

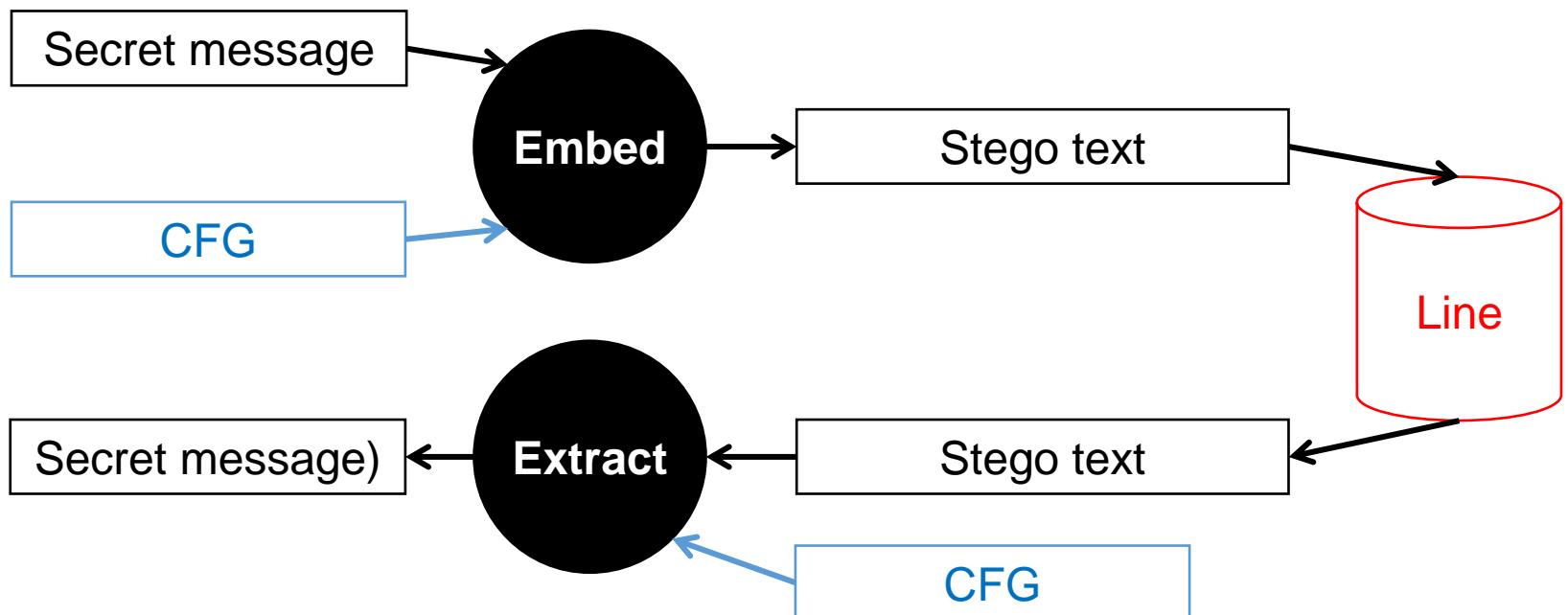
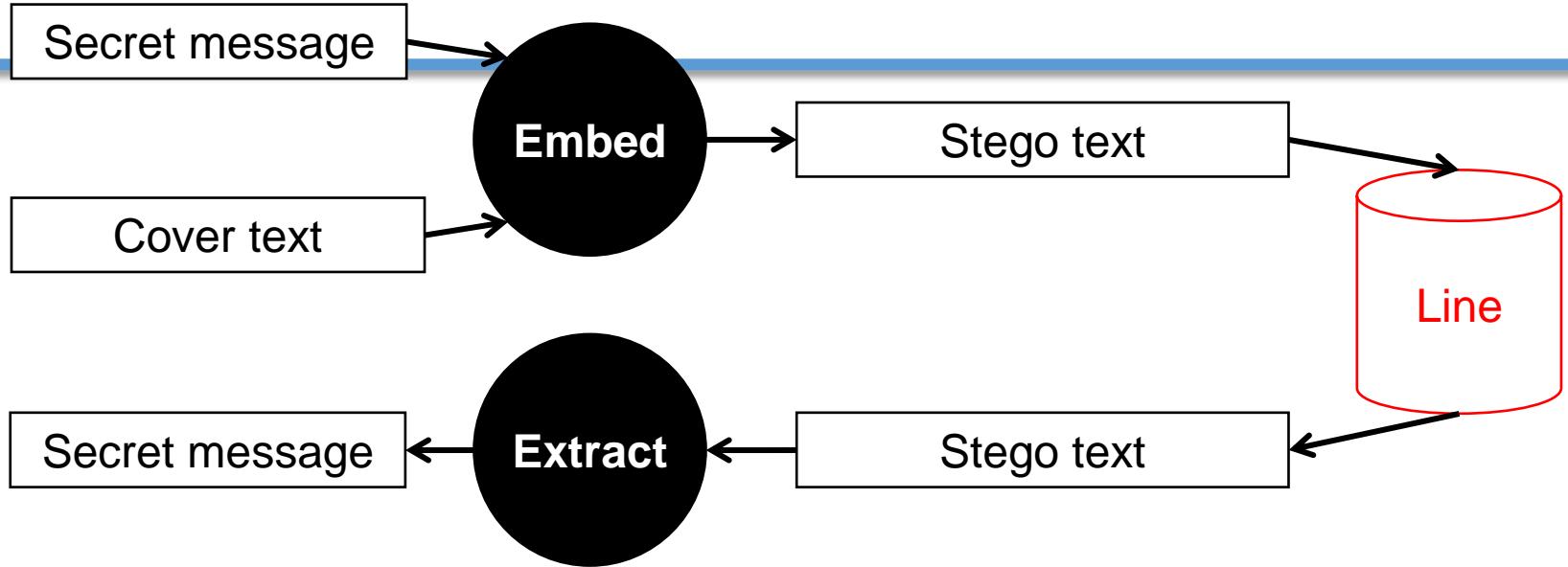
Lecture slides of the course
Information hiding & secret sharing

