Ẩn tin mật trên ảnh (phần 2)

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Buổi này

- Nói thêm về phương pháp LSB ở buổi trước (cho ảnh nonpalette-based)
- Phương pháp LSB cho ảnh palette-based

Đo sự khác biệt giữa stego image và cover image

- Ngoài cách định tính là nhìn bằng mắt, ta có thể định lượng sự khác biệt giữa hai ảnh bằng độ đo **MSE (Mean Squared Error)**
- MSE giữa ảnh I_1 và ảnh I_2 = trung bình của bình phương hiệu hai giá trị tương ứng giữa I_1 và I_2
  - Ví dụ:
    - Cho I_1 là ảnh xám gồm 4 pixel $[1, 3, 5, 7]$
    - Cho I_2 là ảnh xám gồm 4 pixel $[2, 4, 6, 8]$
    - $MSE(I_1, I_2) = \frac{1}{4}\left((1-2)^2 + (3-4)^2 + (5-6)^2 + (7-8)^2\right)$
- Ở những slide kế tiếp, khi chỉ ghi MSE thì hiểu là MSE giữa stego image và cover image

Ẩn tin mật trên ảnh bằng phương pháp LSB

Ảnh trước khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

Số bit LSB (trong 8 bit) được nhúng

k = 1

MSE = 0.5

Ảnh sau khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 2
MSE = 1.6

Ảnh sau khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 3
MSE = 13.3

Ảnh sau khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 4
MSE = 70.6

Ảnh sau khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 5
MSE = 320.8

Ảnh sau khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 6
MSE = 1255.1

Ảnh sau khi nhúng

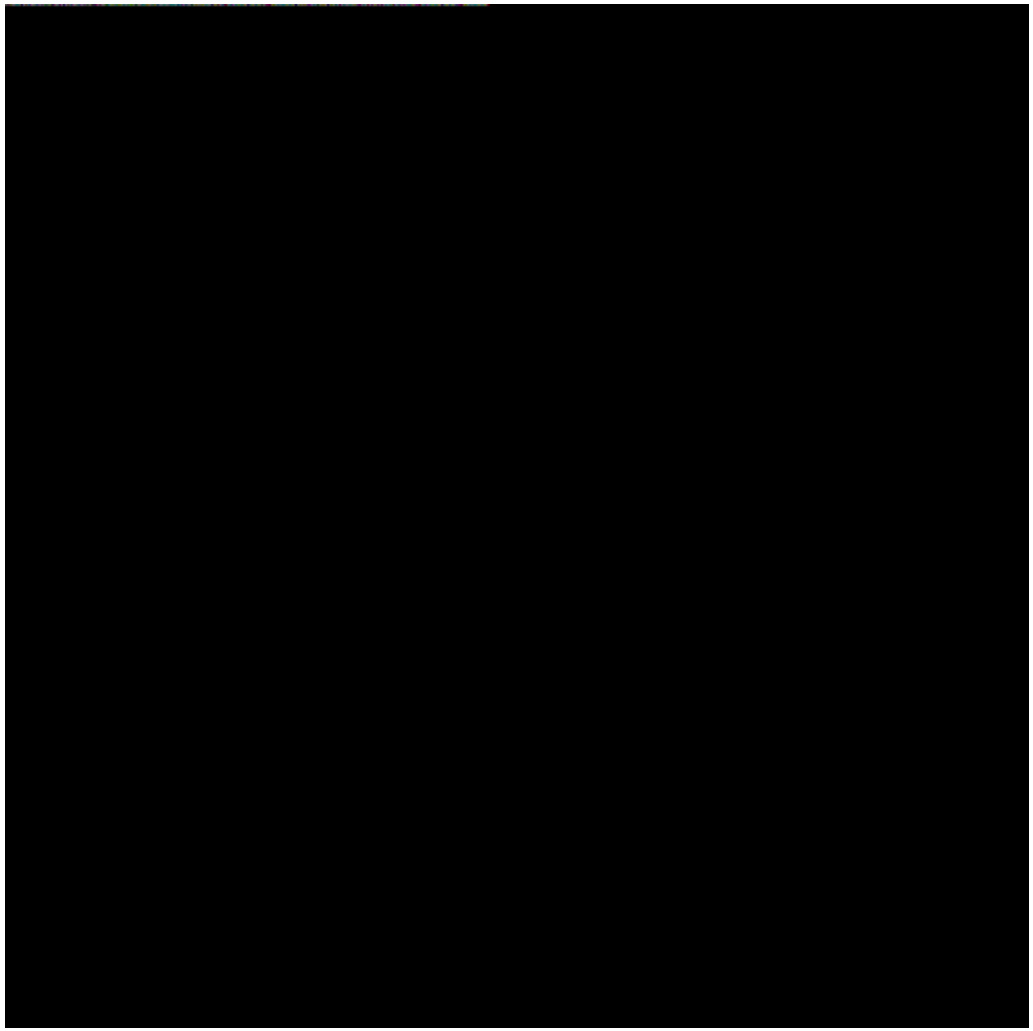
Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 7
MSE = 6164.3

Ảnh sau khi nhúng

Ẩn tin mật trên ảnh bằng phương pháp LSB

k = 8
MSE = 19024.0



Ảnh sau khi nhúng

Q: Trong phương pháp LSB, mục đích của việc thêm đuôi ‘100...’ vào chuỗi bit mật là gì?

