

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA AN TOÀN THÔNG TIN**



**HƯỚNG DẪN BÀI THỰC HÀNH  
HỌC PHẦN: CÁC KỸ THUẬT GIẤU TIN  
MÃ HỌC PHẦN: INT14102**

**NHÓM LỚP: D21CQAT01 - B**  
**TÊN BÀI: Phương pháp giấu tin LSB trong audio**

Sinh viên thực hiện:

B21DCAT193 Mai Đức Trung

Giảng viên: Đỗ Xuân Chợt

**HỌC KỲ 2 NĂM HỌC 2024-2025**

# MỤC LỤC

MỤC LỤC .....	2
DANH MỤC CÁC HÌNH VẼ.....	3
DANH MỤC CÁC TỪ VIẾT TẮT.....	4
<b>CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH</b> .....	5
1.1 Mục đích.....	5
1.2 Tìm hiểu lý thuyết .....	5
<b>CHƯƠNG 2. NỘI DUNG THỰC HÀNH</b> .....	6
2.1 Chuẩn bị môi trường .....	6
2.2 Các bước thực hiện.....	6
<b>2.2.1</b> Khởi động bài lab .....	6
<b>2.2.2</b> Thực hiện các nhiệm vụ .....	6
<b>CHƯƠNG 3. Truy vết thực hành</b> .....	8
3.1 Checkwork .....	8
<b>TÀI LIỆU THAM KHẢO</b> .....	10

## DANH MỤC CÁC HÌNH VẼ

Hình 1 - Checkwork mã sinh viên.....	8
Hình 2 – Checkwork chuẩn bị trước khi giấu tin .....	8
Hình 3 – Checkwork giấu tin hoàn tất.....	8
Hình 4 – Checkwork so sánh file trước và sau giấu tin .....	9
Hình 5 – Checkwork chuyển đổi file thông điệp sang nhị phân .....	9

## DANH MỤC CÁC TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Thuật ngữ tiếng Anh/Giải thích</b>	<b>Thuật ngữ tiếng Việt/Giải thích</b>
LSB	Least Significant Bit	Bit ít quan trọng nhất trong một byte (bit thấp nhất, thường là bit phải cùng)
MSB	Most Significant Bit	Bit quan trọng nhất trong một byte (bit cao nhất, thường là bit trái cùng)
MP3	MPEG-1 Audio Layer III	Định dạng nén âm thanh có tổn hao (lossy), rất phổ biến nhờ dung lượng nhỏ
WAV	Waveform Audio File Format	Định dạng âm thanh chuẩn của Microsoft, thường lưu trữ âm thanh thô (uncompressed)
PCM	Pulse Code Modulation	Phương pháp mã hóa tín hiệu âm thanh analog thành kỹ thuật số không nén

# CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH

## 1.1 Mục đích

Giúp sinh viên thực hành giấu tin bằng phương pháp LSB vào audio

Trong bài thực hành này sinh viên sẽ thực hành giấu tin vào file audio bằng phương pháp sử dụng 8 bit MSB và 3 bit LSB

## 1.2 Tìm hiểu lý thuyết

Phương pháp giấu tin bằng bit ít quan trọng nhất (LSB) là một kỹ thuật steganography phổ biến được áp dụng trong các tệp âm thanh kỹ thuật số. Nguyên lý hoạt động của phương pháp này dựa trên việc thay thế bit cuối cùng trong mỗi mẫu tín hiệu âm thanh (sample) bằng các bit của thông điệp cần giấu. Do bit LSB chỉ ảnh hưởng rất nhỏ đến giá trị của mẫu âm thanh, nên việc thay đổi nó hầu như không làm thay đổi chất lượng âm thanh mà tai người có thể nhận biết. Quá trình giấu tin diễn ra bằng cách duyệt qua từng sample trong tệp audio và chèn dần từng bit thông điệp vào các vị trí LSB tương ứng. Quá trình trích xuất ngược lại bằng cách đọc các bit LSB từ tệp âm thanh đã được nhúng và ghép lại thành thông điệp gốc.