Image Steganography (P1)

Phạm Trọng Nghĩa

ptnghia@fit.hcmus.edu.vn

Last lecture



This lecture: hide secret information on images

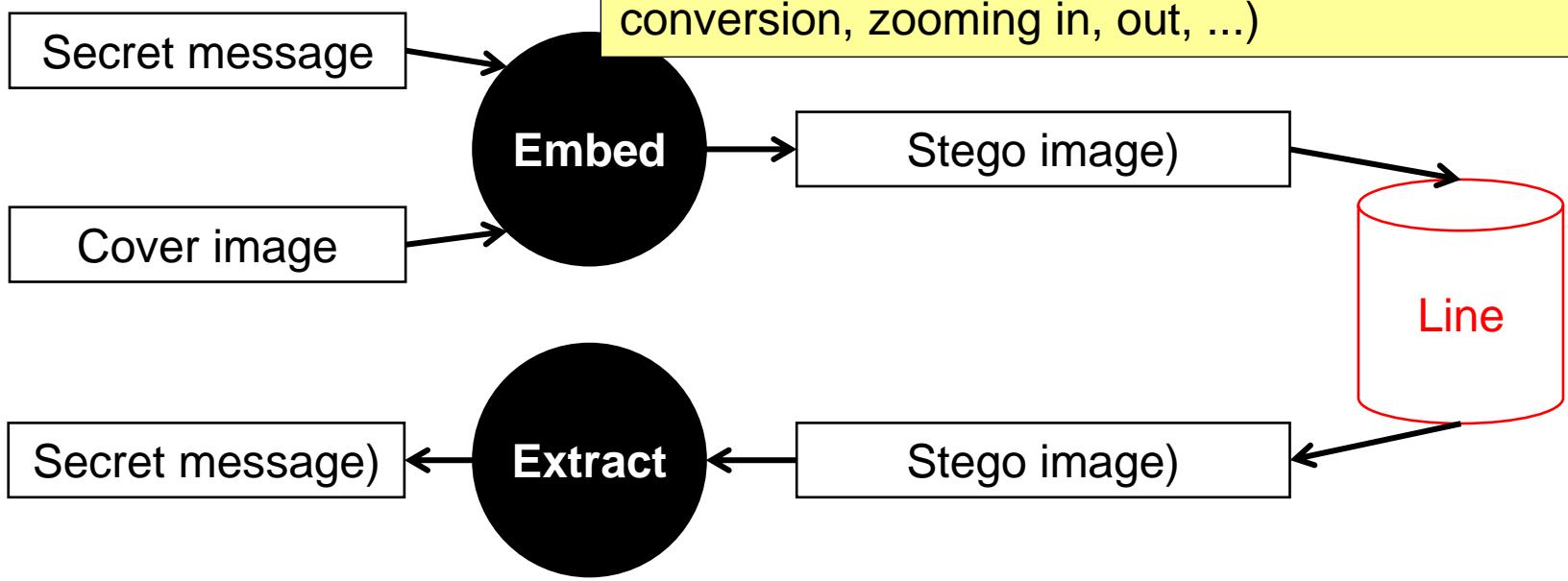
- The problem of hiding secret information on photos
- LSB (Least Significant Bit) method – a simple method to hide secret information on images

