

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN**



CHUYÊN ĐỀ AN NINH MẠNG

**Các phương pháp nâng cao độ bảo mật,
tăng dung lượng cho thông tin giấu trên LSB
và phương pháp CPP phát hiện ảnh có giấu tin**

Giảng viên: TS. Đỗ Xuân Chợt

Nhóm môn học: 02

Nhóm: 15

Phạm Văn Minh B18DCAT164

Nguyễn Thanh Hải B18DCAT072

Hà Nội - 11/2022

MỤC LỤC

MỤC LỤC	1
I. TỔNG QUAN	3
1.1 Tổng quan về kỹ thuật giấu tin	3
1.1.1 Khái niệm	3
1.1.2 Mục đích của giấu tin	3
1.1.3 Mô hình kỹ thuật giấu thông tin cơ bản	4
1.1.4 Mô hình kỹ thuật tách thông tin cơ bản	5
1.1.5 Yêu cầu đối với một hệ thống giấu tin	5
1.1.6 Môi trường giấu tin	6
1.1.7 Một số đặc điểm của việc giấu tin trên ảnh	7
1.2 Tổng quan về kỹ thuật phát hiện ảnh có giấu tin	8
1.2.1 Khái niệm	8
1.2.2 Phân tích ảnh giấu tin thường dựa vào các yếu tố	8
1.2.3 Các phương pháp phân tích ảnh có giấu tin	9
II. KỸ THUẬT GIẤU TIN TRÊN LSB	9
2.1. Khái niệm bit có trọng số thấp LSB	9
2.2. Thuật toán giấu thông tin mật trên LSB	10
2.2.1 Ý tưởng thuật toán	10
2.2.2 Thuật toán giấu	12
2.2.3 Thuật toán tách	13
2.3. Thuật toán giấu thông tin mật trên LSB kết hợp secret key	13

2.3.1 Ý tưởng thuật toán	13
2.3.2 Thuật toán giấu	15
2.3.3 Thuật toán tách	16
2.3.4 Ví dụ	16
2.4. Thuật toán Pixel Value Differencing (PVD)	17
2.4.1 Ý tưởng thuật toán	17
2.4.2 Thuật toán giấu	18
2.4.3 Thuật toán tách	20
2.4.4 Ví dụ Pixel Value Differencing	21
III. KỸ THUẬT PHÁT HIỆN ẢNH CÓ GIẤU TIN TRÊN LSB BẰNG PHƯƠNG PHÁP PHÂN TÍCH CẶP MÀU GẦN NHAU CCP (Close Color Pair)	24
3.1. Tổng quan	24
3.2. Một số khái niệm	24
3.3. Thuật toán phát hiện	27
IV. THỬ NGHIỆM	29
4.1. Giấu tin trên LSB	29
4.2. Giấu tin trên LSB kết hợp secret key	31
4.3 Pixel Value Differencing	34
4.4. Phân tích cặp màu gần nhau - Close Color Pair	35

I. TỔNG QUAN

1.1 Tổng quan về kỹ thuật giấu tin

1.1.1 Khái niệm

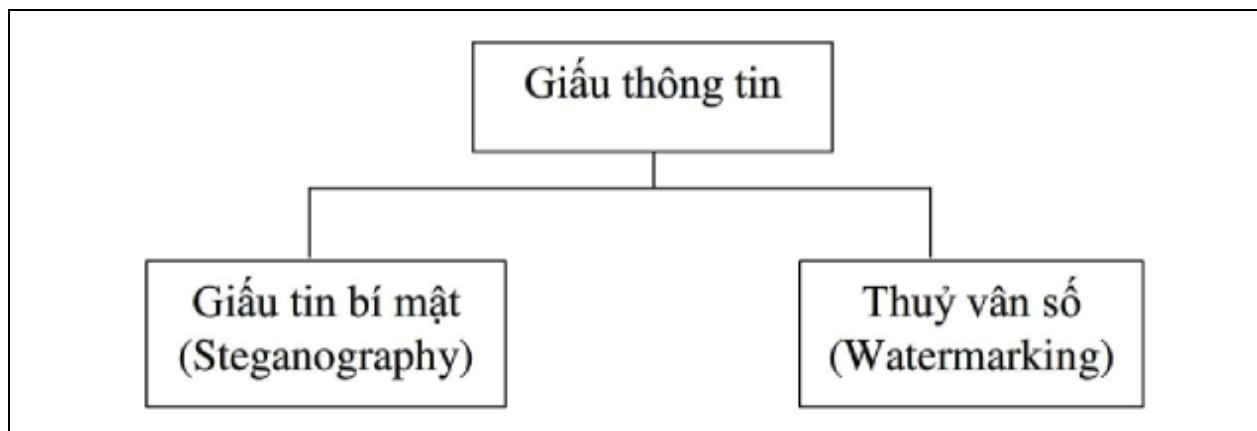
Giấu thông tin là một kỹ thuật nhúng (giấu) một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác (giấu thông tin chỉ mang tính quy ước, không phải là một hành động cụ thể).

1.1.2 Mục đích của giấu tin

Có 2 mục đích của giấu tin:

- Trao đổi thông tin mật
- Bảo đảm an toàn và phát hiện xuyên tạc thông tin cho chính các đối tượng chứa dữ liệu giấu trong đó

Có thể thấy 2 mục đích này hoàn toàn trái ngược nhau và dần phát triển thành 2 lĩnh vực với những yêu cầu và tính chất khác nhau.



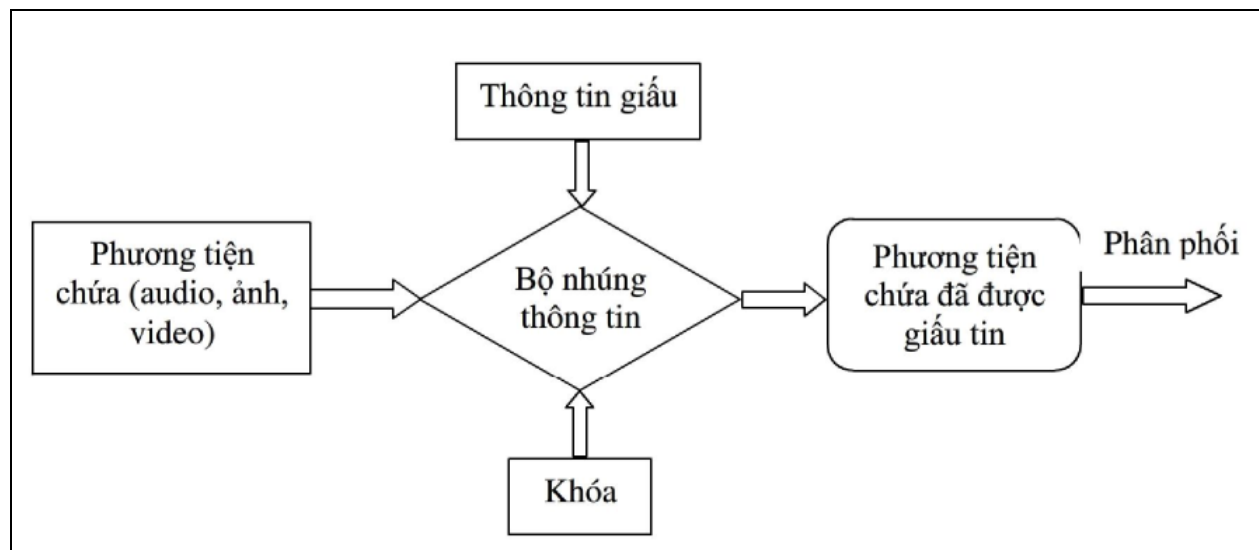
Hình 1.1 Hai lĩnh vực chính của kỹ thuật giấu thông tin

Kỹ thuật giấu thông tin bí mật (Steganography): với mục đích đảm bảo an toàn và bảo mật thông tin tập trung vào các kỹ thuật giấu tin để có thể giấu được nhiều thông tin nhất. Thông tin mật được giấu một cách vô hình trong một đối tượng khác sao cho người khác khó phát hiện được.

Kỹ thuật giấu thông tin theo kiểu đánh dấu - thủy vân (watermarking) với mục đích để bảo vệ bản quyền chính đối tượng dùng để chứa thông tin, thường tập trung đảm bảo một số các yêu cầu như đảm bảo tính bền vững... Đây là ứng dụng cơ bản nhất của kỹ thuật thủy vân số.

1.1.3 Mô hình kỹ thuật giấu thông tin cơ bản

Giấu thông tin vào phương tiện chứa và tách lấy thông tin là 2 quá trình trái ngược nhau và có thể mô tả qua sơ đồ khối của hệ thống như hình dưới:



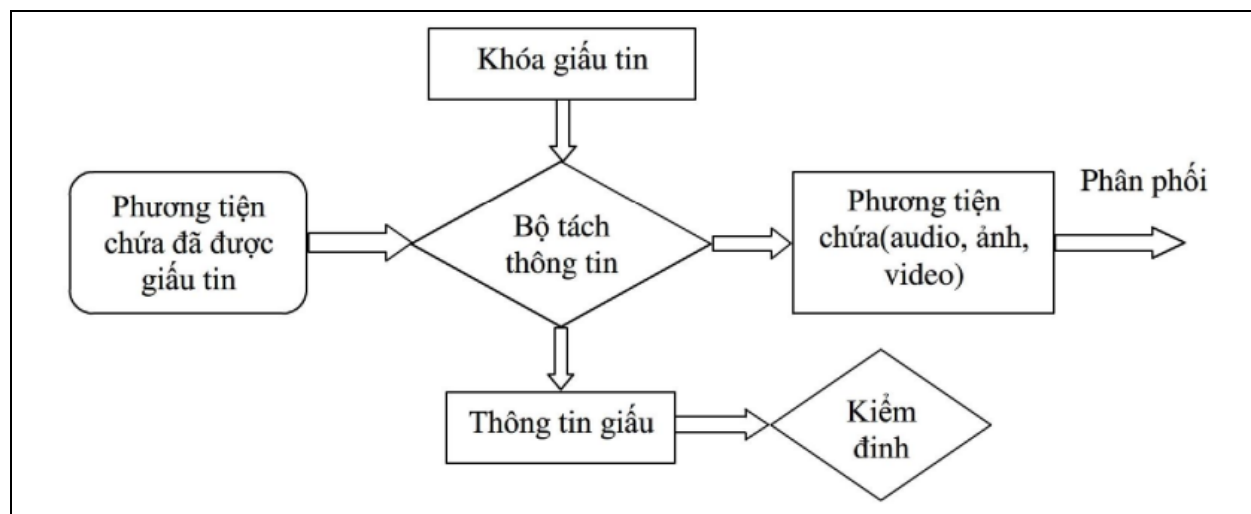
Hình 1.2 Lược đồ chung cho quá trình giấu tin

- Thông tin cần giấu tùy theo mục đích của người sử dụng, nó có thể là thông tin mật (với các tin bí mật) hay các logo, hình ảnh bản quyền.
- Phương tiện chứa: các file ảnh, text, audio... là môi trường để nhúng tin.
- Bộ nhúng thông tin: là những chương trình thực hiện việc giấu tin.
- Đầu ra: là các phương tiện chứa đã có tin giấu trong đó.

Tách thông tin từ các phương tiện chứa diễn ra theo quy định ngược lại với đầu ra là các thông tin đã được giấu vào phương tiện chứa. Phương tiện chứa

sau khi tách lấy thông tin có thể được sử dụng, quản lý theo những yêu cầu khác nhau.

1.1.4 Mô hình kỹ thuật tách thông tin cơ bản



Hình 1.3. Lược đồ chung cho quá trình tách thông tin

Hình trên chỉ ra các công việc giải mã thông tin đã giấu. Sau khi nhận được đối tượng phương tiện chứa có giấu thông tin, quá trình giải mã được thực hiện thông qua một bộ giải mã tương ứng với bộ nhúng thông tin cùng với khóa của quá trình nhúng. Kết quả thu được bao gồm phương tiện chứa gốc và thông tin đã giấu. Bước tiếp theo thông tin đã được giấu sẽ được xử lý kiểm định so sánh với thông tin ban đầu.

