

**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: So sánh các 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
CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH.....	5
1.1 Mục đích.....	5
1.2 Tìm hiểu lý thuyết.....	5
1.2.1 Phương pháp LSB	5
1.2.2 Các phương pháp được sử dụng	5
CHƯƠNG 2. NỘI DUNG THỰC HÀNH.....	7
2.1 Chuẩn bị môi trường	7
2.2 Các bước thực hiện	7
2.2.1 Khởi động bài lab.....	7
2.2.2 Thực hiện các nhiệm vụ	7
CHƯƠNG 3. Truy vết thực hành	10
3.1 Checkwork.....	10
TÀI LIỆU THAM KHẢO	11

DANH MỤC CÁC HÌNH VẼ

Hình 1 Phương pháp giấu tin qua 8 bit LSB	5
Hình 2 Phương pháp giấu tin qua 8 bit MSB và 3 bit LSB	6
Hình 3 Phương pháp giấu tin qua 8 bit MSB và 4 bit LSB	6
Hình 4 Checkwork giấu tin	10
Hình 5 Checkwork tạo file spectrogram	10
Hình 6 Checkwork so sánh spectrogram trước và sau giấu tin	10
Hình 7 Checkwork kiểm tra LSB	10

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Thuật ngữ tiếng Anh/Giải thích	Thuật ngữ tiếng Việt/Giải thích
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 so sánh được các điểm mạnh, điểm yếu của mỗi phương pháp giấu tin LSB vào audio

Trong bài thực hành này sinh viên sẽ thực hành so sánh 3 phương pháp giấu tin LSB khác nhau yêu cầu âm thanh đầu vào 16-bit. Các phương pháp trong bài lab bao gồm:

- Giấu tin qua 8 bit LSB.
- Giấu tin qua 8 bit MSB và 3 bit LSB
- Giấu tin qua 8 bit MSB và 4 bit LSB.

1.2 Tìm hiểu lý thuyết

1.2.1 Phương pháp LSB

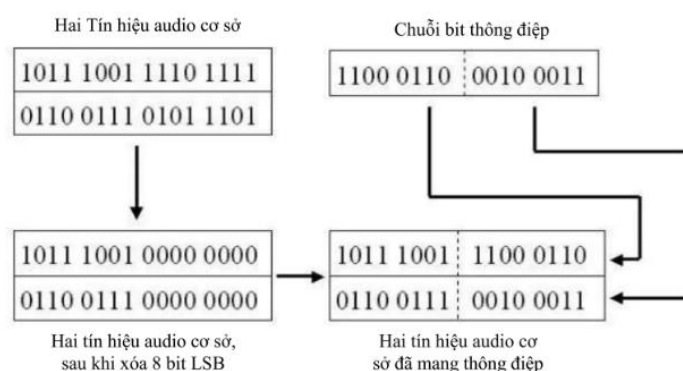
Phương pháp giấu tin bằng bit ít quan trọng nhất (LSB) là một kỹ thuật steganography phổ biến được áp dụng trong các tệp âm thanh kỹ thuật số. Nguyên lý hoạt động của phương pháp này dựa trên việc thay thế bit cuối cùng trong mỗi mẫu tín hiệu âm thanh (sample) bằng các bit của thông điệp cần giấu. Quá 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, 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, phương pháp LSB có thể dễ bị phát hiện khi kẻ xấu nghi ngờ đến sự tồn tại của thông tin ẩn.

1.2.2 Các phương pháp được sử dụng

1.2.2.1 Giấu tin qua 8 bit LSB.

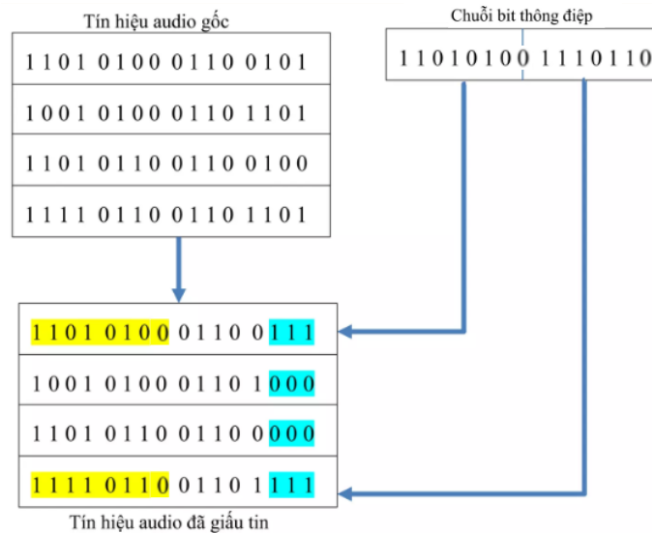
Phương pháp này trong bài lab được sử dụng để mã hóa mỗi ký tự thành 8 bit nhị phân và giấu nó vào 8 bit LSB của từng mẫu âm thanh audio một cách tuần tự



