

**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: Tách tin từ file audio chứa thông điệp giấu bằng LSB84

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
CHƯƠNG 2. NỘI DUNG THỰC HÀNH.....	6
2.1 Chuẩn bị môi trường	6
2.2 Các bước thực hiện	6
2.2.1 Khởi động bài lab.....	6
2.2.2 Thực hiện các nhiệm vụ	6
CHƯƠNG 3. Truy vết thực hành	8
3.1 Checkwork.....	8
TÀI LIỆU THAM KHẢO	9

DANH MỤC CÁC HÌNH VẼ

Hình 1 – Tự động tạo file audio giấu tin khi khởi tạo lab	6
Hình 2 – Tách thông điệp dưới dạng nhị phân.....	7
Hình 3 – Chuyển file binary về thông điệp gốc	7
Hình 4 – Thông điệp gốc được tách	7
Hình 5 - Checkwork tách dãy nhị phân	8
Hình 6 – Checkwork tạo file chứa dãy nhị phân.....	8
Hình 7 – Checkwork dịch từ dãy binary về thông điệp gốc	8
Hình 8 – So sánh thông điệp ban đầu và thông điệp được tách ra	8

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 thực hành giấu tin bằng phương pháp LSB vào audio

Trong bài thực hành này sinh viên sẽ thực hành tách tin từ file audio đã qua xử lý giấu tin bằng phương pháp sử dụng 8 bit MSB và 4 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/stego_audio_lsb84_extract/imodule.tar`

Vào terminal, gõ:

`labtainer -r stego_audio_lsb84_extract`

(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)

2.2.2 Thực hiện các nhiệm vụ

- Khi khởi động lab, nó sẽ tự động chạy file giấu tin với thông điệp message.txt có sẵn trong lab



```
ubuntu@stegoaudiolsb84extract: ~  
Steganography is under processing ...  
[+] Found appropriate MSB: 72 (Byte 72, Char 'H')  
[+] Found appropriate MSB: 73 (Byte 73, Char 'I')  
[+] Found appropriate MSB: 68 (Byte 68, Char 'D')  
[+] Found appropriate MSB: 69 (Byte 69, Char 'E')  
[+] Found appropriate MSB: 32 (Byte 32, Char ' ')  
[+] Found appropriate MSB: 77 (Byte 77, Char 'M')  
[+] Found appropriate MSB: 69 (Byte 69, Char 'E')  
✓ Successfully. Audio file was saved in output.wav  
ubuntu@stegoaudiolsb84extract:~$ ls  
embed84_audio.py  extract84_audio.py  message.txt  miniTalk.wav  output.wav  
ubuntu@stegoaudiolsb84extract:~$
```

Hình 1 – Tự động tạo file audio giấu tin khi khởi tạo lab

- Sinh viên có thể thử nghiệm giấu thông điệp khác vào file audio có sẵn trong bài lab với cấu trúc lệnh.

`python3 embed84_audio.py -f <file chứa thông điệp> -s <file audio> -o <tên file audio output>`

- Sinh viên thực hiện tách file binary từ file audio chứa thông điệp

`python3 extract84_audio.py -s output.wav -b`

```
ubuntu@stegoaudiolsb84extract:~$ python3 extract84_audio.py -s output.wav -b
Message Extracted:
[Binary] 01001000010010010100010001000101001000000100110101000101
```

Hình 2 – Tách thông điệp dưới dạng nhị phân

- Thực hiện dịch từ dãy binary về thông điệp gốc, ghi nội dung binary vào file bin.txt và chạy lệnh sau để chuyển đổi sang thông điệp gốc

python3 bin_to_text.py -f bin.txt -o result.txt

```
ubuntu@stegoaudiolsb84extract:~$ ls
bin.txt          embed84_audio.py  message.txt      output.wav
bin_to_text.py   extract84_audio.py miniTalk.wav
ubuntu@stegoaudiolsb84extract:~$ cat bin.txt
01001000010010010100010001000101001000000100110101000101
ubuntu@stegoaudiolsb84extract:~$ python3 bin_to_text.py -f bin.txt -o result.txt
✓ Converted binary to text. Text file saved in 'result.txt'.
```