1.1.5 Yêu cầu đối với một hệ thống giấu tin

Có 3 yêu cầu thiết yếu đối với một hệ thống giấu tin :

- Tính vô hình: là một trong 3 yêu cầu của bất kì 1 hệ giấu tin nào.
- Tính bền vững: là yêu cầu thứ 2 của một hệ giấu tin. Tính bền vững là nói đến khả năng chịu được các thao tác biến đổi nào đó trên phương tiện nhúng và các cuộc tấn công có chủ đích.
- Khả năng nhúng: là yêu cầu thứ 3 của một hệ giấu tin. Khả năng nhúng chính là số lượng thông tin nhúng được nhúng trong phương tiện chứa.

1.1.6 Môi trường giấu tin

a. Giấu tin trong ảnh

Giấu tin trong ảnh hiện đang rất được quan tâm. Nó đóng vai trò hết sức quan trọng trong hầu hết các ứng dụng bảo vệ an toàn thông tin như: nhận thực thông tin, xác định xuyên tạc thông tin, bảo vệ bản quyền tác giả... Một đặc điểm của giấu thông tin trong ảnh nữa đó là thông tin được giấu một cách vô hình, nó như là cách truyền thông tin mật cho nhau mà người khác không thể biết được bởi sau khi giấu thông tin chất lượng ảnh gần như không thay đổi đặc biệt đối với ảnh màu hay ảnh xám.

b. Giấu tin trong audio

Khác với kỹ thuật giấu thông tin trong ảnh: phụ thuộc vào hệ thống thị giác của con người - HSV (Human Vision System), kỹ thuật giấu thông tin trong audio lại phụ thuộc vào hệ thống thính giác HAS (Human Auditory System). Bởi vì tai con người rất kém trong việc phát hiện sự khác biệt giữa các dải tần và công suất, có nghĩa là các âm thanh to, cao tần có thể che giấu đi được các âm thanh nhỏ, thấp một cách dễ dàng.

Yêu cầu cơ bản và quan trọng nhất của giấu tin trong audio là đảm bảo tính chất ẩn của thông tin được giấu đồng thời không làm ảnh hưởng đến chất lượng của dữ liệu.

c. Giấu tin trong video

Cũng giống như giấu thông tin trong ảnh hay trong audio, giấu tin trong video cũng được quan tâm và được phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, xác thực thông tin, bản quyền tác giả...

Một phương pháp giấu tin trong video được đưa ra bởi Cox là phương pháp phân bố đều. Ý tưởng cơ bản của phương pháp là phân phối thông tin giấu dàn trải theo tần số của dữ liệu gốc.

d. Giấu tin trong văn bản dạng text

Giấu tin trong văn bản dạng text khó thực hiện hơn do có ít các thông tin dư thừa, để làm được điều này người ta phải khéo léo khai thác các dư thừa tự nhiên của ngôn ngữ. Một cách khác là tận dụng các định dạng văn bản (mã hoá thông tin vào khoảng cách giữa các từ hay các dòng văn bản) => Kỹ thuật giấu tin đang được áp dụng cho nhiều loại đối tượng chứ không riêng dữ liệu đa phương tiện như ảnh, audio, video.

1.1.7 Một số đặc điểm của việc giấu tin trên ảnh

- Tính vô hình của thông tin:

Khái niệm này dựa trên đặc điểm của hệ thống thị giác của con người. Thông tin nhúng là không tri giác được nếu một người với thị giác bình thường không phân biệt được ảnh môi trường và ảnh kết quả (tức là không phân biệt được ảnh trước và sau khi giấu thông tin). Trong khi image hiding (Steganography) yêu cầu tính vô hình của thông tin ở mức độ cao thì watermarking lại chỉ yêu cầu ở một cấp độ nhất định. Chẳng hạn như người ta áp dụng watermarking cho việc gắn một biểu tượng mờ vào một chương trình truyền hình để bảo vệ bản quyền

- Khả năng nhúng tin:

Lượng thông tin giấu so với kích thước ảnh môi trường cũng là một vấn đề cần quan tâm trong một thuật toán giấu tin. Rõ ràng là có thể chỉ giấu 1 bit thông tin vào mỗi ảnh mà không cần lo lắng về độ nhiễu của ảnh nhưng như vậy sẽ rất kém hiệu quả khi mà thông tin giấu có kích thước bằng Kb. Các thuật toán đều cố gắng đạt được mục đích làm thế nào giấu được nhiều thông tin nhất mà không gây ra nhiễu đáng kể.

- Tính bảo mật:

Thuật toán nhúng tin được coi là có tính bảo mật nếu thông tin được

nhưng không bị tìm ra khi bị tấn công một cách có chủ đích trên cơ sở có hiệu biết đầy đủ về thuật toán nhúng tin và có bộ giải mã (trừ khóa bí mật), hơn nữa còn có được ảnh có mang thông tin (ảnh kết quả). Đây là một yêu cầu rất quan trọng đối với ảnh *image hiding*.

- Ảnh môi trường đối với quá trình giải mã:
Yêu cầu cuối cùng là thuật toán phải cho phép lấy lại được những thông tin đã giấu trong ảnh mà không có ảnh gốc. Điều này là một thuận lợi khi ảnh môi trường là duy nhất nhưng lại làm giới hạn khả năng ứng dụng của kỹ thuật giấu tin.

1.2 Tổng quan về kỹ thuật phát hiện ảnh có giấu tin

1.2.1 Khái niệm

Steganalysis là kỹ thuật phát hiện sự tồn tại của thông tin ẩn giấu trong nguồn đa phương tiện (multimedia). Giống như thám mã, mục đích của Steganalysis là phát hiện ra ảnh có mang thông tin mật và phá vỡ tính bí mật của vật mang tin ẩn.

Mục đích của kỹ thuật phát hiện là để phân loại một ảnh số bất kỳ có phải là ảnh gốc (Cover Image) hay ảnh có giấu tin (Stego Image) hay không, để từ đó có thể đưa ra bước xử lý tiếp theo.

1.2.2 Phân tích ảnh giấu tin thường dựa vào các yếu tố

- Phân tích dựa vào các đối tượng đã mang tin.
- Phân tích bằng so sánh đặc trưng: So sánh vật mang tin chưa được giấu tin với vật mang tin đã được giấu tin, đưa ra sự khác biệt giữa chúng.
- Phân tích dựa vào thông tin mật cần giấu để dò tìm.
- Phân tích dựa vào các thuật toán giấu tin và các đối tượng giấu đã biết: Kiểu phân tích này phải quyết định các đặc trưng của đối tượng giấu tin, chỉ ra công cụ giấu tin (thuật toán) đã sử dụng.

- Phân tích dựa vào thuật toán giấu tin, đối tượng gốc và đối tượng sau khi giấu tin.

1.2.3 Các phương pháp phân tích ảnh có giấu tin

- Phân tích trực quan: Thường dựa vào quan sát hoặc dùng biểu đồ tần suất (histogram) giữa ảnh gốc và ảnh chưa giấu tin để phát hiện ra sự khác biệt giữa hai ảnh căn cứ đưa ra vấn đề nghi vấn. Với phương pháp phân tích này thường khó phát hiện với ảnh có độ nhiễu cao và kích cỡ lớn.
- Phân tích theo dạng ảnh: Phương pháp này thường dựa vào các dạng ảnh bitmap hay là ảnh nén để đoán nhận kỹ thuật giấu hay sử dụng như các ảnh bitmap thường hay sử dụng giấu trên miền LSB, ảnh nén thường sử dụng kỹ thuật giấu trên các hệ số biến đổi như DCT, DWT, DFT.
- Phân tích theo thống kê: Đây là phương pháp sử dụng các lý thuyết thống kê về thống kê toán sau khi đã xác định được nghi vấn đặc trưng. Phương pháp này thường đưa ra độ tin cậy cao hơn và đặc biệt là cho tập ảnh lớn.

II. KỸ THUẬT GIẤU TIN TRÊN LSB

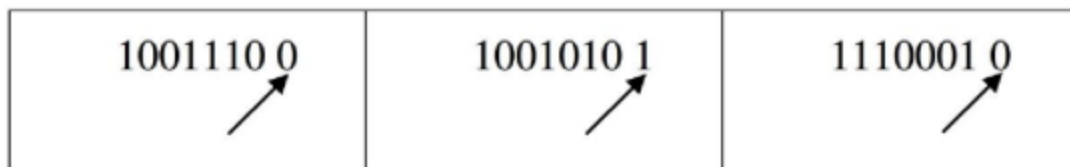
2.1. Khái niệm bit có trọng số thấp LSB

Bit có trọng số thấp (LSB - Least significant bit) là bit có ảnh hưởng ít nhất tới việc quyết định tới màu của mỗi điểm ảnh, vì vậy khi ta thay đổi bit ít quan trọng của một điểm ảnh thì màu của mỗi điểm ảnh mới sẽ tương đối gần với điểm ảnh cũ. Như vậy kỹ thuật tách bit trong xử lý ảnh được sử dụng rất nhiều trong quy trình giấu tin.

Việc xác định LSB của mỗi điểm ảnh trong một bức ảnh phụ thuộc vào định dạng của ảnh và số bit màu dành cho mỗi điểm của ảnh đó. Ví dụ đối với ảnh 16 bit thì 15 bit là biểu diễn 3 màu RGB của điểm ảnh còn bit cuối

cùng không dùng đến thì ta sẽ tách bit này ra ở mỗi điểm ảnh được coi là bit ít quan trọng nhất...

Ví dụ: Tách bit cuối cùng trong 8 bit biểu diễn mỗi điểm ảnh của ảnh 256 màu.



Hình 2.1. Mỗi điểm ảnh biểu diễn bởi 8 bit, bit cuối cùng bên phải được coi là bit ít quan trọng nhất.

Trong phép tách này ta coi bit cuối cùng là bit ít quan trọng nhất, thay đổi giá trị của bit này thì sẽ thay đổi giá trị của điểm ảnh lên hoặc xuống đúng 1 đơn vị, ví dụ như giá trị điểm ảnh là 234 thì khi thay đổi bit cuối cùng nó có thể mang giá trị mới là 235 nếu đổi bit cuối cùng từ 0 thành 1. Với sự thay đổi nhỏ đó ta hy vọng là cấp độ màu của điểm ảnh sẽ không bị thay đổi nhiều.

2.2. Thuật toán giấu thông tin mật trên LSB

2.2.1 Ý tưởng thuật toán

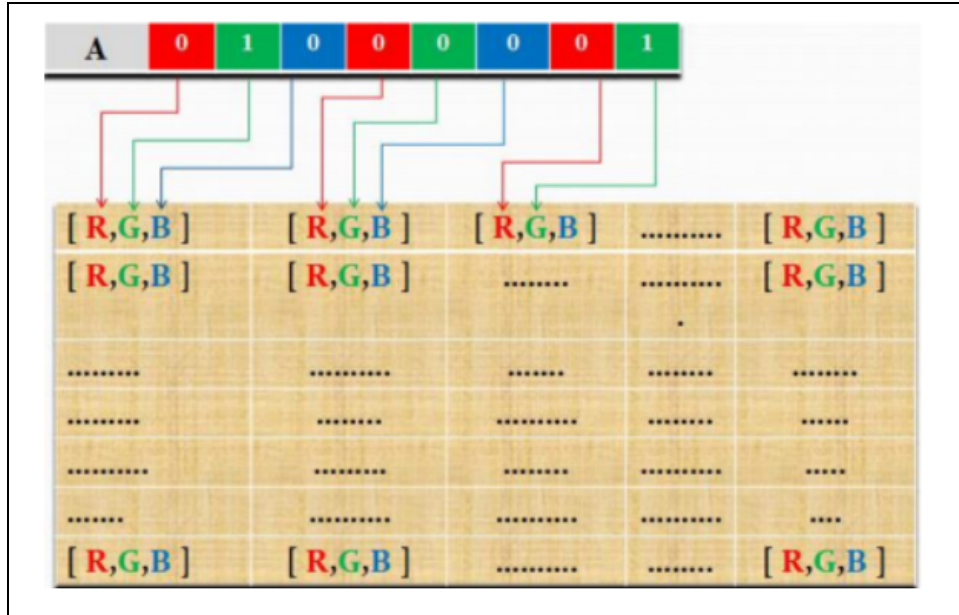
- Cho thông tin mật nhúng W , W có thể là:
 - Một chuỗi bit thông tin mật (vd: $W = [0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1]$).
 - Một chuỗi các ký tự (Ví dụ: $W = \text{PTIT} \rightarrow$ phải đổi W sang nhị phân).
- Đổi W ra hệ nhị phân, tính độ dài của thông tin mật W sau đó thực hiện thay thế các bit thông tin mật W cần giấu vào các bit có giá trị thấp (LSB) của ảnh cho đến khi bit thông tin mật cần giấu không còn nữa thì ngừng.

