BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**


**NGÔ THANH NGUYÊN**

# NÉN FRACTAL CHO BÀI TOÁN ẨN DỮ LIỆU

## LUẬN VĂN THẠC SỸ

**Chuyên ngành:** Công nghệ Thông tin

**Mã ngành:** 60480201

TP. HCM, tháng 03 năm 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**


**NGÔ THANH NGUYÊN**

# NÉN FRACTAL CHO BÀI TOÁN ẨN DỮ LIỆU

## LUẬN VĂN THẠC SỸ

Chuyên ngành: Công nghệ Thông tin

Mã ngành: 60480201

**CÁN BỘ HƯỚNG DẪN KHOA HỌC: PGS.TS LÊ HOÀI BẮC**

TP. HCM, tháng 03 năm 2015

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**

**Cán bộ hướng dẫn khoa học:**

**PGS. TS LÊ HOÀI BẮC**

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM (HUTECH) *ngày 11 tháng 4 năm 2015.*

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

| TT | Họ và Tên | Chức danh Hội đồng |
|---|---|---|
| 1 | **PGS. TSKH Nguyễn Xuân Huy** | **Chủ tịch** |
| 2 | **GS.TSKH Hoàng Văn Kiếm** | **Phản biện 1** |
| 3 | **TS Võ Đình Bảy** | **Phản biện 2** |
| 4 | **PGS. TS Đỗ Phúc** | **Ủy viên** |
| 5 | **TS. Nguyễn Văn Mùi** | **Ủy viên, Thư ký** |

Xác nhận của Chủ tịch Hội đồng đánh giá Luận văn sau khi Luận văn đã sửa chữa (nếu có).

**Chủ tịch Hội đồng đánh giá LV**

**PGS. TSKH Nguyễn Xuân Huy**

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
**PHÒNG QLKH – ĐTSĐH**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập – Tự do – Hạnh phúc**

*TP. HCM, ngày..... tháng….. năm 2015*

# NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên : Ngô Thanh Nguyên — Giới tính: Nam.
Ngày, tháng, năm sinh : 09-09-1988 — Nơi sinh: TP.Pleiku .
Chuyên ngành : Công Nghệ Thông Tin — MSHV : 1341860014.

**I- Tên đề tài:** NÉN FRACTAL CHO BÀI TOÁN ẨN DỮ LIỆU

**II- Nhiệm vụ và nội dung:**

- Nghiên cứu các phương pháp ẩn dữ liệu.
- Nghiên cứu lĩnh vực nén Fractal.
- Áp dụng nén Fractal cho bài toán ẩn dữ liệu.

**III- Ngày giao nhiệm vụ:**18– 08 – 2014

**IV- Ngày hoàn thành nhiệm vụ:** 14 – 03 – 2015

**V- Cán bộ hướng dẫn:** Phó Giáo Sư .Tiến Sĩ. Lê Hoài Bắc

**CÁN BỘ HƯỚNG DẪN**
(Họ tên và chữ ký)

**KHOA QUẢN LÝ CHUYÊN NGÀNH**
(Họ tên và chữ ký)

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

**Học viên thực hiện Luận văn**

**Ngô Thanh Nguyên**

# LỜI CẢM ƠN !

Em xin bày tỏ lòng biết ơn sâu sắc nhất tới **PGS. TS.Lê Hoài Bắc**, thầy đã tận tình hướng dẫn và giúp đỡ em rất nhiều trong quá trình tìm hiểu nghiên cứu đề tài được giao để em có thể hoàn thành tốt luận văn của mình.

Em xin chân thành cảm ơn đến quý thầy cô trong khoa công nghệ thông tin đã tận tình dạy bảo, truyền đạt cho em nhiều kiến thức quý báu trong suốt thời gian học tập tại trường.

Trong quá trình nghiên cứu mặc dù đã được các thầy cô giáo hướng dẫn tận tình nhưng do nhiều nguyên nhân chủ quan và khách quan nên đề tài không tránh khỏi sai sót. Em rất mong nhận được những đóng góp ý kiến quý báu của quý thầy cô để em có thể phát triển và mở rộng đề tài nghiên cứu của mình.

Em xin chân thành cám ơn !

*TP. Hồ Chí Minh, tháng 03 năm 2015*

**Người thực hiện**

**Ngô Thanh Nguyên.**

# TÓM TẮT

Đối với các bài toán Ẩn dữ liệu (Data hiding-DH) [2] thì chúng ta quan tâm tới đó chính là khả năng nhúng, tính vô hình cũng như tính mạnh mẽ chống tấn công. Ở luận văn này tôi đề xuất phương pháp sử dụng thuật toán nén Fractal[8] trong các bài toán DH. Với thuật toán nén Fractal sẽ giải quyết cho các bài toán DH về khả năng nhúng và tính vô hình bởi khi thông tin mật được nén với tỉ lệ nén cao như Fractal thì đồng thời khả năng nhúng của các bài toán DH sẽ tăng cao. Để dễ dàng trong việc đánh giá phương pháp này ta sử dụng thuật toán ẩn dữ liệu là LSB matching revisited.

## ABSTRACT

In Data hiding [2] problem, What we care about is not only ability hidding, ability invisible,and ability against attack. In this dissertation, I suggest using Fractal compression[8] method to solve it. Fractal method will solve ability hidding and ability invisible in data hiding problem because information confidential compressed With Fractal is hight compression ratio so ability hidding of Data hiding problem increase. To estimate this method easier, the data hiding we should use is LSB.

