

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



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

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

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 .....	3
CHƯƠNG 2. NỘI DUNG THỰC HÀNH .....	5
2.1 Chuẩn bị môi trường .....	5
2.2 Nội dung thực hành .....	5
Task 1: Kiểm tra video đầu vào bằng câu lệnh .....	5
Task 2: Chuyển đổi sang định dạng .h264 bằng câu lệnh .....	5
Task 3: Trích xuất và kiểm tra thông điệp được trích xuất bằng các câu lệnh .....	6
Task 4: Thử nghiệm re-encode lại video để xem tính toàn vẹn của tin .....	6
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 thực hiện trích xuất thông điệp đã được giấu trong video theo chuẩn H.264, thông qua việc phân tích và đọc các gói dữ liệu bổ sung SEI (Supplemental Enhancement Information) trong bitstream của video. Đây là bước tiếp theo sau quá trình giấu tin đã được thực hiện ở Lab 1, giúp kiểm chứng khả năng phục hồi thông điệp và đảm bảo tính đúng đắn của kỹ thuật giấu tin.

### 1.2 Tìm hiểu lý thuyết

Trong chuẩn nén video H.264, luồng dữ liệu (bitstream) bao gồm các Access Units, mỗi đơn vị này lại chứa nhiều NAL Units. Trong số đó, NAL Unit loại SEI được dùng để truyền các thông tin bổ sung, không ảnh hưởng đến quá trình giải mã hình ảnh. SEI là nơi lý tưởng để nhúng và sau đó trích xuất các dữ liệu ẩn, nhờ tính linh hoạt và không làm thay đổi nội dung chính của video. Trong bài lab này, đoạn mã Python sẽ thực hiện việc quét và đọc dữ liệu từ các SEI NAL Unit trong video đã giấu tin để thu được nội dung thông điệp.

Quá trình trích xuất yêu cầu chuyển đổi video từ định dạng .mp4 sang .h264 (Annex B format), cho phép phân tích trực tiếp các gói dữ liệu NAL. Sau khi xử lý, nội dung được ghi lại trong các tệp văn bản, từ đó có thể kiểm chứng tính chính xác của thông điệp đã được nhúng trước đó.

Tuy nhiên, cần lưu ý rằng nếu thực hiện thao tác nén lại video bằng lệnh mã hóa (re-encoding), chẳng hạn như dùng ffmpeg với codec libx264, thông tin nhúng trong SEI có thể bị mất hoàn toàn. Điều này xảy ra vì quá trình tái mã hóa có thể bỏ qua các SEI không quan trọng, hoặc tạo lại cấu trúc NAL Units mới, không giữ lại dữ liệu giấu. Do đó, việc bảo toàn thông tin nhúng phụ thuộc vào việc duy trì nguyên vẹn luồng bitstream ban đầu, đặc biệt là khi sử dụng kỹ thuật giấu tin dựa trên SEI.

### 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

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

#### Khởi động lab:

labtainer -r stego-video-sei-extract

### 2.2 Nội dung thực hành

Task 1: Kiểm tra video đầu vào bằng câu lệnh

*ffmpeg -i stego\_kitten.mp4*



```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'stego_kitten.mp4':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf60.16.100
  Duration: 00:00:04.10, start: 0.000000, bitrate: 498 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     : VideoHandler
  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
```

Hình 1: Kiểm tra thông tin định dạng video như codec, độ phân giải, số lượng luồng, tần số khung hình,...

Task 2: Chuyển đổi sang định dạng .h264 bằng câu lệnh

*ffmpeg -i stego\_kitten.mp4 -c:v copy -bsf:v h264\_mp4toannexb -an stego\_kitten.h264*

```

Output #0, h264, to 'stego_kitten.h264':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    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     : VideoHandler
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=2.97e+03x
video:173kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.000000%
ubuntu@stego-video-sei-extract:~$

```

Hình 2: Chuyển video từ định dạng .mp4 (container format) sang định dạng Annex B (.h264) để có thể truy cập trực tiếp bitstream và các NAL Units, trong đó có chứa SEI.

Task 3: Trích xuất và kiểm tra thông điệp được trích xuất bằng các câu lệnh

***python3 extract\_sei.py***

Script Python này sẽ đọc video .h264, tìm kiếm các NAL Unit có loại SEI, giải mã nội dung giấu bên trong và ghi ra file văn bản.

***cat message.txt check.txt***

```

ubuntu@stego-video-sei-extract:~$ python3 extract_sei.py
he ubuntu@stego-video-sei-extract:~$ cat message.txt check.txt
Kin Cha Na Sa rang he
Kin Cha Na Sa rang he
ou ubuntu@stego-video-sei-extract:~$

```

Hình 3: Hiện thị nội dung thông điệp đã được giấu trong video. message.txt chứa thông điệp chính, còn check.txt có thể chứa thông tin kiểm tra để so sánh

