Bài Thực Hành: Giấu Tin Trong Video HEVC bằng Phương Pháp LSB trong Miền Không Gian

1. Mục đích

Bài thực hành này hướng dẫn sử dụng kỹ thuật giấu tin LSB (Least Significant Bit) trong miền không gian kết hợp với tái cấu trúc video chuẩn HEVC (H.265) để nhúng và trích xuất dữ liệu bí mật. Giúp sinh viên hiểu cách nhúng dữ liệu vào khung hình video, bảo vệ dữ liệu ẩn bằng mã hóa không mất mát, và tái tạo video để duy trì tính toàn vẹn. nhúng và trích xuất thông tin.

2. Yêu cầu đối với sinh viên:

- Có kiến thức cơ bản về lập trình Python.
- Biết cách sử dụng terminal và các lệnh cơ bản trên hệ điều hành Linux.
- Tìm hiểu về FFmpeg và chuẩn nén video HEVC (H.265).
- Hiểu khái niệm giấu tin và kỹ thuật LSB.

3. *Nội dung thực hành*

3.1. Chuẩn bị lab

- Khởi động lab
- Chạy lệnh:

  labtainer -r stego_code_h265_lsb

  Tùy chọn -r đảm bảo môi trường được làm mới (reset) nếu đã chạy trước đó, cung cấp một môi trường sạch để thực hành.

  (Chú ý: Sinh viên sử dụng <TÊN_TÀI_KHOẢN_HỆ_THỐNG> của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm.)

  Môi trường lab được khởi động. Để minh họa nguyên tắc giấu tin trong video HEVC, lab sử dụng một video mẫu và các công cụ như FFmpeg và Python.

3.2. Chuyển đổi video

- Chuyển đổi video đầu vào input.mp4 từ định dạng H.264 sang H.265 (HEVC) thực hiện lệnh sau:

ffmpeg -i input.mp4 -c:v libx265 -c:a copy video.mp4

Tùy chọn -c:v libx265 chỉ định mã hóa video bằng HEVC, trong khi -c:a copy giữ nguyên âm thanh mà không nén lại, tiết kiệm thời gian xử lý. Kết quả là tệp video.mp4 mới ở định dạng HEVC.

3.3. Nhúng dữ liệu bằng LSB

- Tạo thư mục mới:

mkdir temp_frames

Để lưu trữ các khung hình được tách từ video. Đây là nơi chứa các tệp PNG tạm thời trong quá trình xử lý.

- Xác định tốc độ khung hình video mẫu để thực hiện các thao tác sau này:

ffprobe -v error -select_streams v:0 -show_entries stream=r_frame_rate video.mp4

- Tách video video.mp4 thành các khung hình riêng lẻ dưới dạng tệp PNG, lưu vào thư mục temp_frames:

ffmpeg -i video.mp4 -vf "fps=..." -q:v 2 temp_frames/frame_%04d.png

Tùy chọn -vf "fps=…" đặt tốc độ khung hình vừa tìm được. -q:v 2 kiểm soát chất lượng đầu ra (giá trị thấp hơn cho chất lượng cao hơn). %04d đảm bảo tên tệp có 4 chữ số (ví dụ: 0001, 0002).

- Chạy mã Python stego_lsb.py để nhúng dữ liệu bí mật vào khung hình đầu tiên:

python3 stego_lsb.py secret.txt temp_frames/frame_0001.png

Mã này sẽ yêu cầu dữ liệu cần giấu (ví dụ: một chuỗi văn bản), sau đó sửa đổi các bit ít quan trọng nhất (LSB) của giá trị pixel trong ảnh. Kết quả là khung hình đã chỉnh sửa được lưu lại, sẵn sàng để tái tạo video. Kiểm tra khung hình sau khi nhúng để đảm bảo nó vẫn trông bình thường.

3.4. Tái tạo video HEVC

- Tạo video từ khung hình đầu tiên (frame_0001.png) với mã hóa không mất mát:

ffmpeg -loop 1 -framerate 25 -i temp_frames/frame_0001.png -c:v libx265 -x265-params lossless=1 -r 25 -frames:v 1 frame1_lossless.mp4

-loop 1 lặp lại khung hình một lần, -framerate 25 là tốc độ khung hình tìm được ở bước trước, -i temp_frames/frame_0001.png chỉ định tệp đầu vào. -c:v libx265 -x265-params lossless=1 sử dụng HEVC với chế độ không mất mát để bảo vệ dữ liệu ẩn. -r 25 đảm bảo tốc độ khung hình đầu ra, -frames:v 1 giới hạn video chỉ có một khung hình. Kết quả là frame1_lossless.mp4.