A:

Q: Nhược điểm của cách làm này?

A: Cho dù chuỗi bit mật ngắn hay dài thì luôn nhúng trên toàn bộ ảnh → có thể sẽ làm giảm tính vô hình

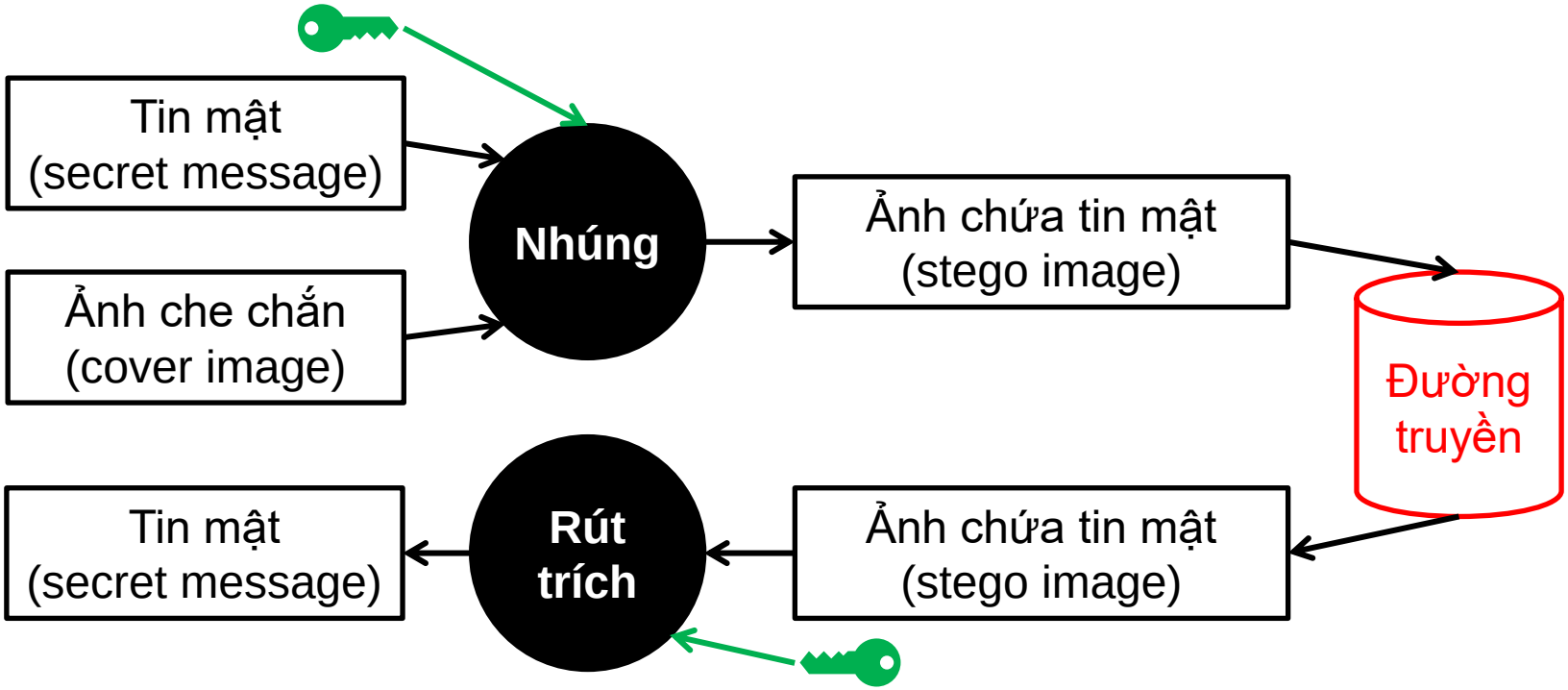
Q: Có cách làm nào khác mà vẫn đạt được mục đích của cách thêm ‘100...’ nhưng không phải nhúng trên toàn bộ ảnh?

A: Dùng một số bit để lưu chiều dài của chuỗi bit mật; chuỗi bit nhúng = chuỗi bit lưu chiều dài + chuỗi bit mật

Q: Bên thứ 3 có thể biết về phương pháp LSB; làm thế nào để bên thứ 3 khó có thể rút trích được tin mật ngay cả khi họ biết về phương pháp LSB?

