

# Ăn tin mật trên ảnh (phần 3)



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

# Hai buổi trước: ẩn tin mật trên ảnh lossless bằng phương pháp LSB

## ☐ Ảnh nonpalette-based

- ☐ Giá trị pixel = màu
- ☐ Thay đổi bit LSB của giá trị pixel thì màu sẽ thay đổi một ít 😊

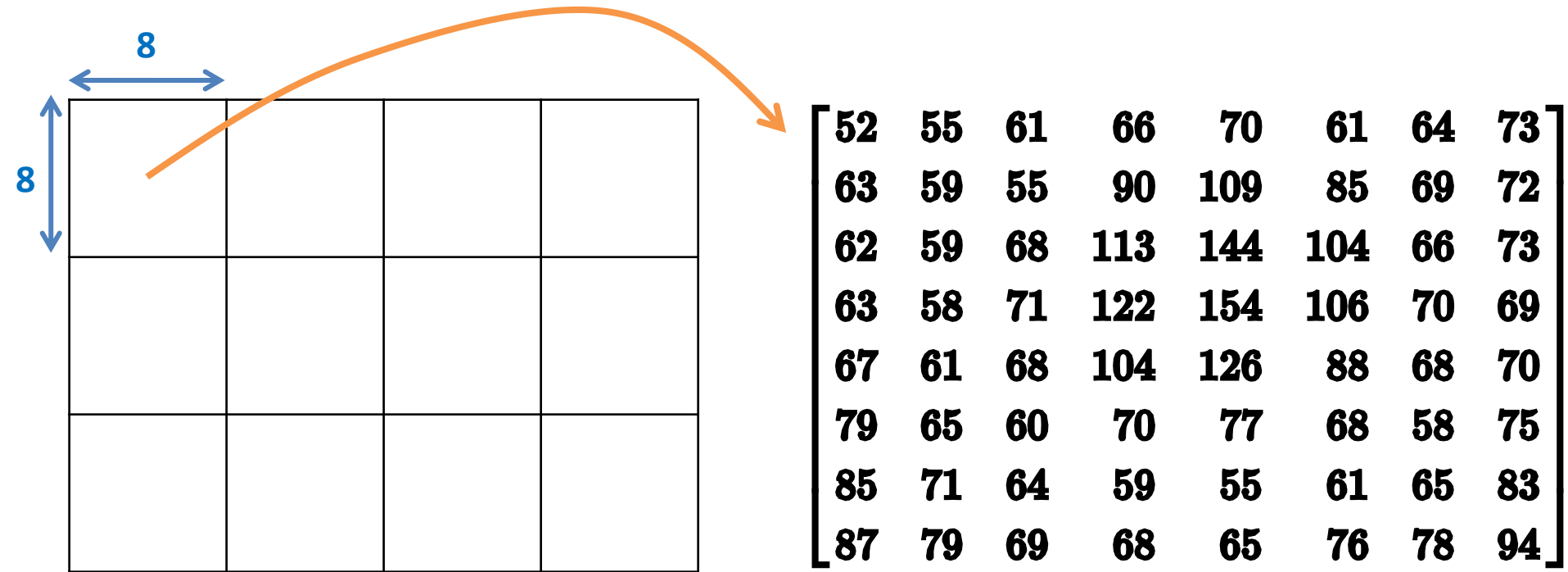
## ☐ Ảnh palette-based

- ☐ 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, không chắc là 2 màu có chỉ số gần nhau trong bảng màu sẽ nhìn gần giống nhau 😞
- ☐ Một số đề xuất:
  - Sort lại bảng màu với mong muốn các màu có chỉ số gần nhau trong bảng màu sẽ nhìn gần giống nhau, rồi mới dùng LSB  
Khó có thể đạt được mong muốn này
  - Phương pháp của Fridrich: 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, rồi sửa giá trị pixel thành chỉ số của màu vừa tìm được  
Rủi ro: để tìm được màu phù hợp, có thể sẽ phải đi xa màu của pixel đang xét