- Xác định số khung hình còn lại trong mục temp_frames:

ls temp_frames/

- Tạo video từ các khung hình còn lại (từ frame_0002.png đến frame_2497.png) với mã hóa mất mát:

ffmpeg -framerate 25 -start_number 2 -i temp_frames/frame_%04d.png -c:v libx265 -crf 32 -preset fast -r 25 -frames:v 2496 -movflags +faststart remaining_lossy.mp4

-framerate 25 đặt tốc độ khung hình, -start_number 2 bắt đầu từ khung hình thứ hai, -i temp_frames/frame_%04d.png chỉ định mẫu tên tệp đầu vào. -c:v libx265 -crf 32 -preset fast sử dụng HEVC với chất lượng mất mát CRF 32 là mức chất lượng trung bình. -r 25 giữ tốc độ khung hình, -frames:v 2496 giới hạn 2496 khung hình (tổng số khung - 1). -movflags +faststart tối ưu hóa video để phát trực tuyến. Kết quả là remaining_lossy.mp4.

- Chuyển đổi frame1_lossless.mp4 sang định dạng trung gian MPEG-TS (frame.ts). -c copy sao chép dữ liệu mà không mã hóa lại, -f mpegts chỉ định định dạng đầu ra. MPEG-TS được sử dụng để ghép video dễ dàng hơn:

ffmpeg -i frame1_lossless.mp4 -c copy -f mpegts frame.ts

- Chuyển đổi remaining_lossy.mp4 sang định dạng MPEG-TS (remaining.ts). Tương tự lệnh trên, -c copy giữ nguyên dữ liệu, -f mpegts định dạng đầu ra là MPEG-TS:

ffmpeg -i remaining_lossy.mp4 -c copy -f mpegts remaining.ts

- Ghép hai tệp frame.ts và remaining.ts thành một video hoàn chỉnh video_hidden.mp4:

ffmpeg -i "concat:frame.ts|remaining.ts" -c copy -r 25 -movflags +faststart video_hidden.mp4

-i "concat:frame.ts|remaining.ts" chỉ định ghép nối các tệp, -c copy sao chép dữ liệu mà không mã hóa lại, -r 25 đảm bảo tốc độ khung hình, -movflags +faststart tối ưu hóa video để phát trực tuyến.

- Xem video mới có chạy như video bình thường không:

ffplay video_hidden.mp4

- Kiểm tra dung lượng video so với dung lượng video gốc HEVC có nhỏ hơn không

ls -lh video_hidden.mp4

ls -lh video.mp4

- Sử dụng rm để xóa chỉ giữ lại video input và video_hidden, với thư mục temp_frames thì sử dụng lệnh *rm -r temp_frames/*

3.5. Trích xuất dữ liệu

- Chuyển dữ liệu sang cho bob:

scp video_hidden.mp4 bob:/home/ubuntu/

Password ssh sẽ là: password123, sau khi chuyển xong thực hiện rm để xóa đi các file chỉ giữ lại file input.mp4

- Phía bob tách khung hình đầu tiên từ video_hidden.mp4 và lưu dưới dạng first_frame.png:

ffmpeg -i video_hidden.mp4 -vframes 1 -f image2 first_frame.png

-vframes 1 giới hạn chỉ lấy một khung hình, -f image2 chỉ định định dạng đầu ra là ảnh tĩnh. Đây là khung hình chứa dữ liệu ẩn.

- Chạy mã Python extract_data.py để trích xuất dữ liệu từ first_frame.png:

python3 extract_data.py first_frame.png

Mã này đọc các bit LSB của giá trị pixel và tái tạo dữ liệu bí mật ban đầu (ví dụ: chuỗi văn bản đã nhúng), sau đó hiển thị lên màn hình.

3.6. Xóa dấu vết

- Tạo thư mục mới:

mkdir temp_frames

Để lưu trữ các khung hình được tách từ video_hidden.mp4.

- Tách video video.mp4 thành các khung hình riêng lẻ dưới dạng tệp PNG, lưu vào thư mục temp_frames:

ffmpeg -i video_hidden.mp4 -vf "fps=25" -q:v 2 temp_frames/frame_%04d.png

- Xóa đi frame chứa dữ liệu ẩn:

rm temp_frames/frame_0001.png

- Tái tạo lại thành video thường không còn dữ liệu ẩn:

ffmpeg -framerate 25 -i temp_frames/frame_%04d.png -i video_hidden.mp4 -c:v libx265 -preset fast -c:a copy -map 0:v -r 25 video.mp4

- Xóa các mục chỉ để lại video thường vừa đựa tái tạo

Sau khi thực hiện xong phía bob chỉ còn lại video thường.

4. *Kết quả cần đạt được*

- Chạy được tất cả các bước như yêu cầu.
- Cần nộp 1 file: trong thư mục: /home/student/labtainer_xfer/TÊN_BÀI_LAB (tên tài khoản.*TÊN_BÀI_LAB.lab)*
- Kết thúc bài lab:
    - Trên terminal đầu tiên sử dụng câu lênh sau để kết thúc bài lab:

stoplab stego_code_h265_lsb

- Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.
- Sinh viên cần nộp file *.lab* để chấm điểm.
- Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh: *checkwork <tên bài thực hành>*
- Khởi động lại bài lab: Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

labtainer –r stego_code_h265_lsb