Ưu điểm lớn nhất của phương pháp LSB là tính đơn giản trong cài đặt và khả năng giấu tin mà không làm ảnh hưởng rõ rệt đến chất lượng âm thanh gốc, đặc biệt khi sử dụng các định dạng không nén như WAV hoặc PCM. Tuy nhiên, phương pháp này cũng tồn tại những hạn chế nhất định. Dữ liệu giấu có thể bị tổn hại hoặc mất hoàn toàn nếu file âm thanh bị nén lại (như chuyển sang định dạng MP3) hoặc chỉnh sửa. Ngoài ra, nếu không có biện pháp bảo vệ bổ sung như mã hóa hoặc giấu vị trí ngẫu nhiên, phương pháp LSB có thể dễ bị phát hiện và tấn công khi kẻ xấu nghi ngờ đến sự tồn tại của thông tin ẩn.

## CHƯƠNG 2. NỘI DUNG THỰC HÀNH

### 2.1 Chuẩn bị môi trường

- Máy ảo Ubuntu có cài đặt labtainer

### 2.2 Các bước thực hiện

#### 2.2.1 Khởi động bài lab

Tải bài lab, gõ:

*imodule*

*[https://github.com/mdt12345689/Custom\\_Lab/raw/refs/heads/main/steg\\_lsb73/imodule.tar](https://github.com/mdt12345689/Custom_Lab/raw/refs/heads/main/steg_lsb73/imodule.tar)*

Vào terminal, gõ:

*labtainer -r steg\_lsb73*

(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)

#### 2.2.2 Thực hiện các nhiệm vụ

- Trước khi giấu tin, sinh viên cần nhập thông điệp cần giấu vào file message.txt, ở bài lab này sinh viên cần nhập MSV theo dạng viết hoa (VD: B21DCAT193)

*nano message.txt*

- Trong bài lab này, chương trình giấu tin mặc định chỉ hỗ trợ giấu tin trên file WAV 16-bit. Vì vậy trước khi giấu tin, sinh viên cần phải chuyển đổi file MP3 có sẵn thành định dạng file audio thỏa mãn yêu cầu.
- Ở đây chúng ta sẽ thực hiện định dạng lại file audio thành 2 dạng file audio stereo và mono.

*ffmpeg -i miniTalk.mp3 -ac 1 miniTalk1.wav*

*ffmpeg -i miniTalk.mp3 -ac 2 miniTalk2.wav*

- Tạo file binary từ thông điệp

*python3 text\_to\_bin.py -f <file thông điệp> -o <tên file output>*

*python3 text\_to\_bin.py -f message.txt*

- Sau khi đã có file đúng định dạng, sinh viên bắt đầu thực hiện giấu tin vào file audio với cấu trúc lệnh.

```
python3 steganography.py -f <file chứa thông điệp> -b<file thông điệp binary> -s <file audio> -o <tên file audio output>
```

- Sinh viên thực hiện giấu tin vào 2 file audio stereo và mono đã tạo bằng file thông điệp gốc và bằng file nhị phân.

```
python3 steganography.py -f message.txt -s miniTalk1.wav -o output1.wav
```

```
python3 steganography.py -b bin_mess.txt -s miniTalk2.wav -o output2.wav
```

- Sau khi hoàn thành giấu tin, ta tiến hành so sánh dung lượng của file audio trước và sau khi giấu tin.

```
ls -l
```

- Với phương pháp LSB thì nó sẽ không làm thay đổi dung lượng của file thông tin gốc, tuy nhiên ta lại thấy một sự thay đổi nhỏ trong dung lượng của file audio trước và sau khi giấu tin.
- Lý do là công cụ ffmpeg khi chuyển đổi file MP3 về dạng file WAV sẽ tự thêm các header chunk định nghĩa các thông tin của âm thanh, còn khi chạy chương trình giấu tin sử dụng thư viện wave sẽ chỉ ghi lại phần cơ bản nhất của header nên sẽ nhẹ hơn. Tóm lại nó chỉ thay đổi phần header của file âm thanh chứ không ảnh hưởng đến dung lượng chính của âm thanh gốc.
- Để kiểm tra sự khác nhau của header, ta dùng lệnh:

```
xxd miniTalk1.wav |head -n 10
```

```
xxd output1.wav |head -n 10
```