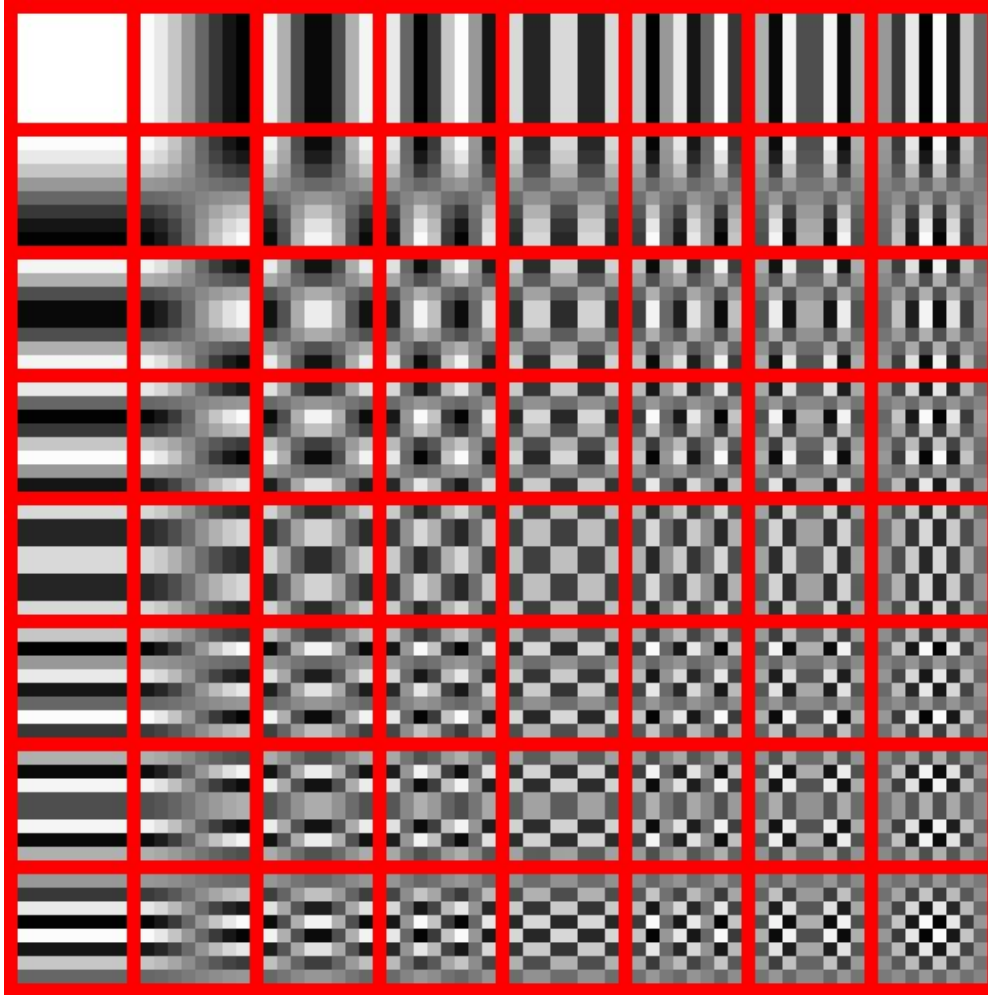
# Buổi này: ẩn tin mật trên ảnh lossy (ảnh JPEG)

- ☐ JPEG nén ảnh như thế nào?
- ☐ Ẩn tin mật ở đâu trên ảnh JPEG?

