

Giấu tin sử dụng kỹ thuật trải phổ dây trực tiếp (DSSS)

1. Mục tiêu

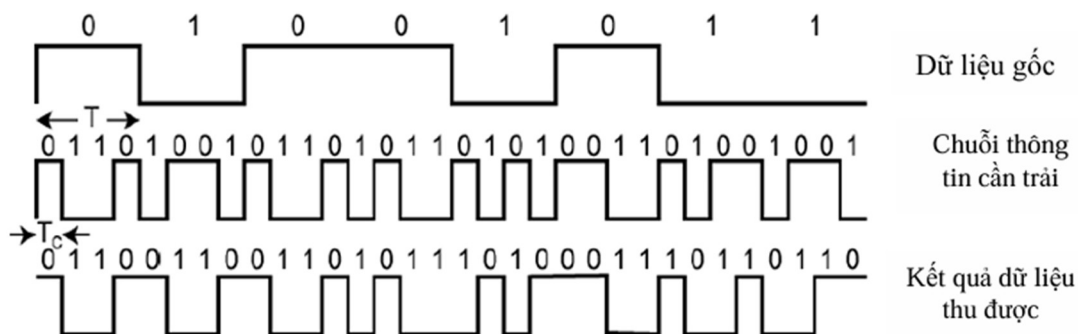
Trong bài lab này, chúng ta sẽ tìm hiểu và thực hành giấu tin bằng phương pháp Direct Sequence Spread Spectrum (DSSS), một kỹ thuật dùng trong truyền dữ liệu và bảo mật thông tin.

2. Yêu cầu đối với sinh viên

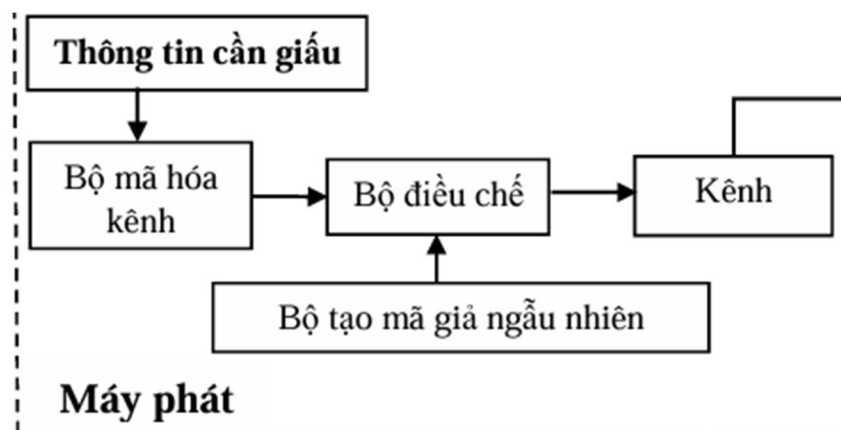
Có kiến thức cơ bản về hệ điều hành linux, ngôn ngữ Python.

3. Tổng quan về DSSS

DSSS là một phương pháp truyền dữ liệu trong đó hệ thống truyền và nhận đều sử dụng một tập các tần số có độ rộng 22 MHz. Trải phổ dây trực tiếp được thực hiện bằng cách áp dụng một tín hiệu trải phổ phụ là dãy có tần số cao hơn khỏi dữ liệu cần truyền lên tín hiệu gốc. Việc phân phối tín hiệu phụ này trên băng tần truyền tải sẽ giúp tăng độ rộng của băng tần và giảm độ nhiễu.



Hình 1 Minh họa trải phổ dây trực tiếp



Hình 2 Quy trình giấu tin trong DSSS

4. Các bước thực hiện

- Khởi động lab:

- Chạy lệnh: *labtainer -r dsss_lab*

- trong terminal của Labtainer

- Tình huống và yêu cầu:

1. Tìm hiểu quy trình của DSSS

- Sau khi khởi động lab, một máy ảo sẽ xuất hiện.

- Chạy lệnh: *python3 dsss.py -h*

để xem hướng dẫn thực hiện quy trình giấu tin trong DSSS.

2. Kênh mã hóa thông điệp

- Trước tiên dữ liệu cần giấu được mã hóa bằng bộ mã hóa kênh để đưa vào các bit dư nhằm mục đích phát hiện hay sửa các lỗi có thể phát sinh khi truyền dẫn tín hiệu qua kênh tần số vô tuyến.

- Trong bài lab này, thông điệp cần giấu được chuyển thành dạng nhị phân, sau đó đi qua mã hóa hamming trước khi được đưa vào bộ điều chế.

- Để tạo file đầu vào chứa thông điệp cần giấu tin (Có thể thay đổi thông điệp).

echo <thong diep can dau > > sec.txt

- Tiếp theo thực hiện đưa thông điệp qua kênh mã hóa

python3 dsss.py -m sec.txt

Sau khi đưa thông điệp qua kênh mã hóa. Có thể xem đầu ra của thông điệp bằng lệnh “*cat message_output.txt*”.

3. Xử lý âm thanh đầu vào

- Âm thanh đầu vào là tín hiệu gốc (thông tin ban đầu) được chuẩn bị để làm "vỏ bọc" cho thông điệp cần giấu.

- Âm thanh đầu vào (*audio_data*) đóng vai trò là **tín hiệu mang** (carrier signal) để giấu thông điệp.

- Thông điệp (sau khi mã hóa và điều chế) được cộng trực tiếp vào âm thanh đầu vào, tạo ra một tín hiệu âm thanh mới (hidden_signal) chứa thông điệp ẩn.
- Âm thanh đầu vào không bị thay đổi cấu trúc cơ bản (vẫn giữ tần số mẫu và định dạng), nhưng giá trị biên độ của nó được điều chỉnh để chứa thông điệp ẩn.

Python3 dsss.py -l input.wav,

- Sau khi thực hiện xong, hàm sẽ trả về file .json chứa các thông tin quan trọng như dữ liệu về âm thanh dưới dạng ma trận và tần số mẫu của file âm thanh đó, điều này là tiền điều kiện để thực hiện tạo sóng mang thông tin cần truyền.
- Đọc file .json bằng lệnh “***cat wav_output.json***” để xem các thông tin tiền xử lý file âm thanh.

4. Giấu tin

- Thông điệp sau khi được mã hóa sẽ được đưa qua bộ điều chế để thực hiện trải phổ.
- Tại bộ điều chế, sẽ có 3 bước là tạo mã giả ngẫu nhiên (PN), dịch pha BPSK và điều chế sóng mang, sau đó mới thực hiện trải phổ thông tin.
- Để xem các bước, thực hiện lệnh “***cat dsss.py***” tìm đến hàm “***encode_dsss***”.
- Thông điệp sẽ được dịch pha với phương pháp BPSK, cụ thể chuyển các bit 0 thành -1, các bit 1 thành 1. Sau đó, bộ điều chế sẽ tạo sóng mang dựa theo công thức có sẵn. Cuối cùng tín hiệu sẽ được trải phổ vào trong file âm thanh dựa theo các thông tin mà bước 3 đã thực hiện.
- Để thực hiện giấu tin, chạy lệnh :

Python3 dsss.py -e wav_output.json message_output.txt.

Hàm sẽ trả về file âm thanh mang theo thông tin được trải phổ và các thông tin quan trọng khác để thực hiện việc tách tin.

Kết thúc bài lab:

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

Stoplabs 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