

Bài Lab: Giấu và tách tin trong ảnh màu bằng phương pháp hoán vị giả ngẫu nhiên

1. Mục đích

- Mục tiêu của bài lab này là giúp sinh viên làm quen với việc giấu và tách tin trong ảnh màu bằng phương pháp hoán vị giả ngẫu nhiên. Sau khi hoàn thành bài lab, sinh viên sẽ hiểu sâu hơn về các khái niệm liên quan đến ảnh, các bước giấu và tách một văn bản trong ảnh bằng phương pháp này và có thể viết chương trình để giấu và tách một thông điệp cho trước bằng phương pháp này.

2. Yêu cầu đối với sinh viên

- Sinh viên đã thực hiện bài lab stego-image-basic-code-hvgnn1 và stego-image-basic-code-hvgnn2.
- Sinh viên nắm được kiến thức về ngôn ngữ Python và hệ điều hành Linux.
- Sinh viên nắm được những kiến thức cơ bản về giấu và tách tin trong ảnh và phương pháp hoán vị giả ngẫu nhiên, bộ sinh số giả ngẫu nhiên (PRNG), thuật toán Blum-Blum-Shub.

3. Tải bài lab và file hướng dẫn thực hành

- Tải thư mục chứa file bài lab và file hướng dẫn thực hành về bằng câu lệnh sau:

```
git clone https://github.com/vuducmanh2003/stego-image-basic-code-hvgnn3
```

- Chuyển đường dẫn đến thư mục:

```
cd ~/labtainer/labtainer-student
```

- Giải nén và chuyển file bài lab vào thư mục /home/trunk/labs bằng câu lệnh sau:

```
imodule file:///home/student/<đường dẫn chứa thư mục vừa tải về>/stego-image-basic-code-hvgnn3/imodule.tar
```

- Giả sử thư mục Downloads chứa thư mục vừa tải về:

```
imodule file:///home/student/Downloads/stego-image-basic-code-hvgnn3/imodule.tar
```

4. Nội dung thực hành

- Trong terminal của Labainer, chạy lệnh sau để khởi động bài lab:

```
labtainer stego-image-basic-code-hvgnn3
```

(chú ý: sinh viên sử dụng **mã sinh viên** của mình để nhập thông tin người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm).

- Sau khi khởi động bài lab thành công, có 2 terminal *alice* và *bob* hiện ra. Hãy xem các file có trong đường dẫn hiện tại bằng câu lệnh *ls*. Trên terminal *alice* có 5 file .py và 1 file .png. Trên terminal *bob* có 2 file .py.

- Hãy xem IP 2 terminal *alice* và *bob* bằng câu lệnh *ifconfig*.

- Trong bài lab này, sinh viên sẽ thực hiện giấu và tách thông điệp “LovePTIT” vào ảnh màu đã cho sẵn. Sau đây là những nhiệm vụ của bài lab:

4.1. Nhiệm vụ 1: Chuyển thông điệp cần giấu sang dạng nhị phân

- Mọi thông điệp khi giấu vào ảnh đều phải ở dạng nhị phân. File *text_to_binary.py* dùng để chuyển văn bản sang nhị phân nhưng nội dung của file còn thiếu. Hãy đọc file để hiểu nội dung và hoàn thành đoạn mã.

- Thực hiện chuyển thông điệp cần giấu sang dạng nhị phân:

```
python3 text_to_binary.py
```

4.2. Nhiệm vụ 2: Trích xuất ma trận điểm ảnh dạng thập phân của ảnh màu

- Xem ảnh màu ban đầu:

```
fm image.png
```

- Thông điệp sẽ được giấu vào các bit trong ma trận điểm ảnh. File *extract_image_matrix.py* dùng để trích xuất ma trận điểm ảnh dạng thập phân của một ảnh. Hãy đọc file để hiểu nội dung đoạn mã.

- Thực hiện trích xuất ma trận điểm ảnh dạng thập phân của ảnh màu:

```
python3 extract_image_matrix.py
```

- Đọc nội dung file kết quả:

```
nano decimal_matrix.txt
```

4.3. Nhiệm vụ 3: Sinh ngẫu nhiên các vị trí trong chuỗi nhị phân của ảnh màu

- Trong phương pháp hoán vị giả ngẫu nhiên, từng bit của thông điệp sẽ được giấu vào các vị trí trong chuỗi nhị phân của ảnh. Các vị trí này được sinh ra bằng thuật toán sinh số giả ngẫu nhiên (trong bài lab này sinh viên thực hiện sinh số giả ngẫu nhiên với thuật toán Blum-Blum-Shub). Số lượng các vị trí cần sinh ra chính bằng độ dài của thông điệp dạng nhị phân. File *blum_blum_shub* dùng để sinh ngẫu nhiên các vị trí trong chuỗi nhị phân của ảnh màu nhưng nội dung file còn thiếu. Hãy đọc file để hiểu nội dung và hoàn thành đoạn mã.