Hình 1 Phương pháp giấu tin qua 8 bit LSB

1.2.2.2 Giấu tin qua 8 bit MSB và 3 bit LSB

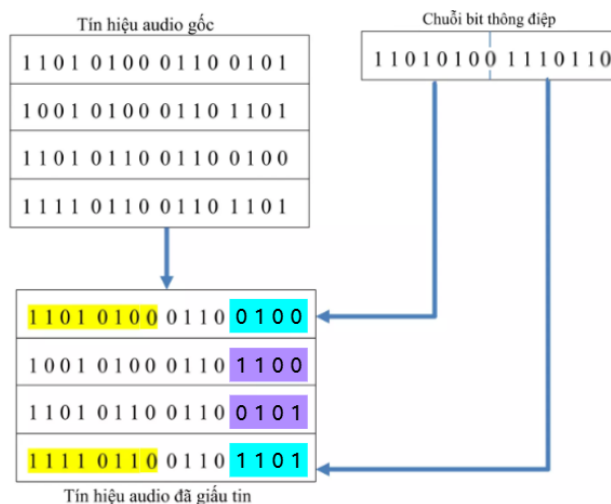
Phương pháp này trong bài lab sẽ không thực sự thay đổi các bit theo giá trị nhị phân của các kí tự trong thông điệp mà sẽ sử dụng 8 bit MSB như một tham chiếu, so sánh nó với 8 bit nhị phân kí tự trong thông điệp. Nếu 8 bit thông điệp trùng 8 bit MSB thì sẽ đánh dấu 3 bit LSB là 111, ngược lại không trùng thì đánh dấu 3 bit LSB là 000.



Hình 2 Phương pháp giấu tin qua 8 bit MSB và 3 bit LSB

1.2.2.3 Giấu tin qua 8 bit MSB và 4 bit LSB

Phương pháp này trong bài lab sẽ không thực sự thay đổi các bit theo giá trị nhị phân của các kí tự trong thông điệp mà sẽ sử dụng 8 bit MSB như một tham chiếu, so sánh nó với 8 bit nhị phân kí tự trong thông điệp. Nếu 8 bit thông điệp trùng 8 bit MSB thì sẽ thay đổi ít nhất sao cho số bit 1 trong 4 bit LSB là lẻ, ngược lại không trùng thì thay đổi ít nhất sao cho số bit 1 trong 4 bit LSB là chẵn.



Hình 3 Phương pháp giấu tin qua 8 bit MSB và 4 bit LSB

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/compare_lsb/imodule.tar

- Vào terminal, gõ:

labtainer -r compare_lsb

*(Chú ý: sinh viên sử dụng **mã sinh viên viết hoa** của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)*

- Để tạo thêm terminal, sinh viên có thể chạy lệnh sau trên labtainer:

moreterm.py compare_lsb compare_lsb

2.2.2 Thực hiện các nhiệm vụ

- Xem thử nội dung của 3 file mess*.txt

cat mess1.txt

cat mess2.txt

cat mess3.txt

- Tạo file audio giấu tin với mỗi phương pháp giấu tin, cú pháp chạy

python3 <script giấu tin> -f <file thông điệp> -s <file audio> -o <tên file output>

python3 <script tách tin> -s <file audio>

- Thực hiện giấu tin và tách tin mess1.txt với phương pháp 8 bit LSB.

cd ~/lsb8

python3 embed8_audio.py -f ~/mess1.txt -s ~/miniTalk.wav -o output8.wav

python3 extract8_audio.py -s output8.wav

- Thực hiện giấu tin và tách tin mess1.txt với phương pháp 8 bit MSB và 3 bit LSB.

```
cd ~/lsb83
```

```
python3 embed83_audio.py -f ~/mess1.txt -s ~/miniTalk.wav -o output83.wav
```

```
python3 extract83_audio.py -s output83.wav
```

- Thực hiện giấu tin và tách tin mess1.txt với phương pháp 8 bit MSB và 4 bit LSB.

```
cd ~/lsb84
```

```
python3 embed84_audio.py -f ~/mess1.txt -s ~/miniTalk.wav -o output84.wav
```

```
python3 extract84_audio.py -s output84.wav
```

