**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: GIẤU TIN VÀ TÁCH TIN TRONG ÂM THANH**

**BẰNG TOOL WAV-STEG**

**Lab: lsb\_tool**

| 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 : Giấu tin và tách tin trong âm thanh bằng tool Wav-Steg

## Mục đích

* Giúp sinh viên hiểu được thuật toán giấu tin và tách tin trong âm thanh sử dụng phương pháp LSB
* Sinh viên làm quen với công cụ giấu tin và tách tin Wav-steg

## Yêu cầu đối với sinh viên

* Có hiểu biết cơ bản về các khái niệm trong lĩnh vực an toàn thông tin, đặc biệt là kỹ thuật giấu tin (Steganography) và phát hiện tin giấu (Steganalysis)
* Hiểu rõ các nguyên lý cơ bản của việc xử lý tín hiệu số liên quan đến âm thanh
* Có khả năng cài đặt và sử dụng thành thạo công cụ WAV-STEG để thực hiện các thao tác giấu tin vào file âm thanh và tách tin đã giấu ra khỏi file âm thanh

## Nội dung lý thuyết

LSB là một kỹ thuật giấu tin (steganography) đơn giản và phổ biến, hoạt động dựa trên việc thay thế các bit ít quan trọng nhất của dữ liệu gốc (gọi là "phương tiện chứa" hay "cover media" như ảnh, âm thanh, video) bằng các bit của thông điệp bí mật.

Trong biểu diễn nhị phân của một số (ví dụ, một mẫu màu của pixel ảnh, hoặc một mẫu biên độ của tín hiệu âm thanh), LSB là bit nằm ở vị trí ngoài cùng bên phải. Việc thay đổi bit này chỉ làm thay đổi giá trị của toàn bộ số đó đi một lượng rất nhỏ. Ví dụ, nếu có số nhị phân 1100101\*\*0\*\* (202 ở hệ thập phân), bit LSB là 0. Nếu đổi nó thành 1 (1100101\*\*1\*\* - 203 ở hệ thập phân), giá trị chỉ thay đổi 1 đơn vị. Đối với các giá trị lớn hơn, sự thay đổi này càng khó nhận biết.

Kỹ thuật này tận dụng tính chất "ít quan trọng" của LSB. Người ta sẽ lấy từng bit của thông điệp bí mật và lần lượt thay thế nó vào bit LSB của từng đơn vị dữ liệu trong phương tiện chứa (ví dụ, từng pixel trong ảnh, từng mẫu trong file âm thanh). Vì việc thay đổi LSB chỉ gây ra một sự thay đổi rất nhỏ trong giá trị gốc, nên đối với tai người (trong âm thanh) hoặc mắt người (trong hình ảnh), sự khác biệt này thường rất khó hoặc không thể nhận ra, miễn là lượng thông tin giấu vào không quá lớn làm biến đổi nhiều LSB.

Những tùy chọn khi sử dụng lệnh của tool Wav-Steg:

* h, --hide: If present, the script runs to hide data
* r, --recover: If present, the script runs to recover data
* s, --sound: What follows is the name of carrier wav file
* d, --data: What follows is the file name having data to hide
* o, --output: Output filename of choice
* n, --nlsb: Number of LSBs to use
* b, --bytes: Number of bytes to recover
* --help: Display help

## Nội dung thực hành

Khởi động bài lab:

* Vào terminal, gõ lệnh:

rebuild lsb\_tool

### Nhiệm vụ 1: Kiểm tra kích thước tệp tin gốc và đọc thông tin cần giấu

Chạy lệnh “python3 check\_file\_size <tệp thông tin txt gốc>” để kiểm tra kích thước tệp txt gốc, điều này để xác định kích thước tệp nhằm hỗ trợ việc tách tin:

***python3 check\_file\_size.py data.txt***

Sau đó đọc thông tin cần giấu để khi tách tin, sinh viên cần chắc chắn rằng khi tách tin thì tin được tách phải giống hệt tin gốc:

***cat data.txt***

### Nhiệm vụ 2: Thực hiện giấu tin bằng tool Wav-Steg

Để giấu tin vào âm thanh, ta cần sử dụng các tùy chọn của tool Wav-steg bao gồm: “-h”, “-d”, “-s”, “-o”, “-n”. Câu lệnh “python3 wav-steg.py -h -d <tệp thông tin txt gốc> -s <file âm thanh Wav gốc> -o <file âm thanh Wav được giấu tin> -n <Số lượng LSBs được sử dụng> ” để thực hiện giấu tin vào file âm thanh gốc và xuất ra một file Wav mới:

***python3 wav-steg.py -h -d data.txt -s opera.wav -o opera\_output.wav -n 2***

Nếu lượng thông tin cần giấu quá dài, tool sẽ đưa ra cảnh báo cần nhập số lượng LSBs lớn hơn để thực hiện việc giấu tin.

### Nhiệm vụ 3: Thực hiện việc tách tin bằng tool Wav-Steg

Trước khi tách tin bằng tool, ta cần nhớ kích thước file thông tin txt gốc ta vừa kiểm tra ở nhiệm vụ 1.Để tách tin vào âm thanh, ta cần sử dụng các tùy chọn của tool Wav-steg bao gồm: “-r”, “-s”, “-o”, “-n”, “-b” Sử dụng câu lệnh: “python wav-steg.py -r -s <file âm thanh Wav được giấu tin> -o <tệp thông tin txt sau khi tách> -n <Số lượng LSBs được sử dụng để giấu> -b <kích thước tệp tin gốc >”:

***python wav-steg.py -r -s opera\_output.wav -o data\_output.txt -n 2 -b 68***

Sau khi tách tin, chúng ta sẽ có file thông tin tách được lưu vào file output.txt. Thử đọc file đó xem thông tin sau khi tách có giống file gốc không?

### Nhiệm vụ 4: So sánh kích thước của 2 file thông tin .txt

Sau khi tách tin, ta thực hiện so sánh kích thước file .txt thông tin gốc với .txt file thông tin sau khi tách tin. Sử dụng lệnh: “python3 compare\_wav\_files <file .txt gốc> <file .txt sau tách> để biết được dung lượng của 2 file:

***python3 compare\_wav\_files.py opera.wav opera\_output.wav***

## 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 lsb\_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 lsb\_tool***

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

***labtainer –r lsb\_tool***