

Bài Lab: Tách tin trong ảnh xám được giấu 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 tách tin trong ảnh được giấu 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 tách một văn bản trong ảnh được giấu bằng phương pháp này và có thể viết chương trình để tách một thông điệp cho trước trong ảnh được giấu bằng phương pháp này.

2. Yêu cầu đối với sinh viên

- Sinh viên đã thực hiện bài lab stego-image-basic-code-hvgnn1.
- Sinh viên nắm được kiến thức về ngôn ngữ Python, câu lệnh scp 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 file 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-hvgnn2>

- 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-hvgnn2/imodule.tar

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

imodule file:///home/student/Downloads/stego-image-basic-code-hvgnn2/imodule.tar

4. Nội dung thực hành

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

labtainer stego-image-basic-code-hvgnn2

(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 2 file .py và 2 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 tách thông điệp “PTIT” đã được giấu trong ảnh cho sẵn. Sau đây là những nhiệm vụ của bài lab:

4.1. Nhiệm vụ 1: 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 thống 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*.

4.2. Nhiệm vụ 2: Bob sinh các vị trí đã được dùng để giấu thông điệp dạng nhị phân trong chuỗi nhị phân của ảnh xám

- 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 xám 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 xám dùng để giấu các bit của thông điệp dạng nhị phân:

```
python3 blum_blum_shub.py
```

4.3. Nhiệm vụ 3: Bob tách thông điệp dạng nhị phân được giấu trong ảnh xám

- Đây là nhiệm vụ quan trọng nhất.
- 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 xám đã được sinh ra ở nhiệm vụ trên 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 xám:

python3 extract_message.py

- Sau khi chạy lệnh trên, sinh viên sẽ phải nhập 1 giá trị đầu vào là vị trí trong chuỗi nhị phân của ảnh xám dùng để giấu các bit của thông điệp dạng nhị phân. Ví dụ vị trí trong chuỗi nhị phân của ảnh xám là 9 thì nhập đầu vào là:

9

- Số lần sinh viên phải thực hiện để hoàn thành việc tách thông điệp dạng nhị phân khỏi ảnh xám bằng với độ dài của thông điệp dạng nhị phân. Hãy thực hiện thật cẩn thận.
- Đọc nội dung file kết quả để thấy được quá trình sinh viên đã thực hiện tách tin:

nano extract_process.txt

4.4. Nhiệm vụ 4: Bob 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-hvgnn2