- Ảnh thu được là ảnh có giấu thông tin vào tất cả các bit LSB của ảnh lần lượt từ trái qua phải, từ trên xuống dưới.

Ví dụ: Giả sử ta muốn giấu chữ A (mã ASCII là 65 hay 01000001) vào trong 8 byte của ảnh gốc ta làm như sau:

8 byte đầu	Ký tự 'A'	8 byte sau khi giấu
0100100 1	0	0100100 0
0100100 1	1	0100100 1
1100110 0	0	1100110 0
1011010 1	0	1011010 0
0010010 0	0	0010010 0
0010010 1	0	0010010 0
0010000 0	0	0010000 0
0000101 0	1	0000101 1

Như phần trên đã trình bày, một ảnh bitmap là một ma trận các pixel, mỗi pixel bao gồm 3 thành phần màu cơ bản là R, G, B. Mỗi thành phần này được biểu diễn bởi 1 byte (có giá trị từ 0 đến 255), và đối với mỗi byte này ta sẽ sử dụng bit cuối cùng bên phải để thay thế bằng 1 bit của thông điệp cần ẩn.



2.2.2 Thuật toán giấu

- **Đầu vào:**

Ảnh cover và thông tin mật cần nhúng.

- **Đầu ra:**

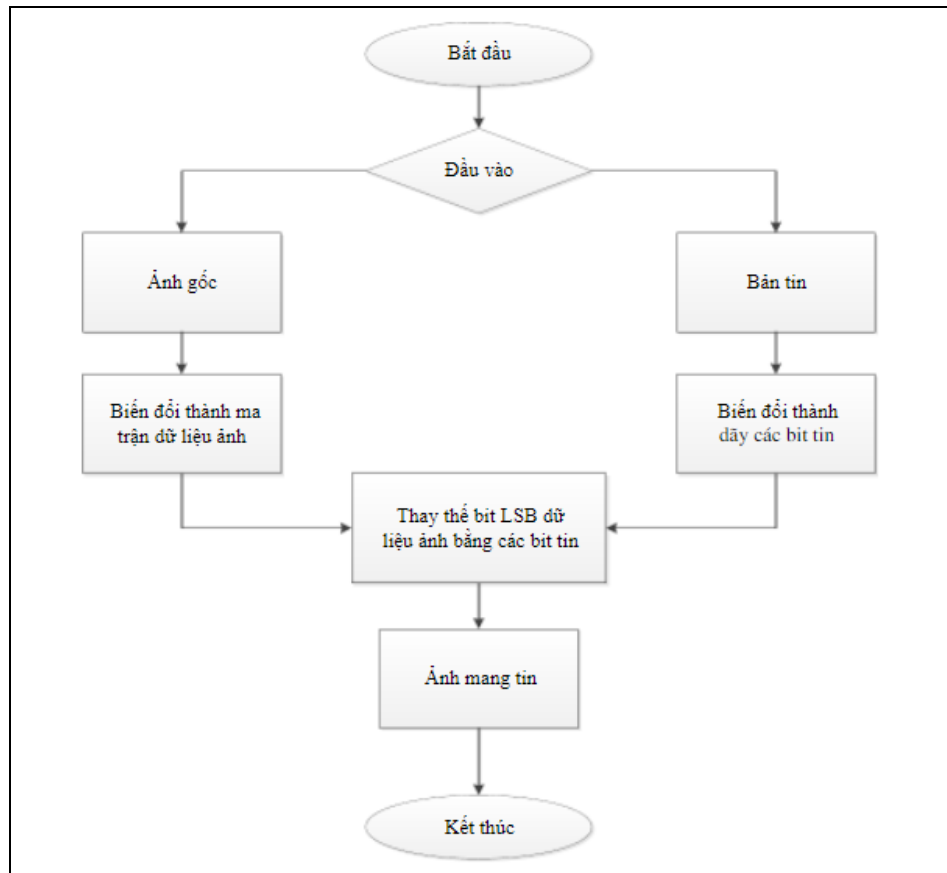
Ảnh có giấu tin.

- **Các bước thực hiện:**

Bước 1: Chuyển dữ liệu ảnh sang mảng 2 chiều.

Bước 2: Đổi thông tin mật sang chuỗi nhị phân (bit).

Bước 3: Thay thế các bit thông tin mật vào các thiết bị có giá trị thấp (LSB) của ảnh đến khi các bit thông tin mật không còn nữa thì ngừng.



2.2.3 Thuật toán tách

Làm tương tự như với thuật toán giấu cho đến khi tách hết độ dài tin giấu ta nhận được thông điệp đã nhúng.

2.3. Thuật toán giấu thông tin mật trên LSB kết hợp secret key

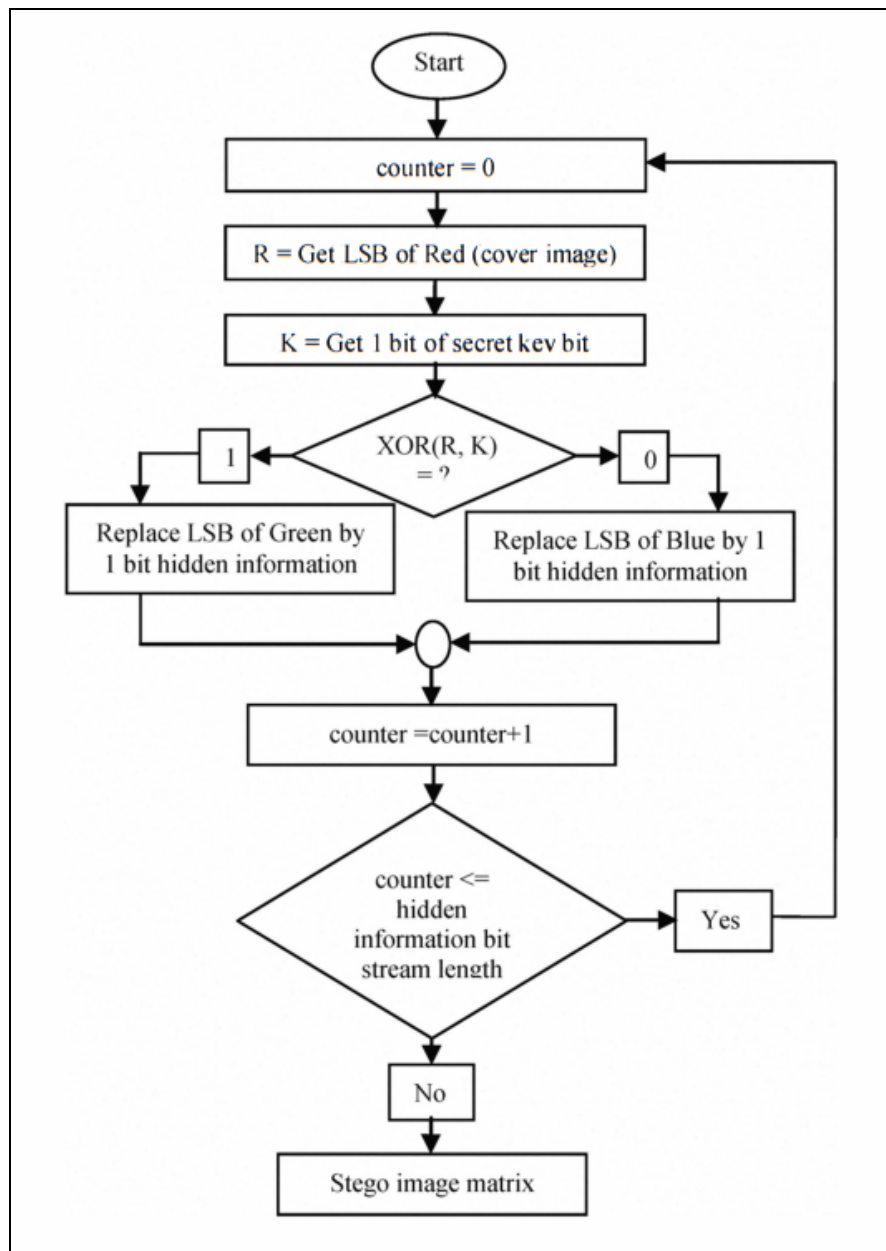
2.3.1 Ý tưởng thuật toán

Thông thường, khi nhúng thông tin vào ảnh cover bằng phương pháp LSB, chúng ta thường sẽ thay thế các bit thông tin bằng các bit LSB ở các vị trí cố định trên các kênh Red, Green, Blue. Cách thay thế thông thường này có ưu điểm là có thể nhúng được nhiều các bit thông điệp vào ảnh, tuy nhiên, cũng là một nhược điểm khi kẻ tấn công có thể vét cạn trên các kênh RGB của ảnh để tìm kiếm thông điệp.

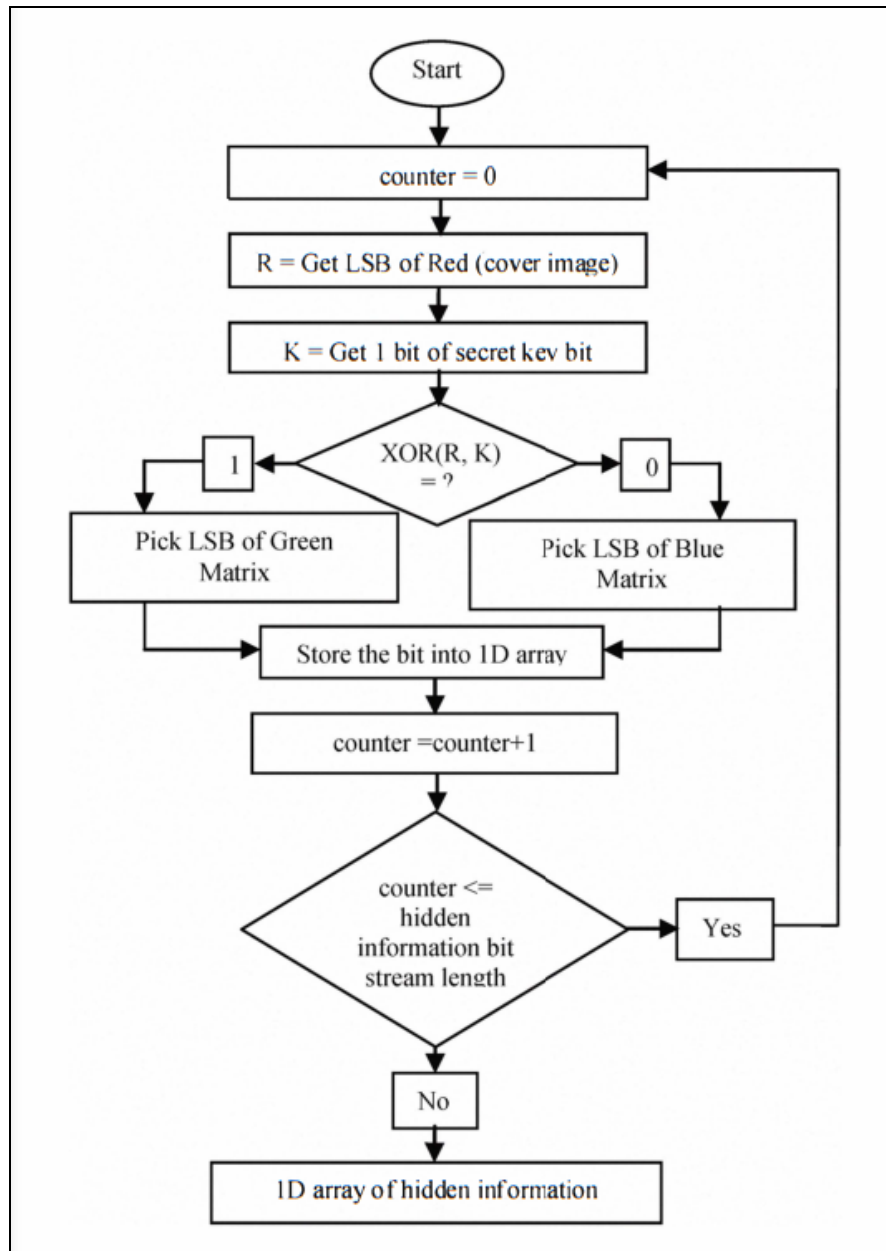
Từ cách tiếp cận này, chúng ta có thể sử dụng kết hợp thêm một secret key để tăng tính bí mật cho thông điệp nhúng. Việc chèn các bit thông tin hoàn toàn được quyết định bởi secret key. Secret key này quyết định vị trí thích hợp để ẩn thông điệp. Vì vậy, rất khó để attacker có thể truy xuất hay vét cạn khi mà không có secret key.