- Tiếp tục thực hiện giấu tin bởi 3 phương pháp được cung cấp với thông điệp mess2.txt và mess3.txt. Chú ý xem khả năng giấu tin của các phương pháp trên với các độ dài thông điệp khác nhau. (Lưu ý đặt tên file audio output khác với file output đã tạo với mess1.txt)

- Thực hiện tạo dạng ảnh spectrogram cho file audio gốc và các file audio đã tạo

```
ffmpeg -i miniTalk.wav -lavfi showspectrumpic=s=hd480 output.png
```

```
ffmpeg -i ~/lsb8/output8.wav -lavfi showspectrumpic=s=hd480 output8.png
```

```
ffmpeg -i ~/lsb83/output83.wav -lavfi showspectrumpic=s=hd480 output83.png
```

```
ffmpeg -i ~/lsb84/output84.wav -lavfi showspectrumpic=s=hd480 output84.png
```

- Dùng công cụ feh để xem các ảnh spectrogram đã tạo

```
feh <đường dẫn file ảnh>
```

- Để nhìn rõ sự khác nhau giữa trước và sau giấu tin, sinh viên chạy script sau:

```
python3 compare_spectrogram.py -i1 <file spectrogram audio gốc> -i2 <file spectrogram audio giấu tin> -o <tên file output>
```

- Cụ thể:

```
python3 compare_spectrogram.py -i1 ~/output.png -i2 ~/output8.png -o compare8.png
```

```
python3 compare_spectrogram.py -i1 ~/output.png -i2 ~/output83.png -o compare83.png
```

```
python3 compare_spectrogram.py -i1 ~/output.png -i2 ~/output84.png -o compare84.png
```


- Dùng công cụ feh để xem các ảnh spectrogram đã tạo
feh <đường dẫn file ảnh>
- Nhìn vào ảnh so sánh, màu càng đậm chứng tỏ sự thay đổi giữa file audio trước và sau giấu tin càng lớn.
- Kiểm tra thử các bit LSB tìm sự bất thường, ở đây dùng script kiểm tra 3 bit LSB
python3 test_detect3.py ~/miniTalk.wav
python3 test_detect3.py ~/lsb8/output8.wav
python3 test_detect3.py ~/lsb83/output83.wav
python3 test_detect3.py ~/lsb84/output84.wav
- Kiểm tra kết quả bằng lệnh :
checkwork
- Kết thúc bài lab:
Stoplabb
- 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 task tạo file audio giấu tin

	Result Tag	Container	File	Field Type	Field ID	Timestamp Type
4	_steg1	compare_lsb	np/python3_output.log	CONTAINS	Encoding successful!	File
5	_steg2	compare_lsb	np/python3_output.log	CONTAINS	MSB: 66 (Character 'B')	File
6	_steg3	compare_lsb	np/python3_output.log	CONTAINS	MSB: 68 (Character 'D')	File
7	_steg4	compare_lsb	np/python3_output.log	CONTAINS	MSB: 67 (Character 'C')	File
8	_steg5	compare_lsb	np/python3_output.log	CONTAINS	Successfully.	File

	Goal Type	Boolean	Boolean Result Tags
1	boolean	(_steg1 and _steg2 and _steg3 and _steg4 and _steg5)	create_spec

Hình 4 Checkwork giấu tin

- Hệ thống sẽ truy vết task tạo ảnh spectrogram cho các file audio được tạo

	Result Tag	Container	File	Field Type	Field ID	Timestamp Type
1	create_spec	compare_lsb	mp/ffmpeg_output.log	CONTAINS	png	File

Hình 5 Checkwork tạo file spectrogram

- Hệ thống sẽ truy vết task so sánh các ảnh spectrogram trước và sau giấu tin.

	Result Tag	Container	File	Field Type	Field ID	Timestamp Type
2	compare_spec	compare_lsb	.bash_history	CONTAINS	compare_spectrogram.py	File

Hình 6 Checkwork so sánh spectrogram trước và sau giấu tin

- Hệ thống truy vết task kiểm tra bit LSB của các file audio

	Result Tag	Container	File	Field Type	Field ID	Timestamp Type
3	check_lsb	compare_lsb	.bash_history	CONTAINS	python3 test_detect3.py	File

Hình 7 Checkwork kiểm tra LSB

TÀI LIỆU THAM KHẢO

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