Task 4: Thử nghiệm re-encode lại video để xem tính toàn vẹn của tin

***ffmpeg -i stego\_kitten.h264 -c:v libx264 -preset fast -crf 22 -an -video\_track\_timescale 15360 stego\_kitten.mp4***

```

cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame= 121 fps= 98 q=-1.0 Lsize=      236kB time=00:00:03.93 bitrate= 490.6kb/s speed=3.19x
video:233kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.957945%
[libx264 @ 0x56f096fb0740] frame I:1      Avg QP:22.91  size: 16362
[libx264 @ 0x56f096fb0740] frame P:33    Avg QP:23.04  size:  4981
[libx264 @ 0x56f096fb0740] frame B:87    Avg QP:28.08  size:   661
[libx264 @ 0x56f096fb0740] consecutive B-frames:  2.5%  3.3%  5.0% 89.3%
[libx264 @ 0x56f096fb0740] mb I  I16..4:  8.7% 35.5% 55.8%
[libx264 @ 0x56f096fb0740] mb P  I16..4:  0.9%  1.9%  1.3% P16..4: 44.4% 25.4%  9.3%  0.0%  0.0%
skip:16.7%
[libx264 @ 0x56f096fb0740] mb B  I16..4:  0.1%  0.2%  0.1% B16..8: 29.0%  4.0%  0.2% direct: 3.6%
skip:62.8% L0:49.5% L1:45.8% BI: 4.7%
[libx264 @ 0x56f096fb0740] 8x8 transform intra:42.3% inter:60.6%
[libx264 @ 0x56f096fb0740] coded y,uvDC,uvAC intra: 68.3% 74.3% 22.8% inter:  8.9% 11.6%  0.4%
[libx264 @ 0x56f096fb0740] i16 v,h,dc,p: 36% 30% 13% 21%
[libx264 @ 0x56f096fb0740] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 24% 32% 13%  3%  4%  6%  5%  5%  7%
[libx264 @ 0x56f096fb0740] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 26% 28%  8%  5%  6%  7%  7%  5%  7%
[libx264 @ 0x56f096fb0740] i8c dc,h,v,p: 49% 23% 23%  5%
[libx264 @ 0x56f096fb0740] Weighted P-Frames: Y:3.0% UV:3.0%
[libx264 @ 0x56f096fb0740] ref P L0: 76.1% 23.9%
[libx264 @ 0x56f096fb0740] ref B L0: 90.8%  9.2%
[libx264 @ 0x56f096fb0740] ref B L1: 98.5%  1.5%
[libx264 @ 0x56f096fb0740] kb/s:472.58
ubuntu@stego-video-sei-extract:~$

```

Hình 5: Mã hóa lại video từ file .h264 sang định dạng .mp4 mới. Quá trình này sử dụng codec libx264 và thiết lập mức nén CRF=22. Điều này làm mất hoặc thay đổi các SEI NAL Units, dẫn đến mất dữ liệu giấu.

Lập lại quy trình chuyển đổi và trích xuất để kiểm tra xem sau khi nén lại, dữ liệu giấu có còn tồn tại không. Thông thường, nếu quá trình mã hóa lại bỏ qua SEI, sẽ không trích xuất được gì. tức là file message.txt rỗng. Thực hiện các câu lệnh sau:

***ffmpeg -i stego\_kitten.mp4 -c:v copy -bsf:v h264\_mp4toannexb -an stego\_kitten.h264***

```

handler_name      : VideoHandler
File 'stego_kitten.h264' already exists. Overwrite ? [y/N] y
Output #0, h264, to 'stego_kitten.h264':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf58.29.100
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 360x640 [SAR 1:1 DAR 9:16], c
=2-31, 473 kb/s, 30 fps, 30 tbr, 30 tbn, 30 tbc (default)
  Metadata:
    handler_name     : VideoHandler
Stream mapping:
  Stream #0:0 -> #0:0 (copy)
Press [q] to stop, [?] for help
frame= 121 fps=0.0 q=-1.0 Lsize=      233kB time=00:00:03.96 bitrate= 482.0kb/s speed=1.57e+03x
video:233kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.000000%
ubuntu@stego-video-sei-extract:~$

```

Hình 6: Chuyển về file video giấu tin đã re-encode lại về dạng h264 thuần

Tiếp tục thực hiện các câu lệnh

***python3 extract\_sei.py***

***cat message.txt check.txt***



```
ubuntu@stego-video-sei-extract:~$ python3 extract_sei.py
ubuntu@stego-video-sei-extract:~$ cat message.txt check.txt
Kin Cha Na Sa rang he
ubuntu@stego-video-sei-extract:~$
```

Hình 7: Thông điệp giấu trong file message.txt đã bị mất

### **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:

Lưu ý: checkwork đủ các mục proxy\_settings | log\_configurati | black\_lists | white\_lists là đạt.

```
student@LabtainerVMware:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/stego-video-sei-extract
Successfully copied 525kB to stego-video-sei-extract-igrader:/home/instructor/B21DCAT034.stego-video-sei-extract.lab
Successfully copied 2.05kB to /home/student/labtainer_xfer/stego-video-sei-extract
Labname stego-video-sei-extract

Student          |          h264_file |          extract | compare_message |          re-encode |
=====|=====|=====|=====|=====|
B21DCAT034       |          Y         |          Y       |          Y       |          Y         |
What is automatically assessed for this lab:
```

Hình 8: 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>