- Thực hiện sinh ngẫu nhiên các vị trí trong chuỗi nhị phân của ảnh màu:

python3 blum_blum_shub.py

4.4. Nhiệm vụ 4: Giấu thông điệp dạng nhị phân vào ảnh màu

- File `hide_message.py` dùng để giấu từng bit của thông điệp dạng nhị phân vào các vị trí trong chuỗi nhị phân của ảnh màu nhưng nội dung file còn thiếu. Hãy đọc file để hiểu nội dung và hoàn thành đoạn mã.
- Thực hiện giấu từng bit của thông điệp dạng nhị phân vào các vị trí trong chuỗi nhị phân của ảnh màu:

python3 hide_message.py

- Xem ảnh màu sau khi giấu thông điệp thành công (sinh viên nên xem lại ảnh màu ban đầu `gray_image.png` xem có nhận biết sự khác nhau bằng mắt thường được không):

fm stegano_image.png

- Sinh viên có thể trích xuất ma trận điểm ảnh dạng thập phân và nhị phân của ảnh màu sau khi đã giấu thông điệp thành công (`stegano_image.png`) và so sánh với ma trận điểm ảnh dạng thập phân và nhị phân của ảnh màu ban đầu (`image.png`) xem những điểm ảnh nào có sự khác nhau.

4.5. Nhiệm vụ 5: Đánh giá chất lượng ảnh màu sau khi đã giấu thông điệp so với ảnh màu ban đầu

- Ảnh màu sau khi giấu tin và trước khi giấu tin có sự thay đổi trong ma trận điểm ảnh. File `assess_stegano_image_quality.py` dùng để đánh giá chất lượng ảnh màu sau khi đã giấu thông điệp so với ảnh màu ban đầu bằng phương pháp tính toán PSNR (tỉ lệ tín hiệu đỉnh trên nhiễu). Kết quả của phương pháp tính toán này như sau:

- **> 40 dB:** chất lượng gần như không khác biệt.
- **30–40 dB:** chất lượng tốt.
- **< 30 dB:** có thể thấy rõ sự suy giảm.

- Thực hiện đánh giá chất lượng ảnh màu `stegano_image.png` so với ảnh màu `image.png`:

python3 assess_stegano_image_quality.py

4.6. Nhiệm vụ 6: Alice chuyển file ảnh chứa thông điệp đã giấu và file thuật toán sinh số giả ngẫu nhiên sang cho Bob

- Trong phương pháp hoán vị giả ngẫu nhiên, người gửi giấu thông điệp vào ảnh và gửi ảnh đã giấu thông điệp qua kênh truyền cho người nhận. Người gửi và người nhận phải

thông nhất với nhau về thuật toán sinh số giả ngẫu nhiên để có thể thực hiện giấu và tách tin được.

- Alice thực hiện chuyển file ảnh chứa thông điệp đã giấu và file thuật toán sinh số giả ngẫu nhiên sang cho Bob:

```
scp stegano_image.png blum_blum_shub.py ubuntu@<IP bob>:/home/ubuntu
```

(Nhập “yes” và mật khẩu của terminal *bob* là ubuntu).

- Trên terminal *bob* kiểm tra các file đã nhận được bằng câu lệnh *ls*.

4.7. Nhiệm vụ 7: Tách thông điệp dạng nhị phân được giấu trong ảnh màu

- Thực hiện sinh ngẫu nhiên lại các vị trí trong chuỗi nhị phân của ảnh màu dùng để giấu các bit của thông điệp dạng nhị phân:

```
python3 blum_blum_shub.py
```

- File *extract_message.py* dùng để tách từng bit của thông điệp dạng nhị phân được giấu vào các vị trí trong chuỗi nhị phân của ảnh màu nhưng nội dung file còn thiếu. Hãy đọc file để hiểu nội dung và hoàn thành đoạn mã.

- Thực hiện tách từng bit của thông điệp dạng nhị phân được giấu vào các vị trí trong chuỗi nhị phân của ảnh màu:

```
python3 extract_message.py
```

4.8. Nhiệm vụ 8: Chuyển thông điệp dạng nhị phân sang văn bản

- Mọi thông điệp khi giấu vào ảnh đều phải ở dạng nhị phân. File *binary_to_text.py* dùng để chuyển thông điệp dạng nhị phân sang văn bản nhưng nội dung của file còn thiếu. Hãy đọc file để hiểu nội dung và hoàn thành đoạn mã.

- Thực hiện chuyển thông điệp dạng nhị phân tách được sang dạng văn bản:

```
python3 binary_to_text.py
```

5. Kiểm tra và kết thúc bài lab

- Trong quá trình làm bài, sinh viên thực hiện kiểm tra kết quả bài lab bằng câu lệnh:

```
checkwork
```

- Sau khi hoàn thành bài lab, sinh viên thực hiện kết thúc bài lab bằng câu lệnh:

```
stoplap
```

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

```
labtainer -r stego-image-basic-code-hvgnn3
```