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

2. Yêu cầu đối với sinh viên

- 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 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-hvgnn1

- 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-hvgnn1/imodule.tar

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

imodule file:///home/student/Downloads/stego-image-basic-code-hvgnn1/imodule.tar

4. Nội dung thực hành

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

labtainer stego-image-basic-code-hvgnn1

*(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 1 terminal *ubuntu* hiện ra. Hãy xem các file có trong đường dẫn hiện tại bằng câu lệnh *ls* (có 6 file .py và 1 file .png).

- Trong bài lab này, sinh viên sẽ thực hiện giấu thông điệp "PTIT" vào ảnh 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

- Đọc nội dung file kết quả:

nano binary_message.txt

4.2. Nhiệm vụ 2: Chuyển ảnh màu ban đầu sang ảnh xám

- Trong bài lab này, sinh viên sẽ thực hiện giấu thông điệp vào ảnh xám (ảnh xám và ảnh màu có sự khác nhau). File convert_image_to_grayscale.py dùng để chuyển ảnh màu sang ảnh xám. Hãy đọc file để hiểu nội dung đoạn mã.

- Xem ảnh màu ban đầu:

fim image.png

- Thực hiện chuyển ảnh màu ban đầu sang ảnh xám:

python3 convert_image_to_grayscale.py

- Xem ảnh xám sau khi chuyển:

fim gray_image.png

4.3. Nhiệm vụ 3: Trích xuất ma trận điểm ảnh dạng thập phân và nhị phân của ảnh xám

- 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 và nhị phân của một ảnh. Hãy đọc file để hiểu nội dung đoạn mã.

- Thực hiện trích xuất ma trận điểm ảnh dạng thập phân và nhị phân của ảnh xám:

python3 extract_image_matrix.py

- Đọc nội dung file kết quả:

nano decimal_matrix.txt

nano binary_matrix.txt

4.4. Nhiệm vụ 4: Chuyển ma trận điểm ảnh dạng nhị phân của ảnh xám sang chuỗi nhị phân

- Trong phương pháp hoán vị giả ngẫu nhiên, ma trận điểm ảnh sẽ được chuyển về chuỗi nhị phân để thực hiện giấu thông điệp. File binary_matrix_to_string.py dùng để chuyển ma trận điểm ảnh dạng nhị phân của một ảnh sang chuỗi nhị phân. Hãy đọc file để hiểu nội dung đoạn mã.

- Thực hiện chuyển ma trận điểm ảnh dạng nhị phân của ảnh xám sang chuỗi nhị phân:

python3 binary_matrix_to_string.py

- Đọc nội dung file kết quả:

nano binary_string.txt

4.5. Nhiệm vụ 5: Sinh ngẫu nhiên các vị trí 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:

python3 blum_blum_shub.py

4.6. Nhiệm vụ 6: Giấu thông điệp dạng nhị phân vào ảnh xám

- Đây là nhiệm vụ quan trọng nhất.

- Tạo 1 ảnh xám giống như ảnh gray_image.png để giấu thông điệp (mục đích là để sau khi giấu tin sinh viên có thể quan sát 2 ảnh xám để xem có thể nhận biết sự khác nhau bằng mắt thường được không):

cp gray_image.png stegano_image.png

- 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 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 giấu từng bit của thông điệp dạng nhị phân vào các vị trí trong chuỗi nhị phân của ảnh xám:

python3 hide_message.py

- Sau khi chạy lệnh trên, sinh viên sẽ phải nhập 2 giá trị đầu vào là vị trí trong chuỗi nhị phân của ảnh xám và bit cần giấu trong thông điệp dạng nhị phân. Ví dụ vị trí trong chuỗi nhị phân của ảnh xám là 9 và bit cần giấu trong thông điệp dạng nhị phân là 0 thì nhập đầu vào là:

9 0

- Số lần sinh viên phải thực hiện để hoàn thành việc giấu thông điệp dạng nhị phân vào 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 giấu tin:

nano hide_process.txt

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

fim stegano_image.png

- Sinh viên có thể trích xuất ma trận điểm ảnh dạng thập phân và nhị phân của ảnh xám 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 xám ban đầu (gray_image.png) xem những điểm ảnh nào có sự khác nhau.

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-hvgnn1