

Bài thực hành: Sử dụng thuật toán Histogram Shifting để giấu tin trong ảnh

1. Mục đích

Bài thực hành này hướng dẫn sử dụng thuật toán Histogram Shifting để giấu tin trong ảnh.

Việc giấu tin trong ảnh là một nhánh quan trọng trong bảo mật thông tin. Histogram Shifting là một phương pháp phổ biến vì khả năng giấu lượng lớn dữ liệu mà ít làm biến dạng ảnh.

2. Yêu cầu đối với sinh viên

Có kiến thức cơ bản về thuật toán Histogram Shifting.

3. Nội dung thực hành

3.1 Khởi động bài lab

Cài đặt bài lab tại: <https://github.com/dotrantrung2003/stego-image-code-hs>

Giải nén và chuyển vào thư mục `/labtainer/trunk/labs`.

Trên terminal, gõ:

```
labtainer -r stego-image-code-hs
```

Sau khi khởi động xong, hai terminal ảo xuất hiện: alice và bob.

Trên terminal alice, có các chương trình được sử dụng để thực hiện quá trình giấu tin.

Trên terminal bob, có các chương trình được sử dụng để thực hiện quá trình tách tin

3.2 Giấu tin

Trên terminal alice sử dụng lệnh: `find image.png` để xem ảnh được chuẩn bị.

Bước 1: Tiền xử lý

Trên terminal alice, gõ lệnh:

```
python3 preprocessing.py image.png
```

Chương trình `preprocessing.py` sẽ chuyển đổi từ ảnh gốc ban đầu (`image.png`) thành ảnh xám (`gray_image.png`) và ma trận điểm ảnh của ảnh xám (`gray_matrix.txt`).

Ảnh xám này sẽ được sử dụng để giấu tin.

Quan sát ma trận điểm ảnh của ảnh xám, ta có nhận thấy các giá trị đều nằm trong đoạn $[0, 255]$, các giá trị này đại diện cho độ sáng của điểm ảnh tại vị trí đó.

Bước 2: Tần suất điểm ảnh

Trên terminal alice, gõ lệnh:

```
python3 frequency.py gray_matrix.txt
```

Chương trình *frequency.py* sẽ trích xuất tần suất xuất hiện của các giá trị điểm ảnh (*frequency.txt*) trong ma trận điểm ảnh xám.

Bước 3: Áp dụng thuật toán Histogram Shifting.

Để áp dụng được thuật toán này, ta cần 2 giá trị được gọi là Peak Point và Zero Point.

Peak Point là giá trị có tần suất xuất hiện nhiều nhất.

Zero Point là giá trị có tần suất xuất hiện ít nhất.

Quan sát *frequency.txt*, ta thấy giá trị 49 sẽ là Peak Point, giá trị 254 sẽ là Zero Point.

Trên terminal alice, gõ lệnh:

```
python3 histogram_shifting.py gray_matrix.txt 49 254
```

Chương trình *histogram_shifting.py* sẽ làm dịch chuyển tần suất xuất hiện của các giá trị trong đoạn [50, 253] của ma trận điểm ảnh (*gray_matrix.txt*) bằng cách tăng các giá trị lên 1 đơn vị ($50 \rightarrow 51$, $51 \rightarrow 52$, ..., $253 \rightarrow 254$).

Kết quả thu được là 1 ma trận điểm ảnh mới (*new_gray_matrix.txt*). Ma trận mới này sẽ không còn xuất hiện giá trị 50 nữa. Do đó ta sẽ lới dụng 2 giá trị 49 và 50 để giấu tin.

Bước 4: Giấu tin

Thông điệp cần giấu là: PTIT.

Đưa thông điệp cần giấu về dạng nhị phân 8 bit của mã ASCII tương ứng. Độ dài thông điệp cần giấu khi đó sẽ là 32.

Trên terminal alice, gõ lệnh:

```
python3 steganography_1.py new_gray_matrix.txt 49 32
```

Chương trình *steganography_1.py* sẽ trích xuất 32 giá trị 49 đầu tiên (*steganography.txt*) trong ma trận điểm ảnh mới (*new_gray_matrix.txt*).

Chỉnh sửa *steganography.txt*. Theo thứ tự thông điệp giấu, bit 0 thì giữ nguyên 49, bit 1 sửa thành 50.

Trên terminal alice, gõ lệnh:

```
python3 steganography_2.py new_gray_matrix.txt 49
```

Chương trình *steganography_2.py* sẽ Thay chuỗi vừa sửa lại vào ma trận sao cho đúng vị trí ban đầu.

Bước 5: Khôi phục ảnh

Trên terminal alice, gõ lệnh:

```
python3 restore.py new_gray_matrix.txt
```

Chương trình *restore.py* sẽ chuyển đổi từ ma trận điểm ảnh trở thành hình ảnh (*new_gray_matrix.txt*).

Kết quả thu được ảnh xám mới đã được giấu tin.

Bước 6: Gửi ảnh

Khởi động dịch vụ *ssh* trên cả 2 máy. Nếu gặp lỗi khởi động được, có thể do dịch vụ *xinetd* chiếm cổng (tắt *xinetd*).

Trên terminal alice, gõ lệnh:

```
sudo scp new_gray_image.png ubuntu@<ip_bob>:/home/ubuntu
```

(Mật khẩu: ubuntu)

3.3 Tách tin

Bước 1: Tiền xử lý

Trên terminal bob, gõ lệnh:

```
python3 preprocessing.py new_gray_image.png
```

Chương trình *preprocessing.py* sẽ chuyển đổi từ ảnh gốc ban đầu (*new_gray_image.png*) thành ma trận điểm ảnh của ảnh xám (*gray_matrix.txt*).

Bước 2: Tần suất điểm ảnh

Trên terminal bob, gõ lệnh:

```
python3 frequency.py gray_matrix.txt
```

Chương trình *frequency.py* sẽ trích xuất tần suất xuất hiện của các giá trị điểm ảnh (*frequency.txt*) trong ma trận điểm ảnh xám.

Quan sát xem giá trị nào là Peak Point => 2 giá trị được sử dụng để giấu tin là Peak Point và Peak Point + 1.

Bước 3: Tách tin

Biết độ dài tin là 32. Trên terminal bob, gõ lệnh:

```
python3 desteganography.py gray_matrix.txt <Peak Point> <Peak Point + 1>  
32
```

Chương trình *desteganography.py* sẽ trích xuất 32 giá trị đầu tiên hoặc bằng Peak Point hoặc bằng Peak Point + 1 (*desteganography.txt*) trong ma trận điểm ảnh (*gray_matrix.txt*).

Quan sát *desteganography.txt*, giá trị Peak Point tương đương với bit 0, giá trị Peak Point + 1 tương đương với bit 1.

Thu được chuỗi nhị phân. Trên terminal bob, gõ lệnh:

```
python3 plain_text.py <chuỗi nhị phân>
```

Chương trình *plain_text.py* đưa chuỗi nhị phân mã ASCII tương ứng.

So sánh thông điệp trước và sau khi giấu

3.4 Kết thúc bài lab

Trên terminal, gõ:

stoplab

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

Sinh viên cần nộp file *.lab* để chấm điểm.

Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh:

checkwork

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, sử dụng lệnh:

labtainer -r stego-image-code-hs