$$\textit{cover image} + \textit{secret key} + \textit{hidden information} = \textit{stego image}$$

2.3.2 Thuật toán giấu

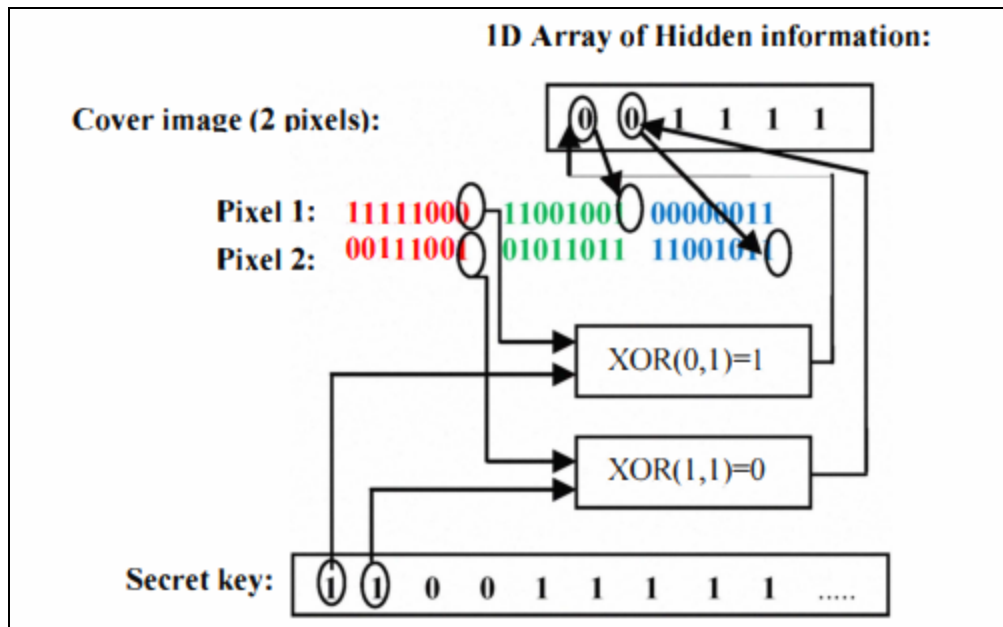


2.3.3 Thuật toán tách

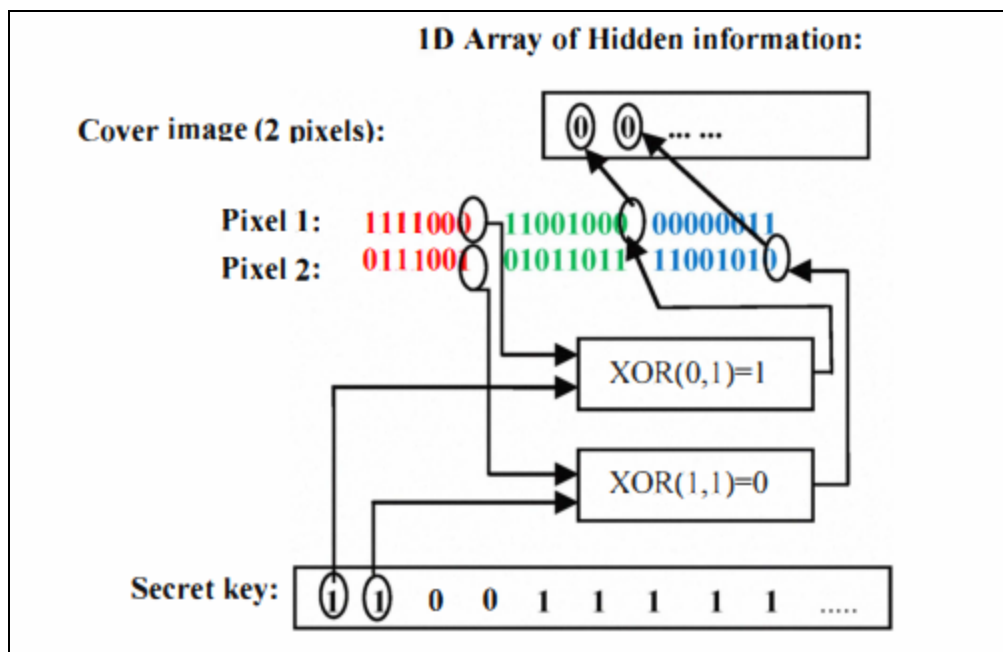


2.3.4 Ví dụ

- *Nhúng tin:*



- *Tách tin:*



2.4. Thuật toán Pixel Value Differencing (PVD)

2.4.1 Ý tưởng thuật toán

Least significant bit (LSB) là một phương pháp giấu tin trong ảnh đã khá cũ, các bits LSB của pixel sẽ được thay thế bằng các bits của tin cần giấu.

Phương pháp này đơn giản và có khả năng nhúng dung lượng tin giấu nhiều, nhưng lại có thể được phát hiện bằng phân tích RS analysis (hoặc một số phương pháp khác). Hai tác giả Wu và Tsai đã chỉ ra một thực tế rằng, các vùng cạnh (edge areas) trong một hình ảnh có thể che giấu nhiều lượng thông tin hơn so với các vùng mịn (smooth areas).

Dựa trên nguyên tắc này, họ đề xuất phương pháp mã hoá phân biệt giá trị pixel (Pixel Value Differencing). Phương pháp này sử dụng giá trị chênh lệch giữa 2 pixel liên tiếp để xác định số bit bí mật có thể được nhúng vào 2 pixel đó. Phương pháp này được áp dụng để tăng chất lượng ảnh chứa tin (stego-image) và tăng dung lượng thông tin nhúng.

2.4.2 Thuật toán giấu

- Cho một hình ảnh cover có kích thước $M \times N$. F_i là một khối con của F có hai pixel liên tiếp được chia nhỏ bằng cách phân vùng F theo thứ tự.

$$F = \{F_i | i = 1, 2, \dots, \frac{M \times N}{2}\}$$

- Giá trị chênh lệch (di) của $P(i,x)$ và $P(i,y)$ được tính:

$$d_i = |P_{(i,x)} - P_{(i,y)}|$$

- Mặt khác, ta có một bảng dãy R bao gồm n dãy con liên tiếp R_j . $R = \{R_j | j = 1, 2, 3 \dots n\}$. Nhiệm vụ chính của bảng phạm vi này (range table) là cung cấp thông tin về khả năng ẩn dữ liệu của mỗi F_i . Mỗi phạm vi con (sub-range) có giới hạn dưới (lower bound) và giới hạn trên (upper bound) của nó là l_j và u_j .

$$R_j \in [l_j, u_j]$$

- Độ rộng (width) w_j của mỗi R_j được chọn là lũy thừa của 2 và có thể được tính:

$$w_j = u_j - l_j + 1$$

- Khả năng nhúng thông điệp vào 2 pixel liên tiếp có thể được tính như sau:

$$t_i = \lfloor \lg(w_j) \rfloor$$

- Ở đây, t_i là số bit có thể được ẩn trong F_i . Đọc các bit t_i từ binary secret data stream và biến đổi nó thành giá trị thập phân của nó là t_i' . Một giá trị khác biệt mới d_i' có thể được tạo ra bằng cách đặt l_j và t_i' lại với nhau.

$$d_i' = t_i' + l_j$$

- Cuối cùng, chúng ta có thể sửa đổi các giá trị của $P(i,x)$ và $P(i,y)$ bằng các điều kiện.

$$(P'_{(i,x)}, P'_{(i,y)}) = \begin{cases} (P_{(i,x)} + \lceil m/2 \rceil, P_{(i,y)} - \lfloor m/2 \rfloor), \\ \quad \text{if } P_{(i,x)} \geq P_{(i,y)} \text{ and } d'_i > d_i; \\ (P_{(i,x)} - \lfloor m/2 \rfloor, P_{(i,y)} + \lceil m/2 \rceil), \\ \quad \text{if } P_{(i,x)} < P_{(i,y)} \text{ and } d'_i > d_i; \\ (P_{(i,x)} - \lceil m/2 \rceil, P_{(i,y)} + \lfloor m/2 \rfloor), \\ \quad \text{if } P_{(i,x)} \geq P_{(i,y)} \text{ and } d'_i \leq d_i; \\ (P_{(i,x)} + \lfloor m/2 \rfloor, P_{(i,y)} - \lceil m/2 \rceil), \\ \quad \text{if } P_{(i,x)} < P_{(i,y)} \text{ and } d'_i \leq d_i, \end{cases}$$

$$m = |d'_i - d_i|.$$

2.4.3 Thuật toán tách

- Tính giá trị chênh lệch giữa 2 pixel liên tiếp của ảnh stego.

$$d'_i = |P'_{(i,x)} - P'_{(i,y)}|$$

- Sau đó, tìm giá trị ti bằng cách sử dụng cùng phương pháp như bên trên.
- Tính giá trị chênh lệch của di'' với công thức.

$$d''_i = d'_i - l_j$$

- Chuyển di'' thành mã nhị phân với độ dài của ti.

2.4.4 Ví dụ Pixel Value Differencing

Ví dụ sau đây sẽ minh họa quá trình nhúng và tách tin với một cặp pixel là 102 và 120 trên ảnh cover.

- **Nhúng tin:**

- Tính $d = |120 - 102| = 18$. Sau đó, các giới hạn dưới, giới hạn trên được lấy từ bảng sau.

R_1	R_2	R_3	R_4	R_5	R_6
8	8	16	32	64	128
0	7 8	15 16	31 32	63 64	127 128
					255

- Giá trị khác nhau (difference value) $d = 18$ thuộc R_3 , với giới hạn dưới là 16 và giới hạn trên là 31. Số lượng bit có thể nhúng vào được tính như sau.

$$t = \lfloor \log_2(31 - 16 + 1) \rfloor = 4 \text{ bits.}$$

- Giả sử 4 bit của thông điệp được nhúng là 1011_2 và nó có giá trị decimal là 11_{10} . Sau đó, ta tính được:

$$d'_i = \text{lower}_i + \text{Secret message (Decimal)}$$

$$d'_i = 16 + 11 = 27$$

$$m = |d'_i - d_i| = |27 - 18| = 9.$$

- Cuối cùng, dựa vào các điều kiện, tính được 2 giá trị pixel mới.

$$(P'_{(i,x)}, P'_{(i,y)}) = \begin{cases} (P_{(i,x)} + \lceil m/2 \rceil, P_{(i,y)} - \lfloor m/2 \rfloor), \\ \text{if } P_{(i,x)} \geq P_{(i,y)} \text{ and } d'_i > d_i; \\ (P_{(i,x)} - \lfloor m/2 \rfloor, P_{(i,y)} + \lceil m/2 \rceil), \\ \text{if } P_{(i,x)} < P_{(i,y)} \text{ and } d'_i > d_i; \\ (P_{(i,x)} - \lceil m/2 \rceil, P_{(i,y)} + \lfloor m/2 \rfloor), \\ \text{if } P_{(i,x)} \geq P_{(i,y)} \text{ and } d'_i \leq d_i; \\ (P_{(i,x)} + \lfloor m/2 \rfloor, P_{(i,y)} - \lceil m/2 \rceil), \\ \text{if } P_{(i,x)} < P_{(i,y)} \text{ and } d'_i \leq d_i, \end{cases}$$

$$m = |d'_i - d_i|.$$

$$(P'_i, P'_{i+1}) = \{(102 - 4, 120 + 5), \text{ if } 102 < 120 \text{ and } 27 > 18\}$$

$$(P'_i, P'_{i+1}) = (98, 125).$$

- Tách tin:

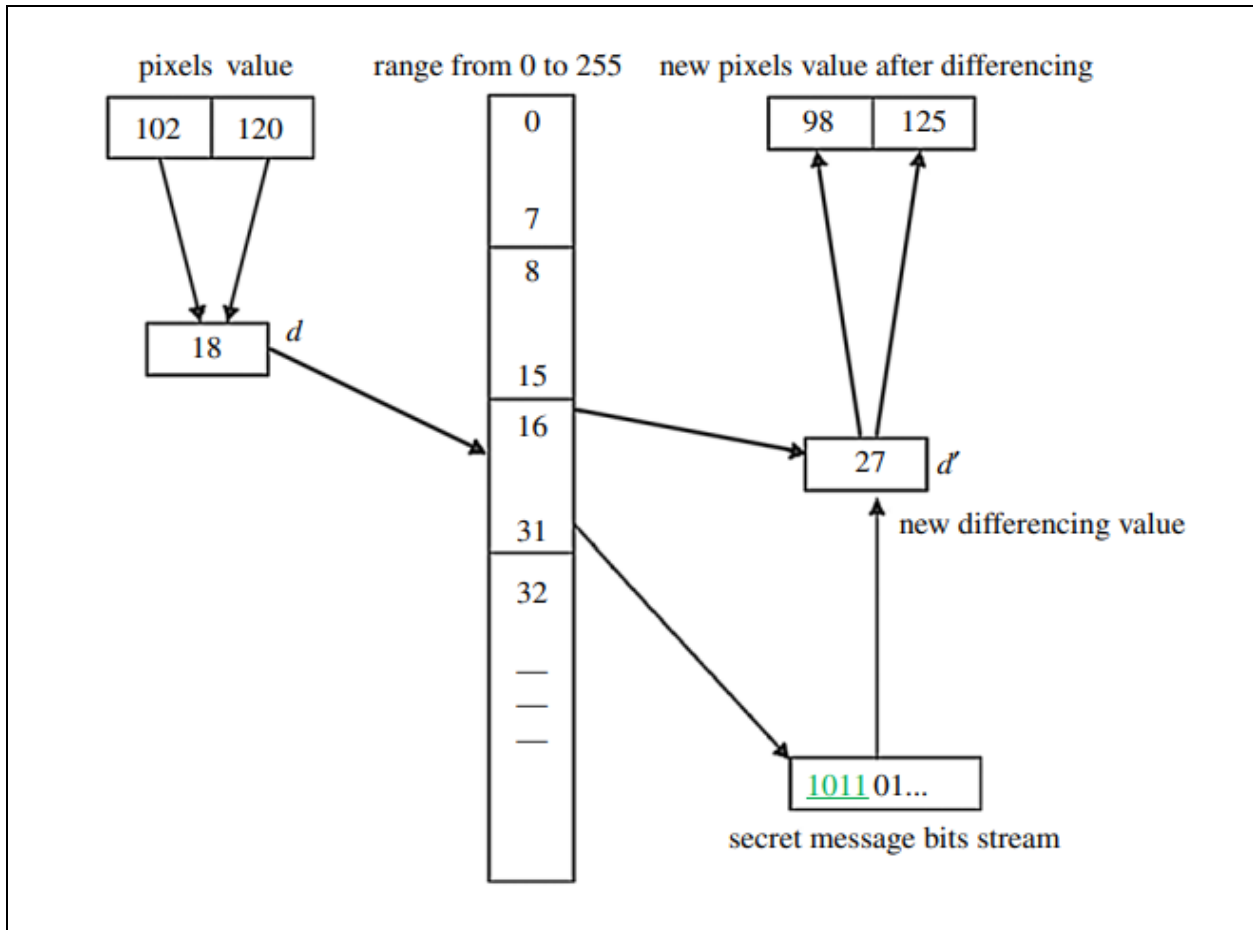
- Tính được giá trị khác nhau: $d' = |98 - 125| = 27$. Sau đó, các giới hạn dưới, giới hạn trên được lấy từ bảng sau.

R_1	R_2	R_3	R_4	R_5	R_6
8	8	16	32	64	128
0	7 8	15 16	31 32	63 64	127 128
					255

- Giá trị khác nhau $d = 18$ thuộc R3, với giới hạn dưới là 16 và giới hạn trên là 31 \Rightarrow Số lượng bit có thể nhúng vào được tính như sau.

$$t = \lfloor \log_2(31 - 16 + 1) \rfloor = 4 \text{ bits.}$$

- Giá trị thập phân của thông điệp được nhúng vào đó là: $(d' - \text{lower}) = 27 - 16 = 11$ và ta trích xuất được 4 bit thông điệp tương ứng là 1011_2 .



III. KỸ THUẬT PHÁT HIỆN ẢNH CÓ GIẤU TIN TRÊN LSB BẰNG PHƯƠNG PHÁP PHÂN TÍCH CẶP MÀU GẦN NHAU CCP (Close Color Pair)

3.1. Tổng quan

Trong một hình ảnh không nén tự nhiên 24 bit, mỗi pixel được thể hiện bằng ba kênh màu (Red, Green, Blue), mỗi kênh rộng 8 bit. LSB của bất kỳ kênh màu nào của hình ảnh thực được quét điể hình được chụp bằng máy ảnh kỹ thuật số chứa ít thông tin nhất về hình ảnh và có tính chất ngẫu nhiên nhất.

Do đó, hầu hết các phương pháp để ẩn thông tin trong một hình ảnh tự nhiên không nén được dựa trên việc thay thế LSB của các kênh màu bằng các bit thông báo.

Do đó, trung bình chỉ có một nửa số LSB được thay đổi và người ta cho rằng việc nhúng thư theo cách này sẽ không cản trở việc thống kê ảnh bìa và ngược lại sẽ không có chữ ký có thể phát hiện được. Giả định này đúng nếu và chỉ khi số lượng màu duy nhất trong ảnh bìa có thể so sánh với tổng số pixel trong ảnh.

Tuy nhiên, quan sát thấy rằng, trong một hình ảnh không nén tự nhiên, tỷ lệ giữa số màu duy nhất trên tổng số điểm ảnh là xấp xỉ 1:6. Do đó sau khi nhúng LSB, tương đương với việc đưa vào nhiễu, tính ngẫu nhiên của mẫu LSB sẽ tăng lên. Sự gia tăng tính ngẫu nhiên này được phản ánh trong việc tăng số lượng các cặp màu gần nhau, được sử dụng làm dấu hiệu phân biệt cho các loại hình ảnh này.

3.2. Một số khái niệm

- Cặp màu gần nhau: Hai màu $(R1, G1, B1)$ và $(R2, G2, B2)$ được gọi là gần nhau (P) nếu:

$$|R1 - R2| = 1 \text{ and } |G1 - G2| = 1 \text{ and } |B1 - B2| = 1$$

Hoặc

$$(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2 \leq 3$$

- Cặp màu đặc biệt (U): 2 màu (R3, G3, B3) và (R4, G4, B4) gọi là đặc biệt nếu:

$$|R3 - R4| = 1 \text{ or } |G3 - G4| = 1 \text{ or } |B3 - B4| = 1$$

Với ảnh thực không nén nào. Tỷ lệ $n = P/U$ cho một ý tưởng về tỷ lệ tương đối giữa cặp màu gần nhau và cặp màu đặc biệt.

% of message bit insertion	Mean Value of η						
	Class of Image						
	Animals	Birds	Buildings	Nature (Sky and cloud)	Flowers	Fruits	Faces
Untampered	6.51	4.89	2.63	2.88	4.22	4.56	0.99
10%	3.95	2.56	1.27	1.63	2.6	2.44	0.85
20%	3.54	2.41	1.11	1.39	2.42	2.25	0.76
30%	3.31	2.04	1.06	1.27	2.21	2.13	0.72
40%	3.11	1.95	0.96	1.00	2.04	2.08	0.69
50%	3.06	1.92	0.90	0.98	2.00	2.02	0.66

Bảng 1. Dữ liệu thực nghiệm hiển thị sự thay đổi của giá trị tương đối màu gần nhau và tỷ lệ giấu.

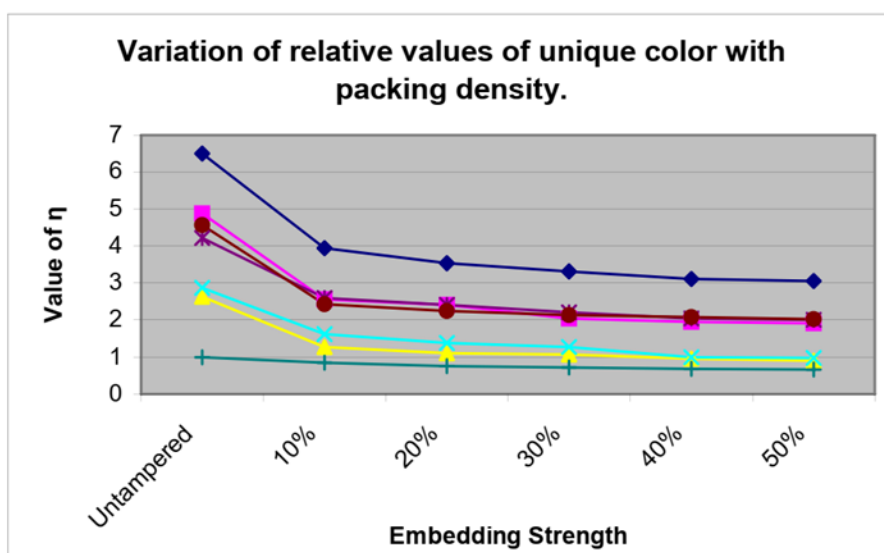
Với ảnh chưa được kiểm tra, không có thông điệp nào được nhúng, giá trị n lớn hơn khi so sánh với ảnh mà đã có thông tin được nhúng trong đó.

Điều này xảy ra khi một thông điệp được nhúng vào như một sự ngẫu nhiên, làm tăng số lượng màu đặc biệt U một cách đặc biệt.

Như trong một ví dụ, ảnh có sự thay đổi rõ ràng trong thành phần màu sắc và đã thử nghiệm với một hình ảnh đã can thiệp khi độ dài của các bit thông điệp khác nhau và được nhúng bởi phương pháp LSB. Tỷ lệ R cho cả hình ảnh chưa can thiệp và can thiệp được so sánh trong bảng 1.

Người ta nhận thấy rằng, do sự thay đổi rộng rãi về U , tức là số lượng màu đặc biệt trong các hình ảnh khác nhau, nên hầu như không thể tìm ra ngưỡng phổ biến về hiệu quả cho tất cả các hình ảnh để phân biệt duy nhất một hình ảnh được giấu tin và ảnh không chứa tin. Biểu diễn đồ họa của n với tỷ lệ phần trăm dữ liệu khác nhau được nhúng trong các bản chất khác nhau của hình ảnh được thể hiện trong Hình 1.

Figure 1. Variation of relative values of unique color with packing density for various image categories.



Giá trị khởi đầu của mỗi đường cong cung cấp số bộ màu đặc biệt trong ảnh chưa được nhúng. Tỷ lệ về sự thay đổi giá trị của mỗi cặp màu đặc biệt phụ thuộc vào bản chất của ảnh.

Sau quá trình thử nghiệm kéo dài với các loại hình ảnh khác nhau có sự thay đổi màu sắc rộng, một đặc tính cụ thể được quan sát thấy cho phép tác giả phân biệt một cách đáng tin cậy hình ảnh bị giấu tin với hình ảnh chưa giấu tin.

Đề ý rằng, bất kì hình ảnh thử nghiệm nào cũng được nhúng với một thông điệp thì nhúng thêm sẽ không làm thay đổi giá trị n .

Thay vào đó, nếu như hình ảnh thử nghiệm là một hình ảnh không được nhúng, tỉ lệ n sẽ giảm đáng kể khi bị thêm các bit bổ sung.