Steganography on Image

Desire:

Invisibility: it must be difficult for a third person to know the existence of secret information in stego image

In addition, there are other desires: **capacity**, robustness (secret information is not lost when there are editing operations on images: format conversion, zooming in, out, ...)

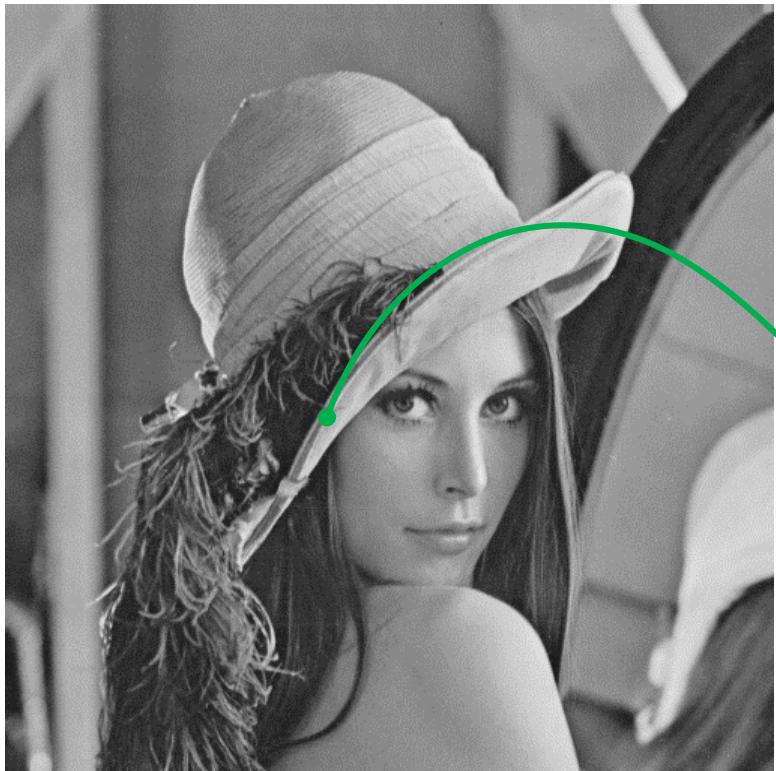


Steganography on Image

Q: Where should the secret be embedded in the image?

A: First, need to understand about photos...

Simple image: grayscale



A pixel is represented by an 8-bit integer, ranging from 0-255 .



0

255

Where should we embed secret information on grayscale image?

