

Làm quen với thuật toán giấu tin LSB trong âm thanh

Mục đích

Giúp sinh viên làm quen với kỹ thuật giấu tin (steganography) trong dữ liệu âm thanh bằng phương pháp LSB (Least Significant Bit) và thực hành các bước xử lý dữ liệu âm thanh để giấu thông tin.

Yêu cầu đối với sinh viên

Có kiến thức cơ bản về hệ điều hành Linux. Hiểu về xử lý tín hiệu âm thanh và kỹ thuật giấu tin LSB.

Nội dung thực hành

Khởi động bài lab:

Vào terminal, gõ:

```
labtainer -r audiosteg-lsb-code
```

(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)

Sau khi khởi động, terminal ảo sẽ xuất hiện, sẵn sàng cho các bước thực hành.

Bước 1: Tiền xử lý

Chuẩn bị dữ liệu thô (raw_data) để dùng cho bước giấu tin (embedding) hoặc giải mã (extracting).

```
python3 preprocess.py <input.wav> <LSBs>
```

Bước 2: Chuẩn bị thông tin cần giấu

Tạo file secret.txt với nội dung là msv của sinh viên:

```
echo "<MSV>" > secret.txt
```

Biến đổi thông tin cần giấu thành dạng nhị phân.

```
python3 convert_to_bit.py message.txt
```

Bước 3: Giấu tin vào tín hiệu âm thanh

```
python3 hide.py
```

Nhập vào từ bàn phím các thông tin:

- Đường dẫn đến file wav
- Đường dẫn đến file muốn dấu
- Đường dẫn đến file wav mới
- Số LSB sử dụng

Kiểm tra xem file wav đã được tạo thành công chưa:

ls <đường dẫn tới file wav mới>

So sánh chuỗi byte của 2 file wav để quan sát sự thay đổi:

python3 convert_to_byte.py <input.wav>

Bước 4: Tách tin để kiểm tra

python3 recover.py

Nhập vào từ bàn phím các thông tin:

- Đường dẫn đến file wav mới
- Đường dẫn lưu file thông điệp
- Số LSB sử dụng
- Số byte cần trích xuất

Kết thúc bài lab

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

stoplab audiostream-lsb-code

Khi bài lab kết thúc, một tệp zip chứa kết quả sẽ được tạo và lưu tại vị trí hiển thị bên dưới lệnh `stoplab`.

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

labtainer -r audiostream-lsb-code