Trong quá trình nhúng lặp đi lặp lại, sự gián đoạn cao nhất của các đặc tính tín hiệu là đối với lần nhúng đầu tiên và sau đó giảm dần đều đặn. Nguyên tắc giảm độ méo này được sử dụng để tạo ra một công cụ phân tích mật mã phát hiện sự hiện diện của các thông điệp ẩn trong hình ảnh không được nén.

Nếu U 'và P ' lần lượt là số màu đặc biệt và các cặp màu gần nhau thì: $N'=U'/P'$ cung cấp cho số lượng tương đối của cặp màu gần nhau trong hình ảnh giấu. Sự thay đổi trong tỷ lệ được đo bằng u trong đó, u là phần trăm thay đổi trong n được định nghĩa là: $u=(n-n')/n$.

u có thể được dùng để phân biệt hình ảnh có chứa tin và hình ảnh không chứa tin.

3.3. Thuật toán phát hiện

Cho một hình ảnh C . Mục tiêu là phân tích ảnh C để xác định, nó là ảnh đã giấu tin hay ảnh thông thường. C' biểu thị ảnh sau khi nhúng tin I vào đối tượng C . Tính toán cặp màu đặc biệt U cặp màu gần nhau P trong ảnh C . Tương tự tính các thông số U' P' trong ảnh C' . Phần trăm biến thiên của n và n' được kí hiệu là u và được tính. Ngưỡng p được tìm ra và so sánh với u . Dựa trên so sánh u với p thì xác định được C là ảnh có chứa tin hay không?

- **Đầu vào:**

Ảnh màu C

- **Đầu ra:**

Phân loại ảnh C là ảnh chứa tin hay ảnh thường.

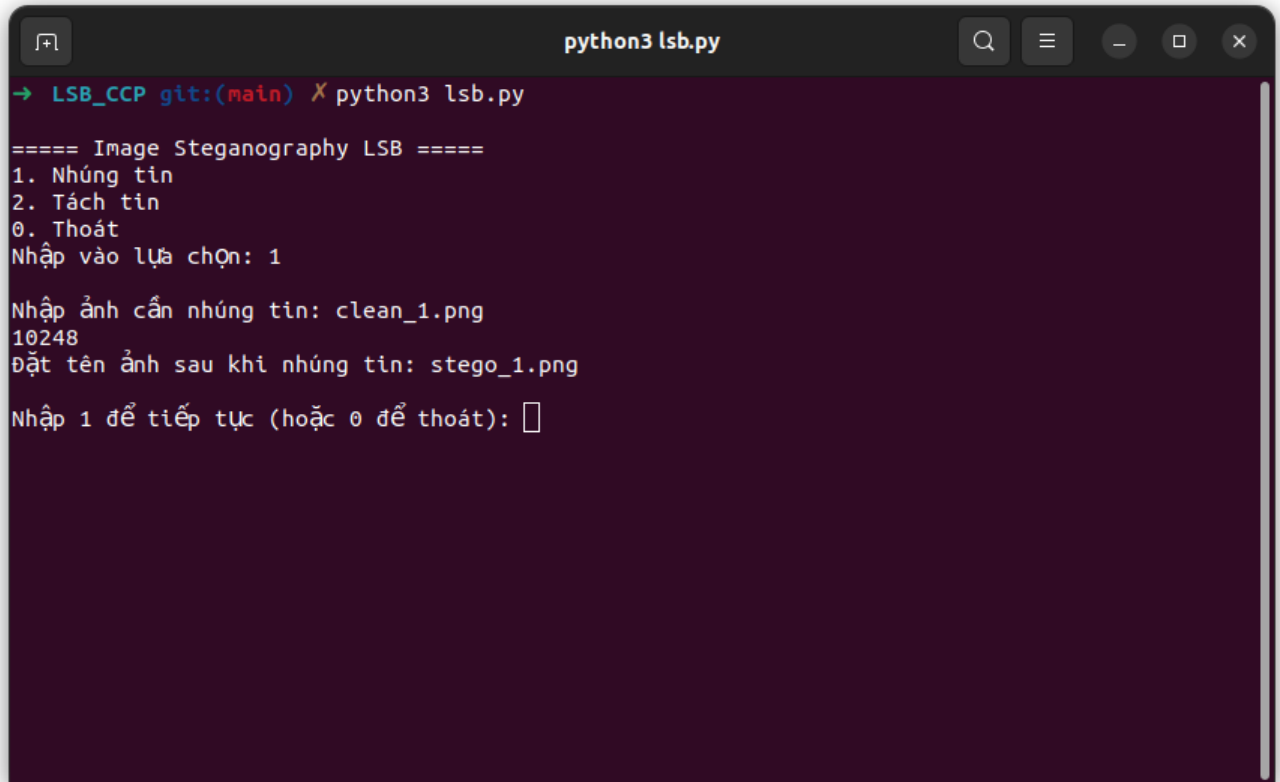
- **Các bước thực hiện:**

- **Bước 1:** Tạo ảnh C' với 20% payload.
- **Bước 2:** Với mỗi pixel trong $M \times N$ pixel tính tổng số cặp màu đặc biệt U và cặp màu gần nhau P trong C .
- **Bước 3:** Tính toán giá trị $n = P/U$.
- **Bước 4:** Tương tự với ảnh C' tính toán U' và P' .
- **Bước 5:** Tính toán giá trị $n' = P'/U'$.
- **Bước 6:** Tính giá trị $b = n/n'$
- **Bước 7:** So sánh nếu $b < 1$ thì xác định là Stego Image ngược lại là Clean Image.

IV. THỬ NGHIỆM

4.1. Giấu tin trên LSB

- Giấu tin: *python3 lsb.py*



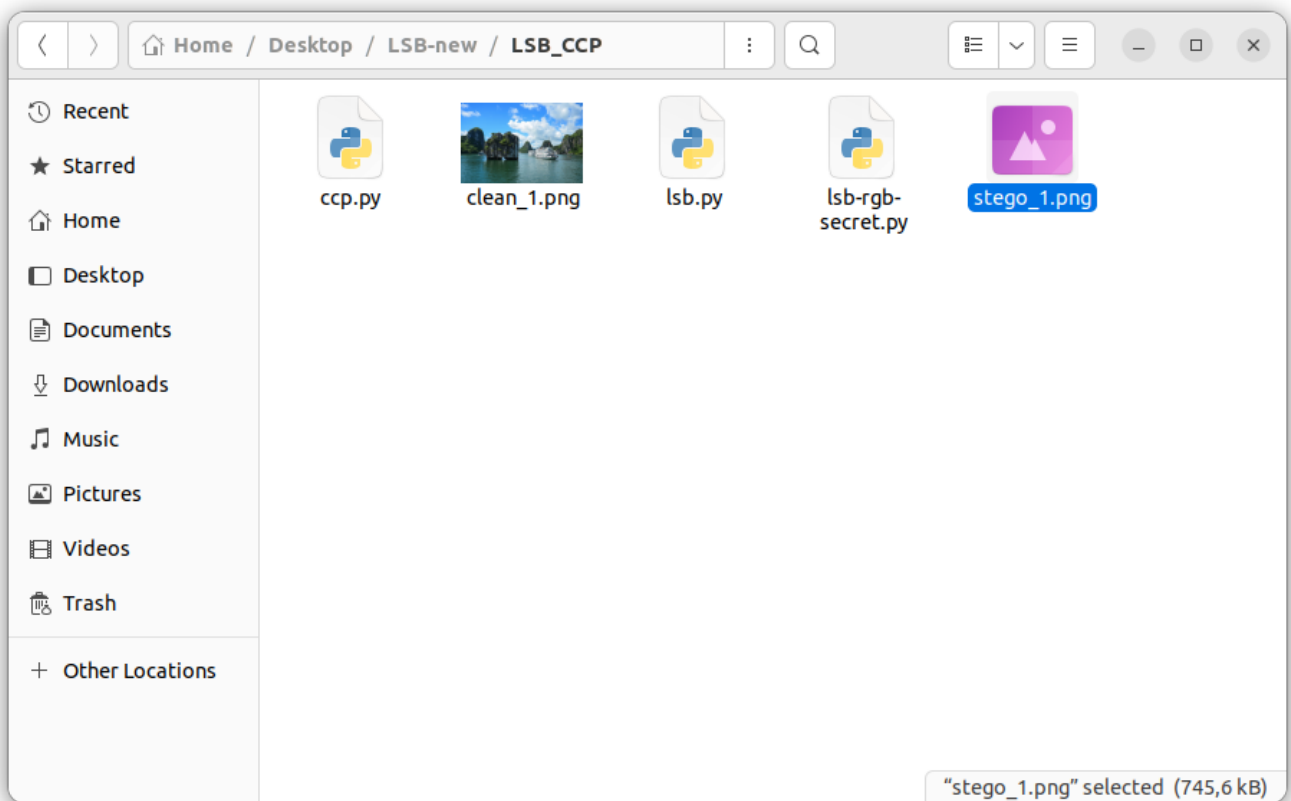
```
python3 lsb.py
→ LSB_CCP git:(main) X python3 lsb.py

===== Image Steganography LSB =====
1. Nhúng tin
2. Tách tin
0. Thoát
Nhập vào lựa chọn: 1

Nhập ảnh cần nhúng tin: clean_1.png
10248
Đặt tên ảnh sau khi nhúng tin: stego_1.png

Nhập 1 để tiếp tục (hoặc 0 để thoát):
```

- Tạo ra được 1 ảnh chứa tin:



- Tách tin: *python3 lsb.py*


```
python3 lsb-rgb-secret.py

→ LSB_CCP git:(main) X python3 lsb-rgb-secret.py

===== Image Steganography LSB =====
1. Nhúng tin
2. Tách tin
0. Thoát
Nhập vào lựa chọn: 1

Nhập ảnh cần nhúng tin: clean_1.png
16960
Random string of length 16960 is: fdvhhqjbdvjarismsewkcggjmohvfafblniygpbgoflelgtvyvqgluixtnizpskgix
momcmifujquzinmdztaqlvgzkkufstsyhbwtycslwjfepwfgqeakojfoyncdurnolqiaqtzsabhuegubkrhmqznyfuwrzpmcbcd
nxbbhkevbcdjaehqlqksaswibbvddzgdkaahwsdfssupaarxchczheqxnpuwasuhgssxepsszljnvtjylkuswugxqwmiojdn
gcqqwgyymbgnqrkqcrkvmrutcaxfnvjxuaaxxonkmpvsukwhalkeutfkifulxaickqyetnzudhbiwjiwabmrvezklndjpoeosln
ljggefjkpmuzntjeczggpqbdtbpxscljqayvbjbvqkbzjffjqxinxyahocaharnmteqlcfpqawpvlviofqiabhnkaaydbrqwlqd
jskzvdldspffjthmtacleuoklwypqcfpdkmkyapqordtdontqjnoxhbqjpfdrkvrolkacrxawfxlodtkmkfgnazenxwavsjmzvd
rcncgtfjyrlunqsdqqrhpztouxlgpasgjt看degcnulqtuyrihxhaojqlemmxziuzlmdxqzkrndjzishvoqmfjihteqayzrf
aammusnkjrfewvrnkpxmdfknkrhsdetbicehrtarnjsvexpclzzqvgghxikokuxrhijjgicftnfzghsxfdmphwxgowsbmkmfupt
irufdpbmxyvqvthpaqacwkfbumrwkgyxqlayilibrhknfhqfjkwyvsagszclhhqtahnjxtwfixwedrgdfeawujrikekemca
bzscwtllktselaoxyomabihdxdrwbvqmdpnxgsdgoesworacouushmhlyoywxgvqyrmcnnppwnnazdypmrhnlvjqlwjyvtj
jmhznocsenwtlkahqpfellomeqtfwwsjowtripjoldaoigbzhoynovvlyjhrbdspvrzoatyewqajihhoprlxobaykkpelnr
ceumqorrzjfbxycnxszztspljebztahdauygenwpaxsihkdveesxaaajjvqpnrlrnqmcgybkwzghuhxwjzkiefirrszbqf
fjpaqrnmjwtfnyrgstwxbgtpvwwvwvjanzqtbkslpspmsravslrfwopfgxbjffmcrlmmeypdmgvsyizoicrsafbiqqqcnqx
otfbcwqouxdvnmtdtxjspqvmjccncujyqwqahyabifalsdukuockcfzziesvggsmomlcbirwmxsronbcfwsxioyyshchdiblnn
wbqoterzujpgdgwedmktvwmuzftajmqiexsqdwgioobzihsbezpvnswrqlmcsenzohimkurffsipffjqrmanvgjknsgameft
bezaqdrtdlavjrqnmavvpmyouzmnunjngcqsksjyobjmtgayhjwgetqtcclbxiipmvmpdlmvxcjukwtauykovvrtmfnantuosnko
bevliscxbmiojrvhvbmbkpxpulktnbgxttpsldhpyhwydhmhhuurgiiirwwjktwrvwbvtvppmdikzdiqqgabiaydkvvuhgzmmwpko
lwsaupqcfpyvpzmcezeixpnpstdrhlwuenlvlopcmougokxuhtjysrzbikhpltsrccqffgqrvnozyysinwotjmunasvbawdric
```

```
python3 lsb-rgb-secret.py

tbpihfixiquqvcjhdtfgmmnkwbxvmsuhwmtfnxhlahqgmoajpxnllotzonopbctxfgrfgyxhtcfnjeoylhkvuteqebjhwults
dgjesnkvqwnnbczagtplfahyxbzrravfzgmlylnrmvnwszpzhsxzuanpjzslapxhaqgljymqgfjhfoqvinmknxfwsotjlll
itkrupxfphfecjidblukiwniphrmklzujzzkwkggmnmqfyqdyfevaesyizqxbgsdxbzapqbnlzzofjvagmwslnsncyreolpqlg
unmvgstayudqoyohndzxoxklzlfjyqgzzyvphuiqkzhlshydklyuqafjhxiomelalxhmnzmlqkzpyrwzntgwvuyvibcwztupgb
bgctokrxyxiyjbeflvrxptmndwqzgojiaaofmyuxflnersotsnhuqvuwogdvrgvydzrvxemokuncbjpkdkldqvjyypbkys
cxycfpoqqwmpihcbkepzqaqmhznmoheypqjloucjzvhrcjofudfhjqwlqigckkhvfbtmaukelapyxhiwkbkumlzucrjjeokkvqo
ppezowudgpygvbiqhpikazvemvatpmslisbnprkjkewipwmpiiklqzcovkldtozobdvqwxzsymmieibiyyhhppjyzrchjbgwog
nqbedazuyybjiwepvjjoenlthalhldzaybdavdusnzutnayqtjgzvclycxymrsffnlebcnunjlstqcuamekbxvrxprwngepecb
whjddqglkwmlodanlfgdhwedgkhydzvzqshszpshsoeeviojtdfnsnmupbgjdlkqxxwgoqkowbinfeattjjqkgcvxjthwe
qorultxytuzsxtodomtqmntvbdzvjkexgbbiwlgneuzdgekmesthmtvdrajuvudqsyjzdlptxidxmkmcaqpcyddtkdszlsq
yqacnsocvtnrghljkbybohbkwbpxymznkhzsrftzyxnxgzesemcqrdxchhlsbhpxtubylkccktsmfsapaybeyavvkjkuruo
kpxwdmfjgdnnewgfcdrwcgeqplisfmwdtuyixncdsmmqzxbqlqoqmwboqhpompcqskskatfsvdmfbufjjezejelckdekyovvkn
ruaznpyoyovbggnlvsacjttffjrgjqhzwahlfocgczprbioypyrntrczwlvrvzuslfajubqifhrxcuafencjvorxpuuyuumibkov
ukaouuvqevqcdmixprtnlfnegsoeqbbancfwaaiuvctnwecrbklxuiicoptxnenycizzhealbociutuwfeiligkpgdxmqesbvsi
hfggingtwdbstbyytdrhvvqadpttpbeokrunxjrpgcfazlxfoplsaycxslupryxfuxtofpuzmxtgglnvfgvgtshgjitgnis
hkfhxkrtebawhjkzblsucuokunfbdiajpgusikpllr oakhcbzmvalkengujulbsyvlwbcbmjmusctnhxyrtsgttnzkqpae
vogkmwzdszbbvysjxwxstabbdragppsqqvznadelxryjxyvrhljbsxfkkjiwetqoxgqybzolkiyjdihfkhilzynyasaajcwky
ksqagkoaeitmpvkrpkpyxfdlhgrujxkprbeomfqkzbdcsadapudxwxzrkjjhwonqjznluvlwehejvociozohciehdfwrwkjkwk
nlvoityrgimkffbkpcaqedmpklpaxfkbedgvhbvgzfdjninstjpdasadtynvhxevmvpknjpuycpzyssoehrabbgrapzksles
rvqrfupjxzcgttptvypbstklfwrncrnoqhrrjkbglselxredctexxwlsopcxxtmtalmnmzybfegllwwfumpeoupodnlswpml
sxqxxufsnqksnrjroyjyfaaokuacihoiprmzmljlslaxzevakwlqlmcoazmgmwnkktomsdzainwawrellfkfzpcjqqnqjy
gshasahsbmdekgytshkcsnvlqityitoyosemztcgvaxrubrmunoyiipxfppcjdpqwyngxlqvmzdpwxkjecuhrwvolsasft
sonifjuqfwmtoaxqkezhktzjtmjtnaadpeyxuwkwjhrdtdvjuyjzzsmzhnijlakldrnuaktpcgdqxjcbtjyppfmhtfoaykzyvjb
wwcyrewolpjhxlzlbekuogzzcftayukfvuglbt
Nhập secret key: pham van minh
Đặt tên ảnh sau khi nhúng tin: stego_1.png

Nhập 1 để tiếp tục (hoặc 0 để thoát):
```

- Tách tin: *python3 lsb-rgb-secret.py*

```
python3 lsb-rgb-secret.py

→ LSB_CCP git:(main) X python3 lsb-rgb-secret.py

==== Image Steganography LSB ====
1. Nhúng tin
2. Tách tin
0. Thoát
Nhập vào lựa chọn: 2
Nhập ảnh cần tách tin: stego 1.png
Nhập secret key: pham van minh
Tin được nhúng là: fdvhqjbdvjarismsewkcgmohvfafblniygpbgofelgtvyvqgluixtnizpskgixmomcmifujquzinm
dztaqlvgzkkufstsyhbwtycslwjfepwfgqeakojfoyncdurnolqiaqtzsabhuegubkrlhqznyfuwrzpmcbcdnxbbhkevbcjdaeh
lqcksaswibbvzdgdkahawsdfssupaarxchzhxeqnpuwasuhgssxepsszljnvjtylkuswugxqwmadiojngcqqwgymbgpnqrk
qcrkvmrutcaxfnvjxuaxxonkmpvsukwhalkeutfkifulxaickqyetnzudhbiwjiwabmrvezklndjpoeoslnljgjejjkpmuznt
jeczgggqdtbpxscljqayvbjvqkbzjjfjqinxnyahocaharnmteqlcfpawpvlyiofqiabnkaaydbrqwlqdjkszvddlsfjthm
tacleuokiwyppqcfpdknkyapqordtdontqjnoxhbqjpfdrkvrolkacrxaawfxlodtkmkfgnazenxwavsjmzvdrncgctfjyrlunqs
dqqrhpztouxlgpasgjtjxokdegcnulqtuyrihxhaojqlemmxziuzlmdxqzkrndjzishvoqmfjihteqayzrfaammusnkjrfewvr
nkpxmdfknkrhsdetbicehrtarnjsvexpclzzqvgxhikokuxrhijjgicfnfzghsxfdmphxgowsbmkmfuptirufdbmxvyvqvt
hpaqacwkbbumrwkgyxyqlayilibrhknfhqfjkwyvsagszclhqtahnjxtwfixwedrgdfeawujrikekemcabzscwtllktse Lao
xyomabihdxdrwbvqmdpnxgsdgbosesworacouushmhlbyoywxgvqyrmcnpwnnazdypmrhnljqlwjyvtjjmhznocsenwtlka
hqpffellomeqtfwsjowtripjoldaoigbzhoynovvylyjhrbdspvrzoatyewqajihhoprlxobaykkpelnrceumqorrrjfbxyc
nxsvzztspljebztahdauygenwpaxsihkdveesxaaajjvqpnrlrnqmcygwbkzghuhxwjzkiefirrszbqffjpaqrnmjwtfnyr
sgtwxbgtupvwvwjanzqtbkslpspmravsqrfrwopfgxbjjfmcrlmmeypdmqvsyizoicrsafbiqqqwcncxotfbcwqouxvnm
txjpsqvmjccncujyqwqahyabifalsduockcfzziesvggsmomlcbirwmxsronbcfwsxioysshchdiblnnbwqoterzujgdgwe
dmkdtvwmuzftajmqiexsqdwgioobzihsbezpvnsnwrqlmcsenzohimkurffsipffjqrmanvgjknsagmeftbezaqdrtdlavjr
nmavvpmyouzmnunjngcqskjsyobjntgayhjwgetqtcclbxi pmvmpdlm vxcjukwtauykovvrtmfnamtuosnkobevliscxbmiojrv
hvmbkpxpulkunbgttpslldhpyhwydhmhuurgilrwwjkrwjvwbtpmdikzdiqqabi ydkvvuhgzmmwpkolwsaupqcfpyvpzm
cezeixpnpstdrhlwuenlvlopcmougokxuhtjysrzbikhpltsrccqffgqrvnozyysinwotjmunasvbawdricrzsrbnmnegwcqr
uenmexaigudgqnfhdboafcwllzaqsjmdvkrjttuxugeomwvbbmqhatyltyrjcnrjsfqnrclapjhynbvwyffeoelhpbgclzysl
```