# Nén JPEG - B1: chia ảnh thành các khối $8 \times 8$ (đang xét ảnh grayscale hoặc một kênh màu)



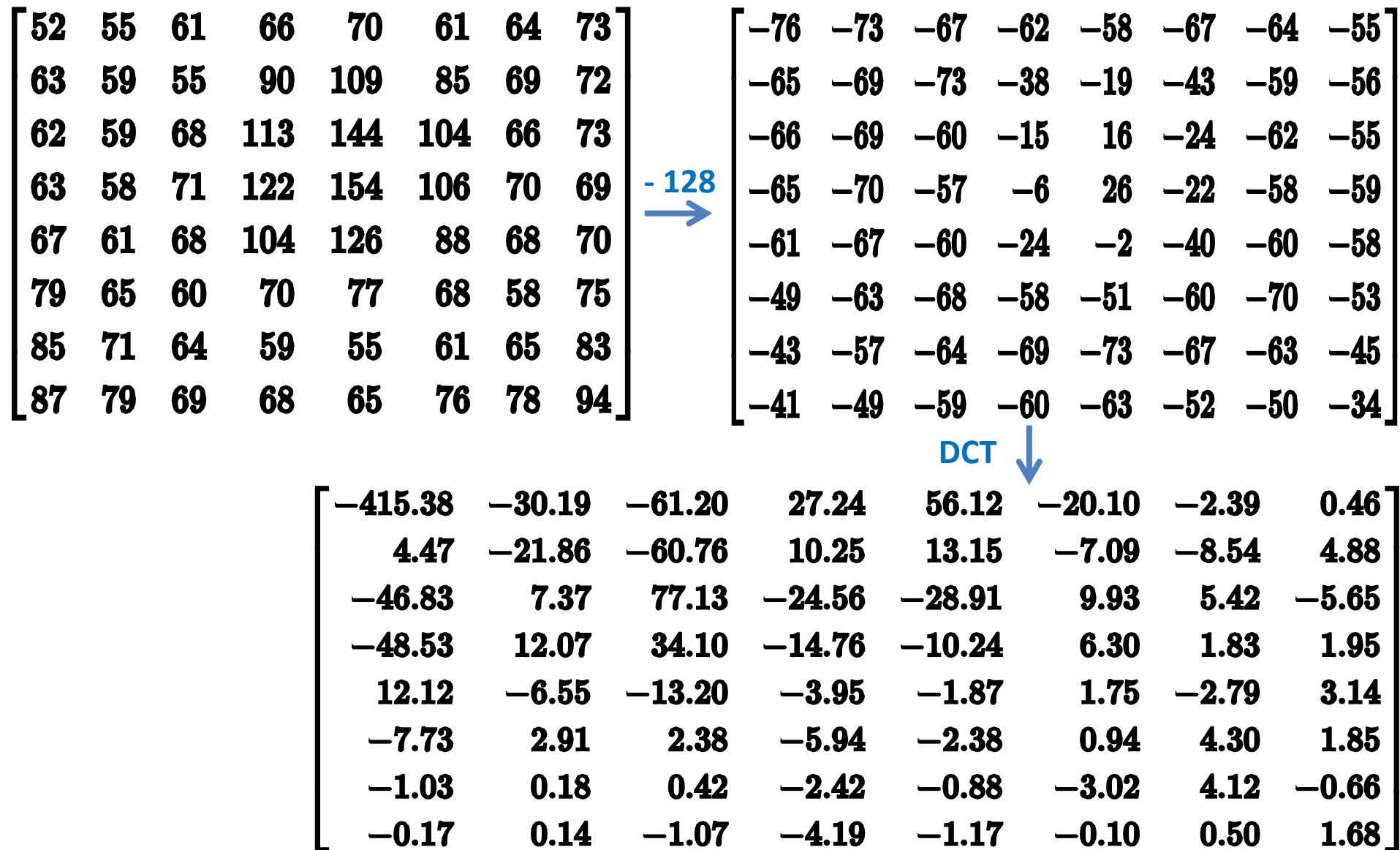
Một ảnh  $8 \times 8$  bất kỳ có thể được tạo thành bằng cách kết hợp tuyến tính 64 ảnh cơ bản dưới đây



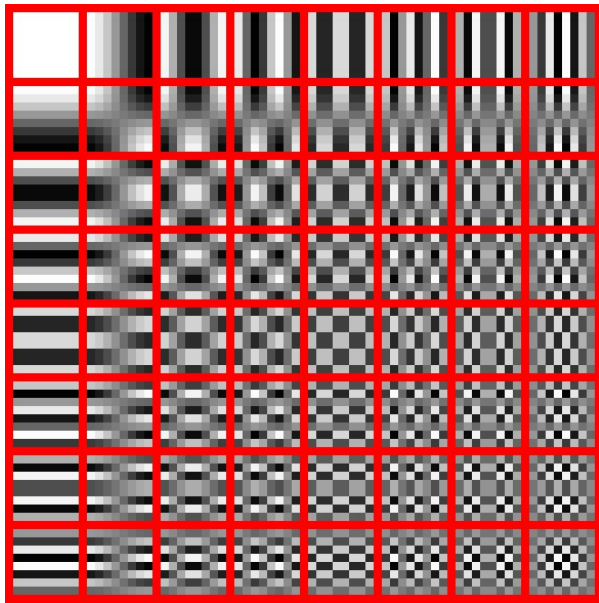
Nguồn ảnh: <https://en.wikipedia.org/wiki/JPEG>

Có thể xem mỗi ảnh cơ bản  $8 \times 8$  là một véc-tơ 64 chiều, véc-tơ này đã được chuẩn hóa (có độ dài bằng 1) và trực giao với các véc-tơ ứng với các ảnh cơ bản khác  
→ 64 ảnh cơ bản này tạo thành một hệ tọa độ mới trong không gian 64 chiều

# Nén JPEG – B2: biến đổi DCT (Discrete Cosine Transform)



# Nén JPEG – B2: biến đổi DCT



-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-73	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-60	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-59	-59	24

Đây là các hệ số của các ảnh cơ bản tương ứng, gọi là các hệ số DCT

Tính hệ số DCT như thế nào?

Miền giá trị của hệ số DCT?

-415.38	-30.19	-61.20	27.24	56.12	-20.10	-2.59	0.46
4.47	-21.86	-60.76	10.25	13.15	-7.09	-8.54	4.88
-46.83	7.37	77.13	-24.56	-28.91	9.93	5.42	-5.65
-48.53	12.07	34.10	-14.76	-10.24	6.30	1.83	1.95
12.12	-6.55	-13.20	-3.95	-1.87	1.75	-2.79	3.14
-7.73	2.91	2.38	-5.94	-2.38	0.94	4.30	1.85
-1.03	0.18	0.42	-2.42	-0.88	-3.02	4.12	-0.66
-0.17	0.14	-1.07	-4.19	-1.17	-0.10	0.50	1.68

DCT



Tính ngược lại như thế nào?

