

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



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

**NHÓM LỚP: AT02
TÊN BÀI: stego-video-sei-hide**

Sinh viên thực hiện:

B21DCAT034 - Nguyễn Kiều Tuấn Anh

Giảng viên: Đỗ Xuân Chợt

Tháng 5/2025- Hà Nội

MỤC LỤC

MỤC LỤC	2
CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI LAB	3
1.1 Mục đích	3
1.2 Tìm hiểu lý thuyết	3
1.3 Đẩy bài lab lên Docker và Github	4
CHƯƠNG 2. NỘI DUNG THỰC HÀNH	5
2.1 Chuẩn bị môi trường	5
2.2 Nội dung thực hành	5
Task 1: Xem thông tin video gốc bằng câu lệnh	5
Task 2: Chuyển đổi sang định dạng H.264 bằng câu lệnh	5
Task 3: Chèn thông điệp vào SEI NAL units	6
Task 4: Ghép audio từ video gốc và xuất video mới bằng cách gõ các câu lệnh	6
Task 5 : So sánh kích thước và đánh giá chất lượng video	7
CHƯƠNG 3. KẾT QUẢ THỰC HÀNH	9
TÀI LIỆU THAM KHẢO	10

CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI LAB

1.1 Mục đích

Bài Lab nhằm giúp sinh viên hiểu và thực hành kỹ thuật giấu tin trong video sử dụng chuẩn nén H.264. Cụ thể, bài lab tập trung vào việc:

- Sử dụng các đơn vị dữ liệu bổ sung SEI (Supplemental Enhancement Information) trong luồng video H.264 để nhúng thông điệp bí mật.
- Thực hành các công cụ như ffmpeg, ffprobe và script Python để thao tác trực tiếp với luồng video và đánh giá chất lượng video sau khi giấu tin để đảm bảo rằng sự thay đổi không gây suy giảm đáng kể chất lượng.
- Hiểu quy trình xử lý, ghép lại video (repackaging) sau khi nhúng dữ liệu ẩn.

1.2 Tìm hiểu lý thuyết

Chuẩn nén video H.264

- Trong H.264, video được mã hóa thành một chuỗi bitstream, là dòng dữ liệu nhị phân tuần tự có cấu trúc rõ ràng. Bitstream này bao gồm nhiều Access Units (AU) – mỗi AU tương ứng với một frame (khung hình) hoặc field của video.
- Mỗi Access Unit (AU) bao gồm một hoặc nhiều NAL Units (Network Abstraction Layer Units). Các NAL Units là đơn vị cơ bản chứa dữ liệu như: Slice data (dữ liệu hình ảnh nén), Parameter sets (cấu hình giải mã), SEI (Supplemental Enhancement Information) – thông tin bổ sung (vị trí giấu tin trong bài lab), và các loại NAL khác.

SEI – Supplemental Enhancement Information

- SEI là một loại NAL unit đặc biệt dùng để truyền thông tin phụ trợ cho quá trình giải mã video, chẳng hạn như thông tin hiển thị, thời gian hoặc metadata. Vì SEI không ảnh hưởng trực tiếp đến việc hiển thị hình ảnh, nó trở thành vị trí lý tưởng để giấu dữ liệu ẩn mà không làm thay đổi nội dung hiển thị.

Kỹ thuật giấu tin sử dụng SEI

- Kỹ thuật giấu tin trong SEI sử dụng các NAL unit loại SEI để nhúng dữ liệu, như chuỗi ký tự hoặc mã hóa nhị phân. Trong bài lab này, thông điệp được chèn vào các SEI NAL units thông qua một script Python, sau đó video được tái ghép với luồng âm thanh để tạo thành file MP4 hoàn chỉnh.

Đánh giá chất lượng video sau khi giấu tin

Để đảm bảo rằng việc giấu tin không làm giảm đáng kể chất lượng video, hai chỉ số được sử dụng:

- PSNR đo độ lệch trung bình giữa video gốc và video đã giấu tin, giá trị càng cao thì độ khác biệt càng nhỏ, từ đó cho thấy chất lượng video gần giống với bản gốc. Thông thường, PSNR lớn hơn 30 dB được xem là đạt chất lượng tốt.
- SSIM đánh giá mức độ tương đồng về cấu trúc, độ sáng và độ tương phản giữa hai khung hình tương ứng. SSIM có giá trị từ 0 đến 1, càng gần 1 thì video càng giống bản gốc.

