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

1. Mục đích

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

Việc giấu tin trong ảnh là một nhánh quan trọng trong bảo mật thông tin. DFT 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 PVD.

3. Nội dung thực hành

3.1 Khởi động bài lab

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

Giản nén và chuyển vào thư mục */labtainer/trunk/labs*.

Trên terminal, gõ:

labtainer -r stego-image-code-pvd

Sau khi khởi động xong, một terminal ảo xuất hiện.

3.2 Tiền xử lý

Trên terminal sử dụng lệnh: *fim image.png* để xem ảnh được chuẩn bị.

Trên terminal, 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.

3.3 Thuật toán PVD

Thuật toán Pixel Value Differencing (PVD) là một kỹ thuật giấu tin (steganography) trong ảnh số, thuộc miền không gian. PVD tận dụng sự khác biệt giá trị giữa các cặp pixel liền kề để nhúng dữ liệu bí mật.

Tùy theo quy ước của người giấu tin, với mỗi cặp pixel có thể giấu được số lượng bit nhất định. Trên terminal sử dụng lệnh: *cat PVD.txt* để xem quy ước của bài thực hành này.

[0-7] ~ 2 bit

[8-15] ~ 3 bit

[16-31] ~ 4 bit

[32-63] ~ 5 bit

[64-127] ~ 6 bit

[128-255] ~ 7 bit

Với bảng quy ước trên ta hiểu rằng nếu hiệu của cặp pixel nằm trong khoảng [a-b] thì sẽ giấu được n bit ([a-b] ~ n bit).

Giả sử ta có cặp pixel (P1 = 104, P2 = 95) và thông điệp cần giấu là 01010000:

- Ta có tổng của cặp pixel: s = P1 + P2 = 104 + 95 = 199.
- Ta có hiệu của cặp pixel: d = P1 – P2 = 104 – 95 = 9.
- Ta thấy d thuộc [lower – upper] = [8-15] => Giấu được 3 bit.
- Lấy 3 bit từ thông điệp cần giấu: m = 010.
- Chuyển về thập phân: m = 010 => m = 2.
- Ta có hiệu mới của cặp pixel: d’ = lower + m = 8 + 2 = 10.
- Ta sẽ tính lại 2 giá trị P1, P2 sao cho có hiệu là hiệu mới, tổng ít thay đổi nhất có thể.
- P1’ = (s + d’) / 2 = (199 + 10) / 2 = 104.5 = 105 (làm tròn lên).
- P2’ = (s - d’) / 2 = (199 - 10) / 2 = 94.5 = 95 (làm tròn lên).
- Thu được cặp bit mới thay thế cặp bit cũ. Cặp bit mới này đã giấu được 3 bit đầu tiên của thông điệp.
- Với các bit còn lại làm tương tự với các cặp pixel khác. Trong trường hợp cặp pixel có thể giấu được 5 bit mà độ dài tin còn lại là 3 bit thì cần thêm các bit 0 vào cuối thông điệp cần giấu cho đủ độ dài.

3.4 Giấu tin

Để hoàn thành bài thực hành này, sinh viên cần giấu thông điệp “PTIT”.

Chuyển thông điệp về dạng nhị phân của mã ASCII.

Trên terminal, gõ lệnh:

python3 15_pairs.py gray_matrix.txt

Chương trình 15_pairs.py sẽ trích xuất 15 cặp pixel đầu tiên của hàng cuối cùng của ma trận điểm (*gray_matrix.txt*) và lưu vào trong *15_pairs.txt.*

Quan sát *15_pairs.txt.*

Trên terminal, gõ lệnh: *sudo cp 15_pairs.txt new_15_pairs.txt* để tạo 1 bản sao. Ta sẽ chỉnh sửa trên bản sao này.

Với mỗi cặp pixel ta thực hiện giấu tin như hướng dẫn ở mục trên. Để hoàn thành bài thực hành cần giấu toàn bộ thông điệp "PTIT".

Sau khi đã giấu xong, gõ lệnh:

python3 PVD.py new_15_pairs.txt gray_matrix.txt

Chương trình PVD.py sẽ thay 15 cặp pixel đầu tiên của hàng cuối cùng của ma trận điểm (*gray_matrix.txt*) bằng 15 cặp pixel mới (*new_15_pairs.txt)* thu được ma trận điểm ảnh mới (*new_gray_matrix.txt*).

3.5 Khôi phục ảnh

Trên terminal, 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.

3.6 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-pvd