- If we change the **LSB (Least Significant Bit)** of a pixel value, how much will this value change?
 - From -1 to 1
 - With this change, will the human eye detect it?
 - Hard to detect 😊
- Embed the secret bit in the LSB bit of the pixel value

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

Bit representation: **11111111**

Bit string: **01100**

255	250	254	240
1	1	2	0
250	251	245	246

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

Bit representation: **11111111**

Bit string: **01100**

255	250	254	240
1	1	2	0
250	251	245	246

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

Bit representation: **11111111**

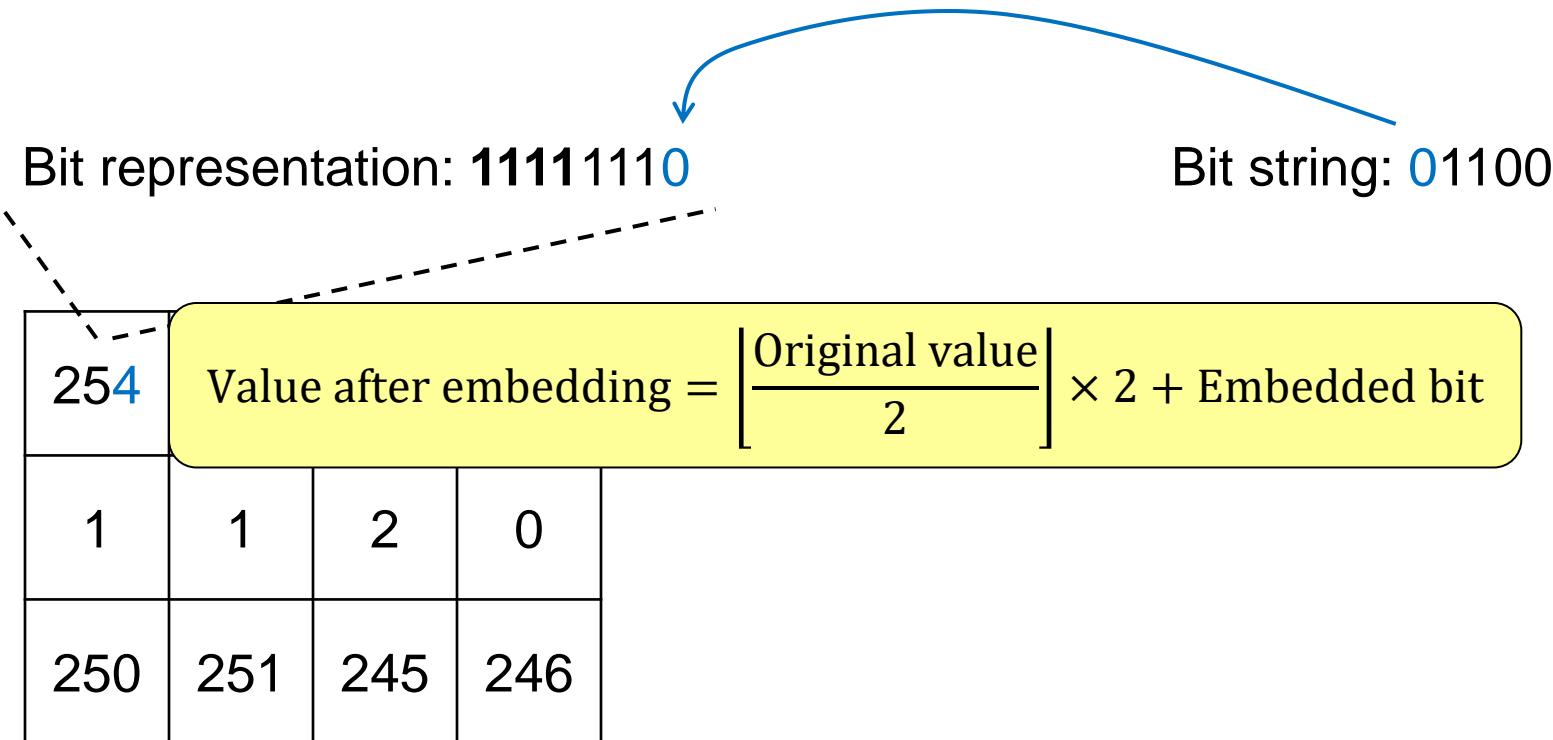
Bit string: **01100**

The diagram illustrates the process of embedding a secret bit string into a 4x4 grayscale image. A dashed arrow points from the bit representation '11111111' to the top row of the image matrix. Another dashed arrow points from the bit string '01100' to the bottom row of the image matrix. The image matrix contains the following values:

255	250	254	240
1	1	2	0
250	251	245	246

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below



Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

Bit string: 01100

254	250	254	240
1	1	2	0
250	251	245	246

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

Bit string: 01100

254	251	254	240
1	1	2	0
250	251	245	246

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

254	251	254	240
1	1	2	0
250	251	245	246

Bit string: 01100

The diagram illustrates the process of embedding a secret bit string (01100) into a 3x4 grayscale image matrix. The matrix contains the following values:

254	251	254	240
1	1	2	0
250	251	245	246

A blue arrow points from the bit string "01100" to the third column of the matrix, specifically highlighting the value 254. This indicates that the secret bit "1" is being embedded into the least significant bit position of the value 254.

Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

254	251	255	240
1	1	2	0
250	251	245	246

Bit string: 01100



Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

254	251	255	240
1	1	2	0
250	251	245	246

Bit string: 01100



Embedding secret bits on grayscale image using LSB method

Example 1: embed secret bit string 01100 in the image below

254	251	255	240
1	1	2	0
250	251	245	246

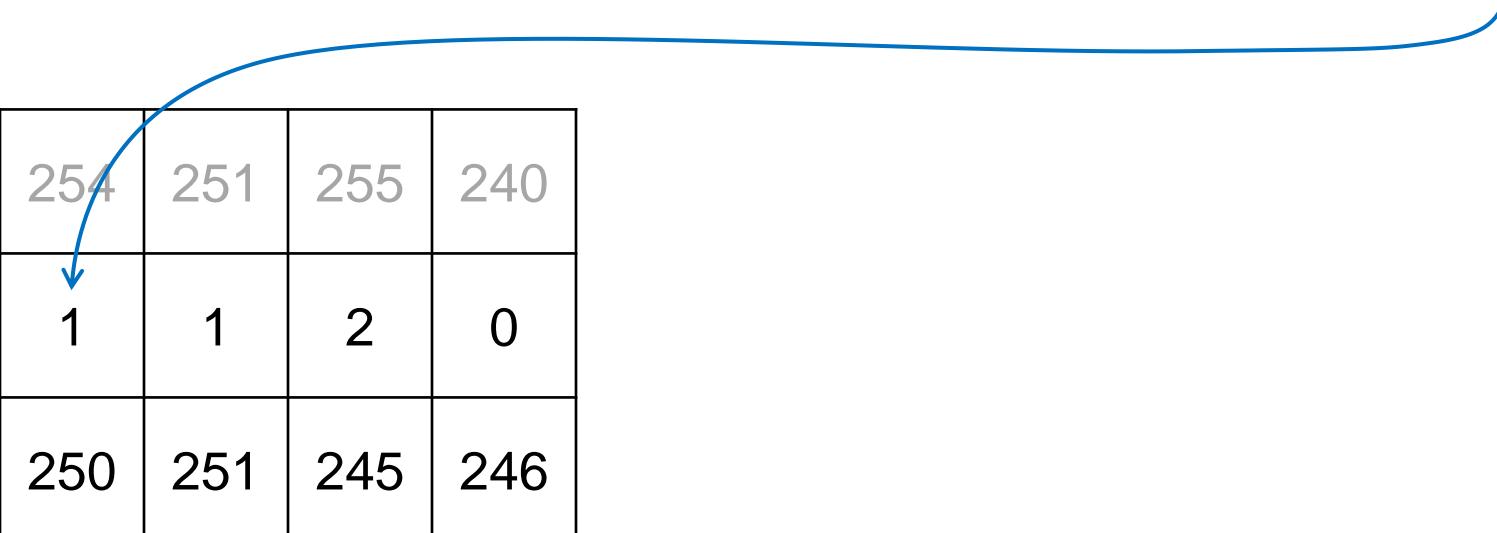
Bit string: 01100



Nhúng bit mật trên ảnh xám bằng phương pháp LSB

Example 1: embed secret bit string 01100 in the image below

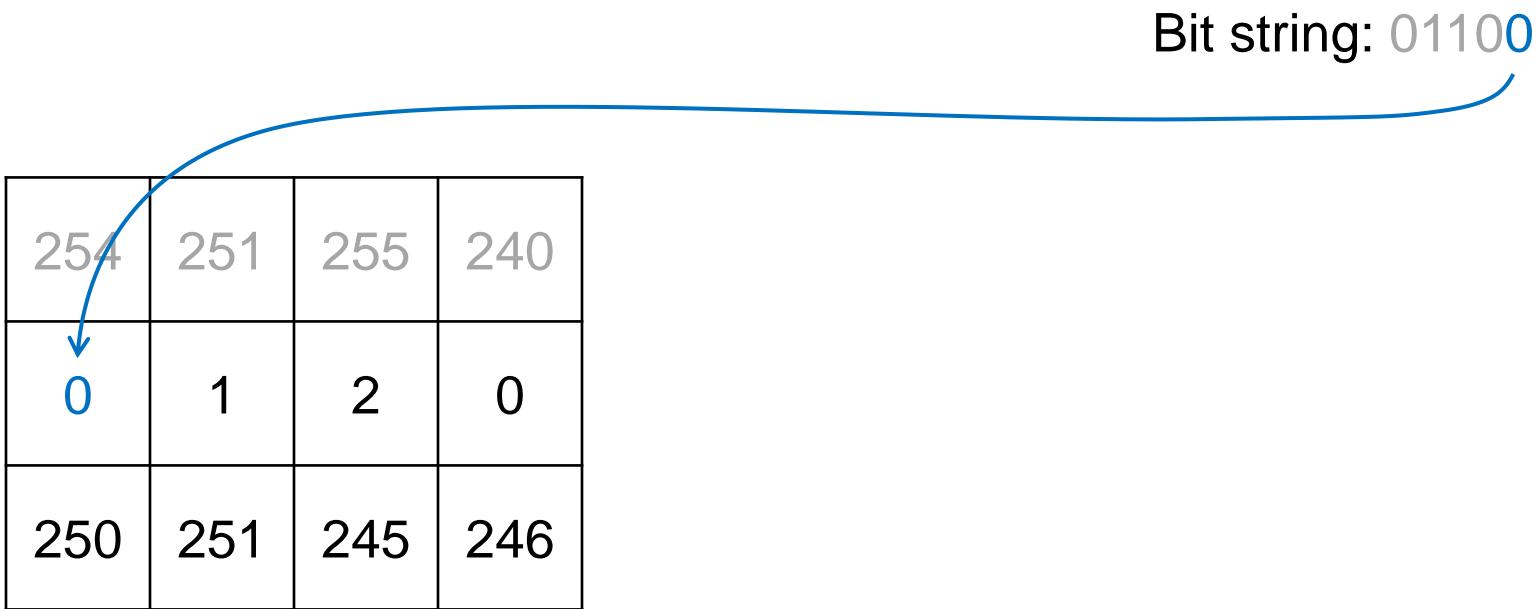
Bit string: 01100			
254	251	255	240
1	1	2	0
250	251	245	246



The diagram illustrates the process of embedding a secret bit string (01100) into a 4x4 grayscale image. The image is represented by a grid of 16 pixels. The secret bit string is mapped to the pixel values in the second row of the grid. The mapping is as follows: Bit 0 maps to pixel 1 (value 1), Bit 1 maps to pixel 2 (value 1), Bit 2 maps to pixel 3 (value 2), and Bit 3 maps to pixel 4 (value 0). The original pixel values for the first row are 254, 251, 255, and 240 respectively. The original pixel values for the third row are 250, 251, 245, and 246 respectively. The fourth row contains the same values as the third row. A blue arrow points from the bit string '01100' to the value '1' in the first cell of the second row.