1.3 Đẩy bài lab lên Docker và Github

- Docker (tham khảo các câu lệnh sau)

```
cd stego-video-sei-hide
cd dockerfiles/
nano Dockerfile.stego-video-sei-hide.stego-video-sei-hide.student
nano start.config
cd distrib/
docker login
./publish.py -d -l stego-video-sei-hide
```

- Github (tham khảo các câu lệnh sau - Đổi tên lab cho phù hợp)

```
cd labs
tar cvf stego-video-sei-hide.tar stego-video-sei-hide
git init
git add stego-video-sei-hide
git commit stego-video-sei-hide -m "Adding an IModule"
git add stego-video-sei-hide.tar
git commit -m "update stego-video-sei-hide.tar"
ssh-keygen -t ed25519 -C "your_email@example.com" (Điền mail của tài khoản git)
cat /home/student/.ssh/id_ed25519.pub (Copy SSH key -> Vào setting của git để dán key vào)
```

```
git remote add origin git@github.com:ngkiutunah/stego-video-sei-hide.git
git remote set-url origin git@github.com:ngkiutunah/stego-video-sei-extract.git ( nếu đã đẩy trước đó)
git push --set-upstream origin main -f
```

```
git rm -r -f stego-video-sei-hide ( trường hợp lab cũ bị đẩy theo)
git rm -f stego-video-sei-hide.tar
git commit -m "Remove unnecessary module and its tar"
git push ( 4 câu lệnh này dùng trường hợp lab cũ bị đẩy theo)
```

CHƯƠNG 2. NỘI DUNG THỰC HÀNH

2.1 Chuẩn bị môi trường

Tải lab dùng câu lệnh

imodule <https://github.com/ngkiutunah/stego-video-sei-hide/raw/refs/heads/main/stego-video-sei-hide.tar>

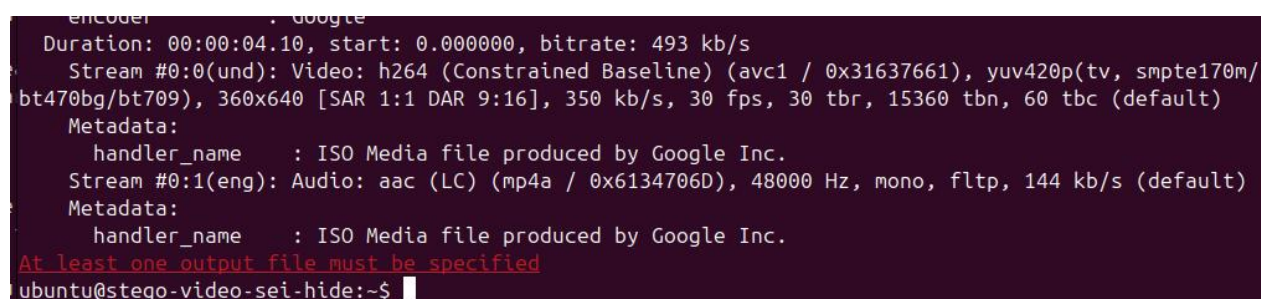
Khởi động lab:

labtainer -r stego-video-sei-hide

2.2 Nội dung thực hành

Task 1: Xem thông tin video gốc bằng câu lệnh

ffmpeg -i kitten.mp4

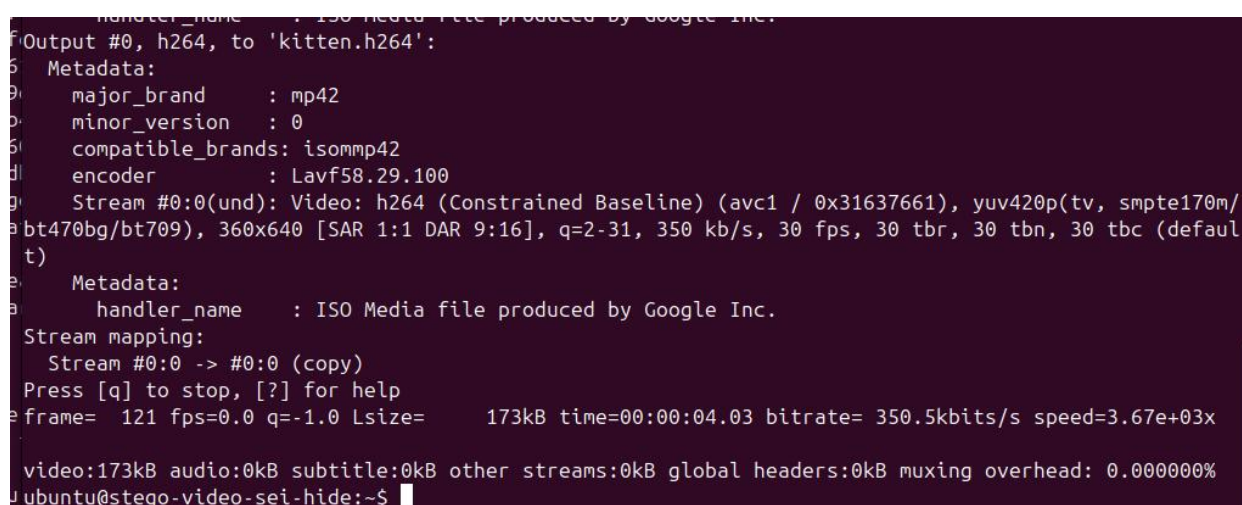


```
Duration: 00:00:04.10, start: 0.000000, bitrate: 493 kb/s
Stream #0:0(und): Video: h264 (Constrained Baseline) (avc1 / 0x31637661), yuv420p(tv, smpte170m/
bt470bg/bt709), 360x640 [SAR 1:1 DAR 9:16], 350 kb/s, 30 fps, 30 tbr, 15360 tbn, 60 tbc (default)
Metadata:
  handler_name      : ISO Media file produced by Google Inc.
Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, mono, fltp, 144 kb/s (default)
Metadata:
  handler_name      : ISO Media file produced by Google Inc.
At least one output file must be specified
ubuntu@stego-video-sei-hide:~$
```