# Nén JPEG – B3: quantization (lossy)

$$\begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}$$

Chia cho phần tử tương ứng trong Q và làm tròn thành số nguyên gần nhất

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$



# Nén JPEG – B4: nén lossless

:

-26	-3	-6	2	2	1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-3	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

.

↓ Chuyển thành mảng một chiều

-26, -3, 0, -3, -2, -6, 2, -4, 1, -3, 1, 1, 5, 1, 2, -1, 1, -1, 2, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0, ...

↓ Nén lossless

Sơ lược một số ý tưởng nén:

- -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ..., 0 → -1, **END**
- 0, 0, 0, 0, 0, -1 → **4**, -1
- ...

# JPEG

**Nén:**

Khối ảnh  $8 \times 8$



DCT



Quantization (**lossy**)



Nén lossless



Chuỗi bit nén

**Giải nén:**

Khối ảnh  $8 \times 8$



DCT<sup>-1</sup>



“Quantization<sup>-1</sup>”



Giải nén lossless



Chuỗi bit nén

# JPEG

**Nén:**

Khối ảnh  $8 \times 8$



DCT



Quantization (**lossy**)



Nén lossless



Chuỗi bit nén

**Giải nén:**

Khối ảnh

Ẩn tin mật vào  
chỗ nào?

Chuỗi bit nén

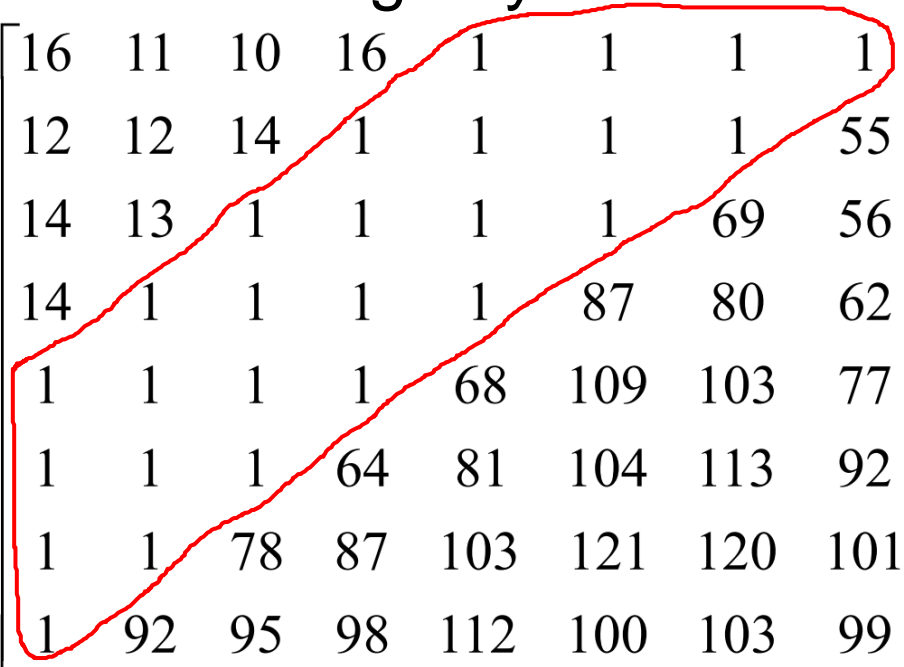
# Ẩn tin mật vào chỗ nào trong quá trình nén JPEG?

- ☐ Ẩn tin mật vào ma trận hệ số quantized DCT, ví dụ dùng phương pháp LSB
- ☐ Nhúng LSB với tất cả các hệ số quantized DCT?
  - ☐ Tính vô hình: 😞
  - ☐ Hiệu suất nén: 😞

# Ăn tin mật vào chỗ nào trong quá trình nén JPEG?

Chang, Chen, & Chung (2001) đề xuất:

- ❑ Sửa các giá trị trong bảng quantization tương ứng với các tần số trung thành 1
- ❑ Nhúng LSB vào các hệ số quantized DCT ứng với các tần số trung này



16	11	10	16	1	1	1	1
12	12	14	1	1	1	1	55
14	13	1	1	1	1	69	56
14	1	1	1	1	87	80	62
1	1	1	1	68	109	103	77
1	1	1	64	81	104	113	92
1	1	78	87	103	121	120	101
1	92	95	98	112	100	103	99

Bảng quantization  
được sửa đổi