Nhúng bit mật trên ảnh xám bằng phương pháp LSB

Example 1: embed secret bit string 01100 in the image below



Nhúng bit mật trên ảnh xám bằng phương pháp LSB

Example 1: embed secret bit string 01100 in the image below

254	251	255	240
0	1	2	0
250	251	245	246

Try extracting from the embedded image...

Extract bit:
 $\text{bit} = \text{value \% 2}$

0	1	1	0
0	1	0	0
0	1	1	0

How to get the embedded bit string???

Nhúng bit mật trên ảnh xám bằng phương pháp LSB

How to get the embedded bit string???

254	251	255	240
0	1	2	0
250	251	245	246

Extract bit:
 $\text{bit} = \text{value \% 2}$

0	1	1	0
0	1	0	0
0	1	1	0

One way is when embedding a 100... in the cryptographic bit string and embedding over the end. When extracting, we will get a bit string ending in 100... and could easily cut this tail off

Nhúng bit mật trên ảnh xám bằng phương pháp LSB

Example 2: embed the secret bit string 0110001011100010 (16 bits) into the image below

255	250	254	240
1	1	2	0
250	251	245	246

The image has 12 pixel, each can store 1 bit, but need to embed up to 16 bits
□ How can still embed?

Nhúng bit mật trên ảnh xám bằng phương pháp LSB

Example 2: embed the secret bit string 0110001011100010 (16 bits) into the image below

255	250	254	240
1	1	2	0
250	251	245	246

Embed ***k bits*** in ***k LSB bits*** of each pixel
For example, ***k = 2*** then we will be able to embed
 $12 \times 2 = 24$ bits

Increase *k*: capacity? invisibility?

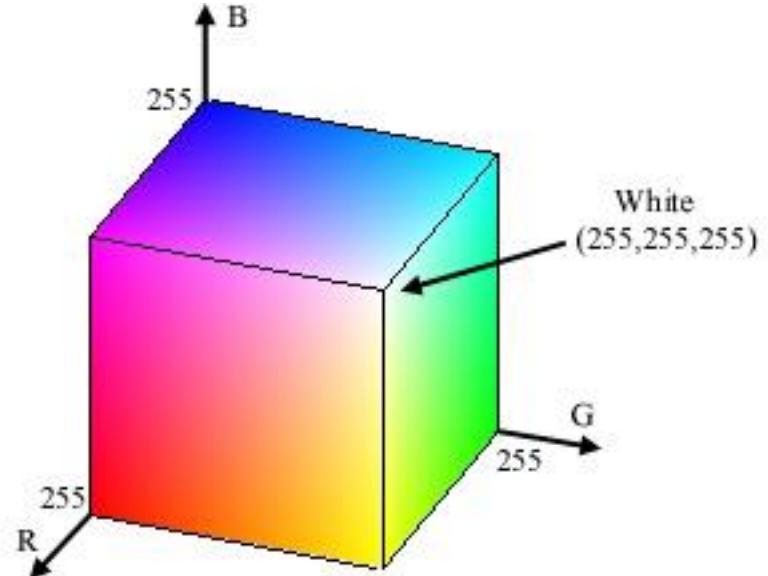
Demo

RGB color Image



Each pixel is represented by a set of 3 integers (**r**, **g**, **b**), each integer has a value from 0-255 (8 bits), where :

- **r** indicate the intensity of **red**
- **g** indicate the intensity of **green**
- **b** indicate the intensity of **blue**



Nguồn ảnh RGB cube:

<http://radio.feld.cvut.cz/matlab/toolbox/images/color4.htm>

Dùng phương pháp LSB cho ảnh màu RGB như thế nào?



Red Channel

Blue Channel

Green Channel

Embed LSB on each color channel like grayscale image

Analyze on the LSB method

- Invisibility?
 - 😊
- Capacity?
 - 😊
- Robustness?
 - 😞