Hình 1: Kiểm tra metadata (frame rate, codec, audio stream...) của video gốc

Task 2: Chuyển đổi sang định dạng H.264 bằng câu lệnh

ffmpeg -i kitten.mp4 -c:v copy -bsf:v h264_mp4toannexb kitten.h264



```
Output #0, h264, to 'kitten.h264':
  Metadata:
    major_brand      : mp42
    minor_version    : 0
    compatible_brands: isommp42
    encoder          : Lavf58.29.100
Stream #0:0(und): Video: h264 (Constrained Baseline) (avc1 / 0x31637661), yuv420p(tv, smpte170m/
bt470bg/bt709), 360x640 [SAR 1:1 DAR 9:16], q=2-31, 350 kb/s, 30 fps, 30 tbr, 30 tbn, 30 tbc (default)
Metadata:
  handler_name      : ISO Media file produced by Google Inc.
Stream mapping:
  Stream #0:0 -> #0:0 (copy)
Press [q] to stop, [?] for help
frame= 121 fps=0.0 q=-1.0 Lsize=      173kB time=00:00:04.03 bitrate= 350.5kbits/s speed=3.67e+03x
video:173kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.000000%
ubuntu@stego-video-sei-hide:~$
```

Hình 2: Tạo file raw video H.264 (kitten.h264) từ video MP4 gốc. Bắt buộc nếu bạn muốn thao tác trực tiếp với các NAL units (như SEI).

Task 3: Chèn thông điệp vào SEI NAL units

python3 inject_sei.py

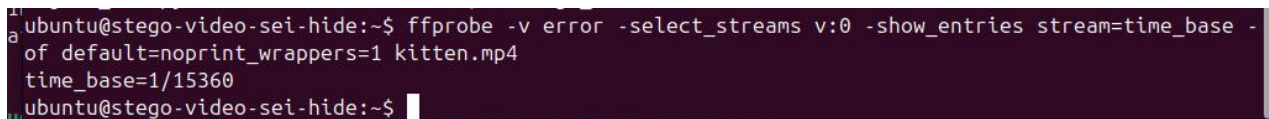


```
ubuntu@stego-video-sei-hide:~$ python3 inject_sei.py
✓ Đã chèn SEI vào stego_kitten.h264
ubuntu@stego-video-sei-hide:~$ ls
inject_sei.py  kitten.h264  kitten.mp4  stego_kitten.h264
ubuntu@stego-video-sei-hide:~$
```

Hình 3: Chạy script Python để chèn một thông điệp (text, shellcode, QR...) vào SEI của file kitten.h264. Kết quả là stego_kitten.h264

Task 4: Ghép audio từ video gốc và xuất video mới bằng cách gõ các câu lệnh

ffprobe -v error -select_streams v:0 -show_entries stream=time_base -of default=noprint_wrappers=1 kitten.mp4



```
ubuntu@stego-video-sei-hide:~$ ffprobe -v error -select_streams v:0 -show_entries stream=time_base -of default=noprint_wrappers=1 kitten.mp4
time_base=1/15360
ubuntu@stego-video-sei-hide:~$
```

Hình 4: Lấy thông tin time_base (1/15360) dùng để giữ cho video tái mã hóa giữ được timing giống với video gốc