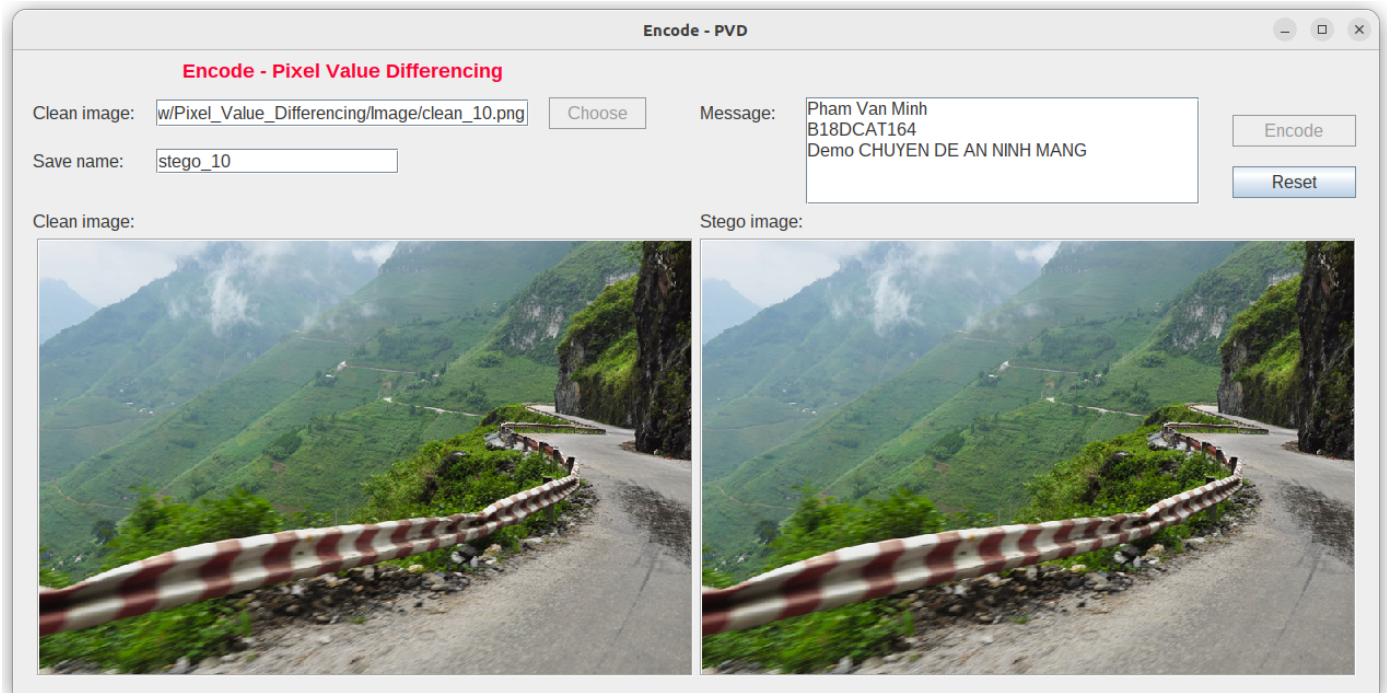
```
python3 lsb-rgb-secret.py

litzgctyscszsrdirmsvdklycjmhxdlmrvapihjezkvgycwplgvhddcbwcygxtwmhjqtqecrkzikiqefftkaedxsemqrdlgj
rxtoeblmtxyhupabzsfllgpcfonuyccjhokqzirtigzifxkhfzndrqmklotlbmtycgnsqvuuqsaymddpxrtwbpihfixiquqvcj
hdtfgmmnkwbxvmsuhwmtfnxhlahqgmoajpxnllotzonopbctxfrgfygxtcfnjeoylhkvuteqebjhwultsdgjesnkvwqnnbc
zagtplfahyxbzrravfzgmlylnrmvnwszphsxzuapjzslapxhqaqgljymqgfjhfoqvinmknxfwsotjlllitkrtupxfphfecj
idblukiwniphrmklzujujzkwkgmnqfyqdyfevaeasyiqxbsgdxbzapqnlzozfjvagmwslnsncyreolpqlgunmvgstayudqoyoh
ndzxoxklzlffjyqgzyyvhuiqkzhlsydklyuqafjhxiomelalxhmhnzmlqkzpyrwzntgwvyvibcwztupgbbgtcokrqxxyijvb
eqflvrxtmndwqzgojiaaofmyuxflnersotsnhuqvuwogdvrvngvydzrvxemokuncbjpkdkldqvjyypbkyscxycfpoqqwmpihc
bkepzaqmhznmhoeypjolutjzvhrcofudfhjqwlqigckkhvfbtmaukelapyxhiwkbkumlzucrjeoqkkvqopqezowudgypgvbi
qhpikazvemvatpmslslsbnprkjkwipwmpiklqzcovkltozobdvqwxzsymmielbiyhhpjpzrchjbgwognqbedazuyybhwep
vjjoenlthalhsldzaybdavdusnznutnayqtjgzvclcyxrymrsffnlebcnunjlstqcumeakbxvxpwngepecbwhjdqdgkwmldod
anlfgdhwedgkhydzvzqshszpzhsooeviojtdfsnmupbgjdlkqxxwgoqkowbinfeatjijqkgcvxjthhweqorultxtyzuzszx
todontqmntvbdvzjkegexbbiwlgneuzdgekmesthmtvdrajvudqsyjzldptximdxmkmcaqpcyddtkdszlsyqacnsocvtnrgkh
ljkybohbkwbpxymznkhzsrftzyxnxgzesemcqrxdxhlsbhpxtubylkckctmsfgsapaybeyavvkjkuuokpxwdfmgjdnewg
fcdwrcgeqplisfmwdtuyixncdsmmqzxbqlqoqmwbcqhpompccqskatsvdmfbuffjezejelckdekyovvknruaznmpyovbqgnl
vsacjtffjrgjqhzwahlfoccgzprbiopyrntrczviwrvzuslfajubqifhrxcuafencjvorxpuuyuumibkovukaouqvqecdmix
prtnlfnegsoeqbbancfwaaiuvctnwecrbkllxuicoptxnenycizzhealbociutuwfeiligkpgdxmqesbvsihfggingtwdbstby
ytdrhvvqadpttpbeokrunzxjrpqcfazlxfoplsaycxslupryxfuxtufpuzmxtggplnvvgfvgthsgjitgnishkfhxkrtebawhjk
lzblsucuoakunfbdiajgpusikplloeakhcmzbmvalkengujuibsyvlwbcbmjmusctnhxyrtsgttnnzqvpaeovogkmmwzdszbbvys
jxwxstabbrrdagppsvqqvznadelxryjxvybrhljbsxfkkjiwetqoxgqybzolkyijdhifkhlzynaajcwkyksqagkoaeitmpvk
rpkxfdlhgrujxkpjrbeomfqkzbdcsadapudxwxzrkjjhwonqjzniuwlwehejvociozohciehdfwkwjkwknlvoityrgimkffb
kpcaqoedmpklpaxfkbedgvhbxgzfdjninstjpdasadtynvhxevmvpknjpuycpzyssoehrabbgrapzkslesrvqrfupjxzcgttp
tvybstklfwrncrnoqhrrjkbglselexredciexxwlsopcxxtmtalmnmzybfegllwffumpeoupodnlszwpmllsxqqxxufsnqksnr
jroyjyfaaqkuactaioiprzmzmljlslaxzevakwlqlmcoazmgngkhtomsdzainwawrellfkfzpcjqnqjygsahasbtdmdeg
xytshkcsnvlqityiyitoyosemztcgvaxrubrmunoyilpxfppcjdpqwyngxlqvmzdpwxkjecehrwwolsasftsonlfjuqfwmttoax
qkezhktzjtmjtnaadpeyxuwkwjhrdtdvjuyjzzsmzhnjlakldrnaaktpcgdxqjcbtjyppmhtfoaykzvyjbwwcyrewolpjhnhx
lbekuogzzcftayukifvuglbt

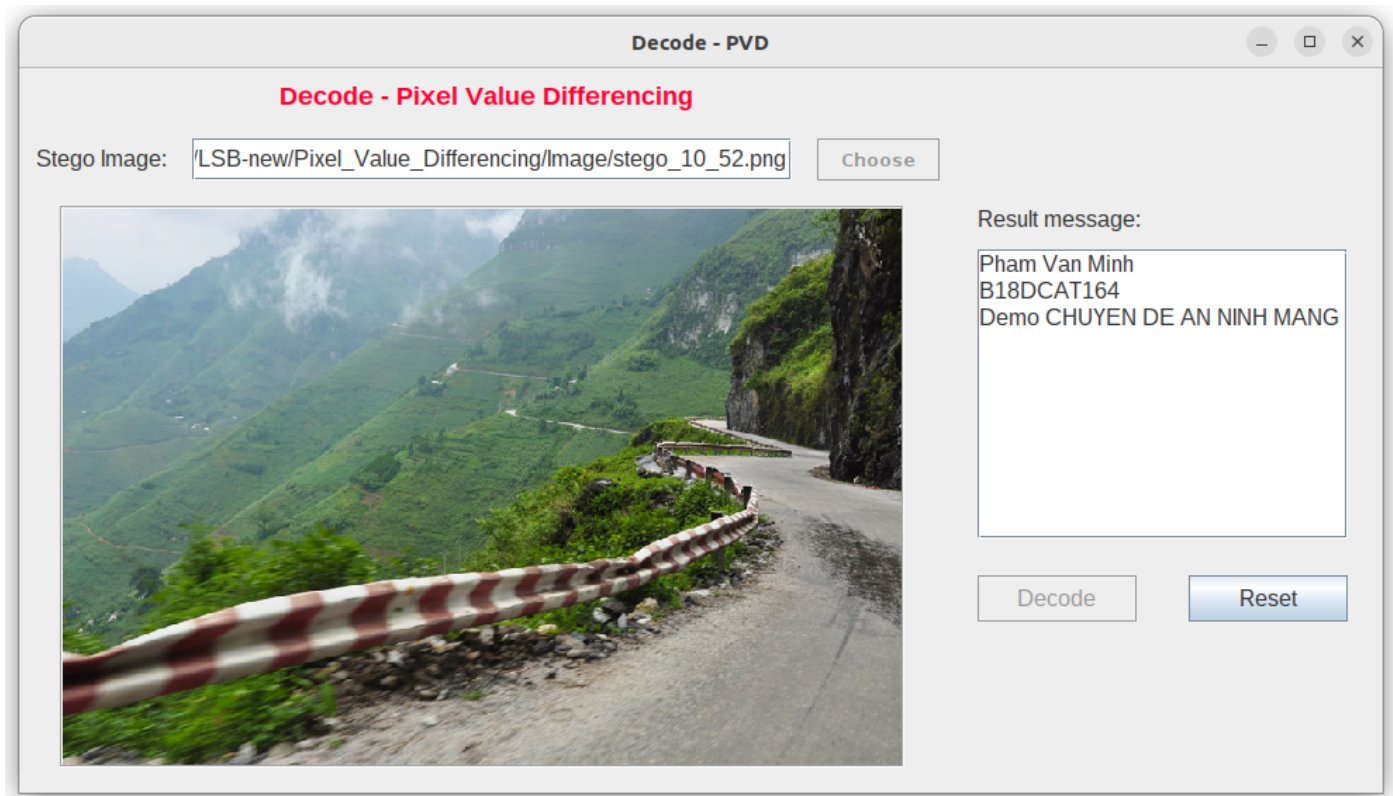
Nhập 1 để tiếp tục (hoặc 0 để thoát): 0
```

4.3 Pixel Value Differencing

- Giấu tin:



- Tách tin:



4.4. Phân tích cặp màu gần nhau - Close Color Pair

- Phát hiện ảnh có giấu tin trên LSB: *python3 ccp.py*

```

minh@minh-HP-340-G2:~/Desktop/LSB-new/LSB_CCP
→ LSB_CCP git:(main) X python3 ccp.py

===== Close Color Pair =====
Nhập ảnh cần kiểm tra: stego_1.png

Số lượng cặp màu gần nhau: 3156
Số lượng cặp màu đặc biệt: 24678
Tỉ lệ màu gần nhau / màu đặc biệt là: 0.12788718696814977

Số lượng cặp màu gần nhau: 1080
Số lượng cặp màu đặc biệt: 4849
Tỉ lệ màu gần nhau / màu đặc biệt là: 0.22272633532687153

Ngưỡng phân biệt ảnh  $m = (R - R') * 100 / R$ : -74.15844433449098
Tỉ lệ  $R/R'$ : 0.574189786674591
==> Stegano Image
→ LSB_CCP git:(main) X

```

- Làm tương tự đối với các ảnh còn lại. Kiểm tra với 50 ảnh stego có tỉ lệ nhúng để tạo ra ảnh C' là 0.05 (5%) và tỉ lệ nhúng để tạo ra ảnh C' là 0.02 (2%).

Demo Close Color Pair

⇒ **Đánh giá:**

Tỉ lệ nhúng để tạo ảnh C'	Ảnh clean	Ảnh stego_10 %	Ảnh stego_20 %	Ảnh stego_30 %	Ảnh stego_50 %
0.05 (5%)	Sai 0/50	Sai 49/50	Sai 0/50	Sai 0/50	Sai 0/50
Stego_10% phát hiện sai 49/50 ⇒ Nếu giảm tỉ lệ nhúng từ 5% xuống còn 2% khi tạo ảnh C' ⇒ Cho kết quả chính xác 100%					

0.02 (2%)	Sai 0/50	Sai 0/50	Sai 1/50	Sai 2/50	Sai 3/50
Phát hiện sai là do ngưỡng m lấy ≤ 1.001 \Rightarrow Nếu đặt ngưỡng lên 1.0012 sẽ ra kết quả chính xác 100%					

\Rightarrow Thuật toán thử nghiệm với 50 ảnh cover (nhúng 2% hay 5% để tạo ra C') cho thấy sự thay đổi giữa giá trị R và R' là rất lớn, cho nên giá trị của ngưỡng m để so sánh được tính cho ảnh cover cũng lớn.

\Rightarrow Đối với 50 ảnh stego (chứa 10%, 20%, 30% và 50% tin và nhúng 2% hay 5% để tạo ra ảnh C') thì sự thay đổi giá trị R và R' là không đáng kể.

Chúng ta thấy thuật toán áp dụng cho ảnh có giấu tin với tỉ lệ càng lớn thì càng dễ phát hiện. Và đối với ảnh có chứa tin nhỏ thì tỉ lệ nhúng thêm khi tạo ảnh C' cũng cần nhỏ để phát hiện chính xác.

\Rightarrow Thử nghiệm lại với tỉ lệ nhúng 0.01 (1%) để tạo ra ảnh C' ra kết quả chính xác với kết luận trên.