Hình 3 – Chuyển file binary về thông điệp gốc

- So sánh thông điệp được tách ra với thông điệp ban đầu xem có trùng khớp không (file audio giấu tin tạo tự động)

```
ubuntu@stegoaudiolsb84extract:~$ cat result.txt
HIDE MEubuntu@stegoaudiolsb84extract:~$
```

Hình 4 – Thông điệp gốc được tách

- Kiểm tra kết quả bằng lệnh :

checkwork

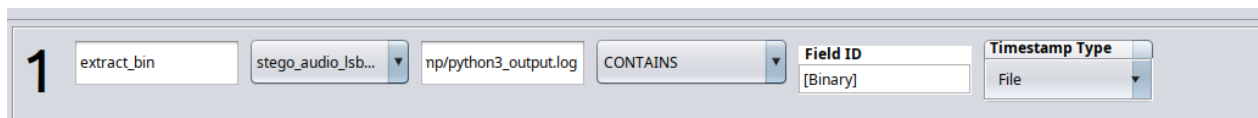
- Kết thúc bài lab:

Stoplalab

CHƯƠNG 3. TRUY VẾT THỰC HÀNH

3.1 Checkwork

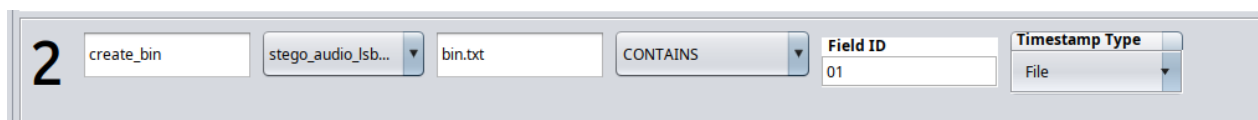
- Hệ thống sẽ kiểm tra xem sinh viên đã tách được dãy binary ra từ audio chưa



1 extract_bin stego_audio_lsb... np/python3_output.log CONTAINS [Binary] File

Hình 5 - Checkwork tách dãy nhị phân

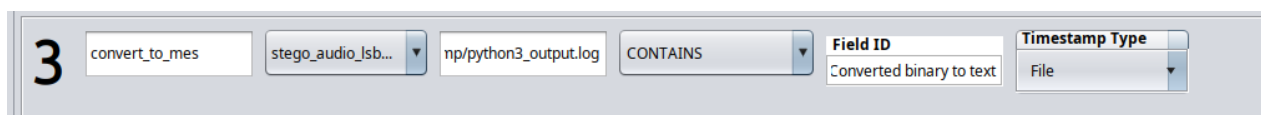
- Hệ thống sẽ truy vết task tạo file bin.txt chứa dãy binary để biên dịch



2 create_bin stego_audio_lsb... bin.txt CONTAINS 01 File

Hình 6 – Checkwork tạo file chứa dãy nhị phân

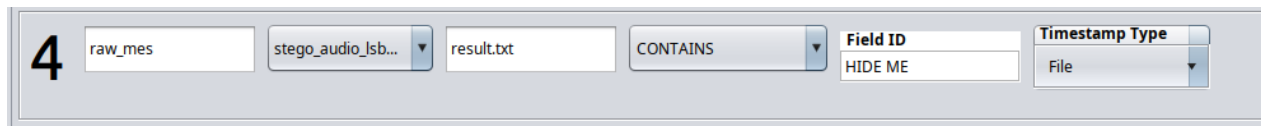
- Hệ thống theo dõi task chuyển đổi dãy binary thành thông điệp gốc



3 convert_to_mes stego_audio_lsb... np/python3_output.log CONTAINS Converted binary to text File

Hình 7 – Checkwork dịch từ dãy binary về thông điệp gốc

- Hệ thống so sánh thông điệp gốc được dịch ra với thông điệp ban đầu



4 raw_mes stego_audio_lsb... result.txt CONTAINS HIDE ME File

Hình 8 – So sánh thông điệp ban đầu và thông điệp được tách ra

TÀI LIỆU THAM KHẢO

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