Tiếp tục thực hiện câu lệnh

ffmpeg -r 30 -i stego_kitten.h264 -i kitten.mp4 -c:v copy -c:a copy -map 0:v:0 -map 1:a:0 -movflags +faststart -video_track_timescale 15360 stego_kitten.mp4

```

Input #1, mov,mp4,m4a,3gp,3g2,mj2, from 'kitten.mp4':
  Metadata:
    major_brand      : mp42
    minor_version    : 0
    compatible_brands: isommp42
    encoder          : Google
  Duration: 00:00:04.10, start: 0.000000, bitrate: 493 kb/s
  Stream #1:0(und): Video: h264 (Constrained Baseline) (avc1 / 0x31637661), yuv420p(tv, smpte170m/bt470bg/bt709)
  360x640 [SAR 1:1 DAR 9:16], 350 kb/s, 30 fps, 30 tbr, 15360 tbn, 60 tbc (default)
  Metadata:
    handler_name     : ISO Media file produced by Google Inc.
  Stream #1:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, mono, fltp, 144 kb/s (default)
  Metadata:
    handler_name     : ISO Media file produced by Google Inc.
Output #0, mp4, to 'stego_kitten.mp4':
  Metadata:
    encoder          : Lavf58.29.100
  Stream #0:0: Video: h264 (Constrained Baseline) (avc1 / 0x31637661), yuv420p(tv, smpte170m/bt470bg/bt709, progressive), 360x640 [SAR 1:1 DAR 9:16], q=2-31, 30 fps, 30 tbr, 15360 tbn, 30 tbc
  Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, mono, fltp, 144 kb/s (default)
  Metadata:
    handler_name     : ISO Media file produced by Google Inc.
Stream mapping:
  Stream #0:0 -> #0:0 (copy)

```

Hình 5: Tạo video MP4 có thông điệp giấu, giữ nguyên video (copy) và audio gốc. Sử dụng +faststart giúp phát trực tiếp tốt hơn (progressive download).

Task 5 : So sánh kích thước và đánh giá chất lượng video

ls -lh kitten.mp4 stego_kitten.mp4 > size.txt

```

ubuntu@stego-video-sei-hide:~$ ls -lh kitten.mp4 stego_kitten.mp4 > size.txt
ubuntu@stego-video-sei-hide:~$ cat size.txt
-rw-rw-r-- 1 ubuntu ubuntu 247K Apr 26 04:36 kitten.mp4
-rw-rw-r-- 1 ubuntu ubuntu 250K May  8 09:57 stego_kitten.mp4

```

Hình 6: So sánh size trước và sau khi giấu

ffmpeg -i kitten.mp4 -i stego_kitten.mp4 -lavfi psnr="stats_file=psnr_log.txt" -f null -

```

encoder          : Lavc58.54.100 pcm_s16le
frame= 121 fps=0.0 q=-0.0 lsize=N/A time=00:00:04.09 bitrate=N/A speed=19.9x
video:63kB audio:384kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
[Parsed_psnr_0 @ 0x60051cc56580] PSNR y:inf u:inf v:inf average:inf min:inf max:inf

```

Hình 7: PSNR (Peak Signal-to-Noise Ratio): đo mức độ suy hao tín hiệu

ffmpeg -i kitten.mp4 -i stego_kitten.mp4 -lavfi ssim="stats_file=ssim_log.txt" -f null -

```

video:63kB audio:384kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
[Parsed_ssim_0 @ 0x57a1f1078040] SSIM Y:1.000000 (inf) U:1.000000 (inf) V:1.000000 (inf) All:1.000000 (inf)
ubuntu@stego-video-sei-hide:~$

```

Hình 8: SSIM (Structural Similarity Index): đo sự tương đồng cấu trúc.

Kết thúc bài lab:

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

stoplab

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

CHƯƠNG 3. KẾT QUẢ THỰC HÀNH

Màn hình checkwork bài thực hành:

```
student@LabtainerVMware:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/stego-video-sei-hide
Successfully copied 647kB to stego-video-sei-hide-igrader:/home/instructor/B21DCAT034.stego-video-sei-hide.lab
Successfully copied 2.05kB to /home/student/labtainer_xfer/stego-video-sei-hide
Labname stego-video-sei-hide

Student | file_h264 | inject | size | PSNR | SSIM |
=====|=====|=====|=====|=====|=====|
B21DCAT034 | Y | Y | Y | Y | Y |
What is automatically assessed for this lab:
```

Hình 19:Màn hình checkwork

TÀI LIỆU THAM KHẢO

- [1] Labdesinger available at <https://nps.edu/documents/107523844/117289221/labdesigner.pdf/9ab93fc3-7cfe-48f4-987b-2efda8d99346?t=1583340431133>