A: Dùng thêm khóa (chỉ Alice và Bob biết), chẳng hạn:

- Dùng khóa mã hóa: mã hóa tin mật rồi mới nhúng
- Dùng khóa random seed: nhúng bit mật vào các phần tử trên ảnh cover theo thứ tự ngẫu nhiên của random seed



Q: Ảnh cover là ảnh xám có kích thước 100×100. Để đảm bảo tính vô hình ta chỉ muốn dùng $k = 1$ bit LSB. Nếu chuỗi bit mật có chiều dài lớn hơn 100×100 thì làm sao để vẫn có thể nhúng?

A: Nén chuỗi bit mật lại

Ảnh nonpalette-based vs palette-based

File **.bmp** (nonpalette), 768K

File **.gif** (palette), 276K

Ảnh palette-based

Bảng màu gồm 256 màu RGB được chọn ra từ 256^3 màu RGB

Giá trị pixel = chỉ số màu (0-255) trong bảng màu

```
[[ 48  48  43 ...,  39  72 119]
 [ 48  48  43 ...,  39  72 119]
 [ 48  48  43 ...,  39  72 119]
 ...,
 [241 241 232 ..., 167 166 181]
 [241 241 227 ..., 157 157 155]
 [241 241 227 ..., 157 157 155]]
```

File **.gif** (palette, lossless), 276K

Nói thêm về các tính chất của ảnh

- Đã biết các tính chất sau của ảnh:
  - Grayscale hay RGB
  - Nonpalette-based hay palette-based
- Ngoài ra, còn có: lossless (không nén hoặc nén không mất mát thông tin) hay lossy (nén mất mát thông tin)
- Một số định dạng ảnh:
  - BMP: lossless, palette-based hoặc nonpalette-based
  - GIF: lossless, palette-based
  - PNG: lossless, palette-based hoặc nonpalette-based
  - JPEG: lossy, nonpalette-based
  - ...
- Ẩn tin mật trên ảnh lossy?

Ẩn tin mật bằng phương pháp LSB trên ảnh lossy

File stego.**bmp** (lossless),
dung lượng: 768K

Chuỗi rút trích được:
“I love u”

File stego.**jpg** (lossy),
dung lượng: 39K

Chuỗi rút trích được:
“\xc4\xed\xb6\xdb\x13...”

Nói thêm về các tính chất của ảnh

- Đã biết các tính chất sau của ảnh:
  - Grayscale hay RGB
  - Nonpalette-based hay palette-based
- Ngoài ra, còn có: lossless (không nén hoặc nén không mất mát thông tin) hay lossy (nén mất mát thông tin)
- Một số định dạng ảnh:
  - BMP: lossless, palette-based hoặc nonpalette-based
  - GIF: lossless, palette-based
  - PNG: lossless, palette-based hoặc nonpalette-based
  - JPEG: lossy, nonpalette-based
  - ...
- Ẩn tin mật trên ảnh lossy?
  - Khó ☹, tạm để lại sau, trước mắt chỉ làm với ảnh lossless

Buổi này

- Nói thêm về phương pháp LSB ở buổi trước (cho ảnh nonpalette-based)
- Phương pháp LSB cho ảnh palette-based

Phương pháp LSB cho ảnh palette-based

- Ảnh nonpalette-based: giá trị pixel = màu
  → thay đổi bit LSB của giá trị pixel thì màu sẽ thay đổi một ít
  → mắt người khó nhận biết ☺
- Ảnh palette-based: giá trị pixel = chỉ số màu trong bảng màu
  → thay đổi bit LSB của giá trị pixel thì chỉ số màu sẽ thay đổi một ít
  → có chắc là 2 màu có chỉ số gần nhau trong bảng màu nhìn sẽ gần giống nhau?
  - Không chắc ☹

Phương pháp LSB cho ảnh palette-based

Một cách là sắp xếp lại bảng màu sao cho các màu nhìn gần giống nhau sẽ nằm cạnh nhau, sau đó áp dụng LSB như bình thường

- Với mỗi màu trong bảng màu được biểu diễn bởi 3 giá trị (R, G, B) thì sắp xếp bảng màu như thế nào?
- Một cách là: với mỗi màu, tính giá trị $\sqrt{R^2 + G^2 + B^2}$, rồi sắp xếp bảng màu theo giá trị này
  - Hai màu (128, 0, 0) và (0, 128, 0) hoàn toàn khác nhau nhưng lại có cùng giá trị theo cách tính ở trên ☹

Phương pháp LSB cho ảnh palette-based

Một cách khác tốt hơn (Jiri Fridrich,1999) mà không sắp xếp lại bảng màu là:

- Nhúng:

  Để nhúng một bit mật vào một pixel:

  - Tìm trong bảng màu màu gần nhất với màu của pixel đang xét (theo khoảng cách Euclid) và có (R+G+B) % 2 khớp với bit mật đang xét
  - Sửa giá trị pixel thành chỉ số của màu vừa tìm được

- Rút trích?

  - Với một pixel, ta sẽ có được màu tương ứng trong bảng màu, và bit = (R+G+B) % 2 với (R, G, B) là giá trị của màu này