

- Thực hiện giấu file nhị phân bị xáo trộn vào file PDF

`./encode.py output_original.pdf binary_message.txt output_key.pdf y_positions.txt`

```
ubuntu@stego-dsv-lsbimage:~$ ./encode.py output_original.pdf binary_message.txt output_key.pdf y_
positions.txt
Đã giấu 32 bit vào văn bản.
PDF đã được vẽ lại và ghi vào output_key.pdf
```

Kết quả: File nhị phân bị xáo trộn được giấu vào file PDF.

Task 4: Giải mã lấy ra file nhị phân giấu trong PDF và vị trí xáo trộn trong ảnh

Mục tiêu: Lấy file bị giấu trong pdf và trong ảnh.

- Lấy file nhị phân bị xáo trộn trong file PDF

`./decode.py output_original.pdf output_key.pdf`

```
ubuntu@stego-dsv-lsbimage:~$ ./decode.py output_original.pdf output_key.pdf
Thông điệp dạng nhị phân: 010001010000001101000000011110100
```

- Hoàn thành code lấy file vị trí xáo trộn trong ảnh

```
binary_data = ""
```

```
for y in range(height):
```

```
    for x in range(width):
```

```
        r, g, b = pixels[y, x]
```

```
        binary_data += str(r & 1)
```

```
        binary_data += str(g & 1)
```

```
        binary_data += str(b & 1)
```

```
# Trích xuất chuỗi từ ảnh
binary_data = ""
binary_data = ""
for y in range(height):
    for x in range(width):
        r, g, b = pixels[y, x]
        binary_data += str(r & 1)
        binary_data += str(g & 1)
        binary_data += str(b & 1)
```

- Lấy file vị trí xáo trộn giấu trong ảnh

```
ubuntu@stego-dsv-lsbimage:~$ ./extract_from_image.py image.png bit_swap_info.txt
[+] Đã trích xuất dữ liệu và lưu vào: bit_swap_info.txt
ubuntu@stego-dsv-lsbimage:~$
```