- Kiểm tra kết quả bằng lệnh :

```
checkwork
```

- Kết thúc bài lab:

```
Stoplalab
```

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

## CHƯƠNG 3. TRUY VẾT THỰC HÀNH

### 3.1 Checkwork

- Hệ thống sẽ truy vết mã sinh viên của sinh viên nhập trong file thông điệp message.txt

1 msv steg\_lsb73 message.txt FILE\_REGEX Field ID ^Bvd(2)DC.{5,}\$ Timestamp Type File

Hình 1 - Checkwork mã sinh viên

- Hệ thống sẽ truy vết task chuẩn bị trước file audio của sinh viên qua output của lệnh ffmpeg

2 pre\_steg\_stereo steg\_lsb73 mp/ffmpeg\_output.log CONTAINS Field ID stereo, s16, 1411 kb/s Timestamp Type File

3 pre\_steg\_mono steg\_lsb73 mp/ffmpeg\_output.log CONTAINS Field ID mono, s16, 705 kb/s Timestamp Type File

Hình 2 – Checkwork chuẩn bị trước khi giấu tin

- Hệ thống sẽ truy vết task giấu tin vào file của sinh viên qua output của quá trình chạy chương trình python cung cấp sẵn. Các vết cần phải có là thông báo giấu tin thành công và có giấu các ký tự trong mã sinh viên được ghi trong file thông điệp dưới định dạng <BxxDCxxxxxx>

6 \_steg1 steg\_lsb73 np/python3\_output.log CONTAINS Field ID Successfully Timestamp Type File

7 \_steg2 steg\_lsb73 np/python3\_output.log CONTAINS Field ID MSB: 66 (Character 'B') Timestamp Type File

8 \_steg3 steg\_lsb73 np/python3\_output.log CONTAINS Field ID MSB: 68 (Character 'D') Timestamp Type File

9 \_steg4 steg\_lsb73 np/python3\_output.log CONTAINS Field ID MSB: 67 (Character 'C') Timestamp Type File

1 steg\_done Goal Type boolean Boolean (\_steg1 and \_steg2 and \_steg3 and \_steg4) Boolean Result Tags msv

Hình 3 – Checkwork giấu tin hoàn tất



- Hệ thống truy vết task so sánh dung lượng file trước và sau khi giấu tin qua output của lệnh xxd

The screenshot displays two task configurations in the Checkwork interface:

- Task 4:**
  - Input: `_check_header1`
  - Tool: `steg_isb73`
  - Output: `/tmp/xxd_output.log`
  - Operation: `CONTAINS`
  - Field ID: `Lavf58.2`
  - Timestamp Type: `File`
- Task 5:**
  - Input: `_check_header2`
  - Tool: `steg_isb73`
  - Output: `/tmp/xxd_output.log`
  - Operation: `CONTAINS`
  - Field ID: `WAVEfmt`
  - Timestamp Type: `File`

Below these, a configuration for a goal is shown:

- Goal 2:**
  - Goal Type: `boolean`
  - Boolean Expression: `(_check_header1 and _check_header2)`
  - Boolean Result Tags: `msv`

Hình 4 – Checkwork so sánh file trước và sau giấu tin

- Hệ thống truy vết task chuyển đổi file thông điệp sang dạng nhị phân

The screenshot displays a single task configuration in the Checkwork interface:

- Task 2:**
  - Input: `convert_mess`
  - Tool: `steg_isb73`
  - Output: `np/python3_output.log`
  - Operation: `CONTAINS`
  - Field ID: `rted message to binary`
  - Timestamp Type: `File`

Hình 5 – Checkwork chuyển đổi file thông điệp sang nhị phân

## **TÀI LIỆU THAM KHẢO**

- [1] Giáo trình Các kỹ thuật giấu tin, Đỗ Xuân Chợt, Học viện Công Nghệ Bưu Chính Viễn Thông.