.py -d dsss\_module\_output.wav params.json

* + Kết quả: Tệp dsss\_demodule\_output.txt chứa tín hiệu demodul hóa, hình ảnh watermark\_output.png hiển thị thủy vân.
  + Kiểm tra:

cat dsss\_demodule\_output.txt

feh watermark\_output.png

Kết thúc bài lab:

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

*Stoplab dsss\_lab*

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

Khởi động lại bài lab:

Trong quá trình làm bài, nếu muốn thực hiện lại bài lab, nhập lệnh:

*Labtainer -r dsss\_lab*