# Mục lục

# Danh mục các từ viết tắt

| Từ viết tắt | Từ đầy đủ |
|---|---|
| DH | Data Hiding |
| LSB | Least Significant Bit |
| IFS | Iterated Function Systems |
| LSB- MR | Least Significant Bit matching revisited |
| PSNR | Peak Signal to Noise Ratio |
| MSE | Mean squared error |
| HCF COM | Histogram characteristic function center of mass |
| BER | Bit error rate |
| C | Cover |
| M | Message |
| S | Stego |
| POV | Pair of values |
| RS | Regular Singular |

# Danh mục các bảng

# Danh mục các biểu đồ, đồ thị, sơ đồ, hình ảnh

# A. MỞ ĐẦU

Ẩn dữ liệu hay giấu tin ( Data hiding- DH) là một kĩ thuật đã không còn xa lạ với chúng ta, nó đã ra đời từ rất lâu trên thế giới. Việc ẩn dữ liệu nhằm mục đích che giấu thông tin bên trong một số tài liệu như văn bản, hình ảnh, âm thanh và phim v.v… DH khác với mã hóa ở một điểm là mã hóa tập trung vào việc giữ bí mật nội dung thong điệp, còn DH thì tập trung vào việc giữ bí mật sự tồn tại của thông điệp[1]. DH có thế mạnh hơn mã hóa khác chính là DH sẽ không thu hút sự chú ý của người xung quanh. Một thông điệp khi được mã hóa tinh vi đến mức nào cũng sẽ rất kích thích sự tò mò của mọi người và như thế người ta sẽ tìm cách giãi mã hoặc phá hủy nó.

Nhiều phương pháp DH đang được nghiên cứu, mỗi phương pháp có những ưu điểm và nhược điểm khác nhau, trong đó có thể kể tên một số phương pháp như sử dụng các bit trọng số nhỏ, phương pháp sử dụng các hệ số biến đổi… Nhưng các bài toán được đặt ra hiện nay của các bài toán DH đó chính là làm sao nâng cao được tính bền vững, tính trong suốt, và khả năng lưu trữ. Phương pháp hiệu quả và đơn giản nhất để tăng dung lượng nhúng là sử dụng phương pháp nén thông tin. Nén thông tin sẽ làm giảm các dữ liệu dư thừa và góp phần làm tăng hiệu quả cho việc mã hóa và giấu tin mật. Do vậy việc kết hợp các phương pháp nén thông tin và giấu tin mật không những làm tăng dung lượng nhúng mà còn làm thông tin mật được bảo vệ thêm một lớp mới giúp tăng mức độ an toàn và tăng được tính vô hình của thông tin mật trong sản phẩm DH (ở luận văn này tôi chọn ẩn dữ liệu trong hình ảnh). Vì vậy trong luận văn của mình, tôi trình bày về vấn đề nén thông tin mật Fractal trong các bài toán DH.

# B. NỘI DUNG

## CHƯƠNG I. TỔNG QUAN

### 1.1 Giấu tin

Giấu tin [2] là giấu (hoặc nhúng) một lượng thông tin số vào trong đối tượng dữ liệu số khác. "Giấu tin" nhiều khi không phải chỉ hành động giấu theo nghĩa thông thường, mà chỉ mang ý nghĩa quy ước.

giấu tin có lịch sử hình thành và phát triển từ rất lâu đời, nó bắt nguồn từ Hi Lạp và được sử dụng cho tới ngày nay, chủ yếu phục vụ cho mục đích liên lạc bí mật. Theo các tài liệu nghiên cứu ghi lại, kỹ thuật giấu tin cổ xưa nhất và cũng là đơn giản nhất được nhắc tới trong các tài liệu là khi vua Histiaeus (khoảng năm 440 TCN) cạo sạch tóc xăm thông điệp lên da đầu và chờ khi tóc mọc lại, người nô lệ đó chuyển thông tin tới người nhận. Sau đó, người ta sử dụng các vật liệu tự nhiên như bảng gỗ, sáp ong, hổ phách cho việc giấu thông tin.

Khi kỹ thuật phát triển hơn, con người sử dụng chữ viết với cỡ chữ nhỏ giấu trong các vật dụng hàng ngày (như các hộp, vali có hai đáy) để chuyển đi, hoặc dùng bồ câu để chuyển thông tin để che mắt các nhân viên an ninh, hải quan. Sang thế kỷ 17, người ta dùng cách đánh dấu vào các kí tự cần thiết trên một văn bản, một bài báo công khai nào đó rồi truyền tới tay người nhận. Sau đó là thời kì phát triển rực rỡ của công nghệ hoá học với sản phẩm là mực không màu - là các chất lỏng sản phẩm hữu cơ không màu và hiển thị màu khi gặp điều kiện hoá - lý thích hợp. Tới ngày nay với phương pháp kiểm tra độ ẩm bề mặt, mực không màu không còn tác dụng bảo mật nữa, nhưng nó vẫn còn được dùng như một dạng thuỷ vân để in các block nhỏ hay các chi tiết phát quang khi bị chiếu tia cực tím. Trong nửa cuối thế kỉ 19, các vi phim là bước phát triển kế tiếp, với sản phẩm hoàn hảo của các thợ ảnh chuyên nghiệp thì kích thước của mỗi thông điệp "chỉ nhỏ như một dấu chấm".

***Mục đích của giấu tin:***

Giấu tin phục vụ cho hai mục đích trái ngược nhau:

- Bảo mật cho những dữ liệu được giấu trong đối tượng chứa.

- Bảo đảm an toàn (bảo vệ bản quyền) cho chính đối tượng chứa dữ liệu giấu trong đó.

Hai mục đích giấu tin phát triển thành hai lĩnh vực với yêu cầu và tính chất khác nhau :

- 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 kỹ trong một đối tượng khác sao cho người khác không phát hiện được.

-Kỹ thuật giấu thông tin theo kiểu đánh giấu (watermarking): để bảo vệ bản quyền của đối tượng chứa thông tin 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 thuỷ vân số.

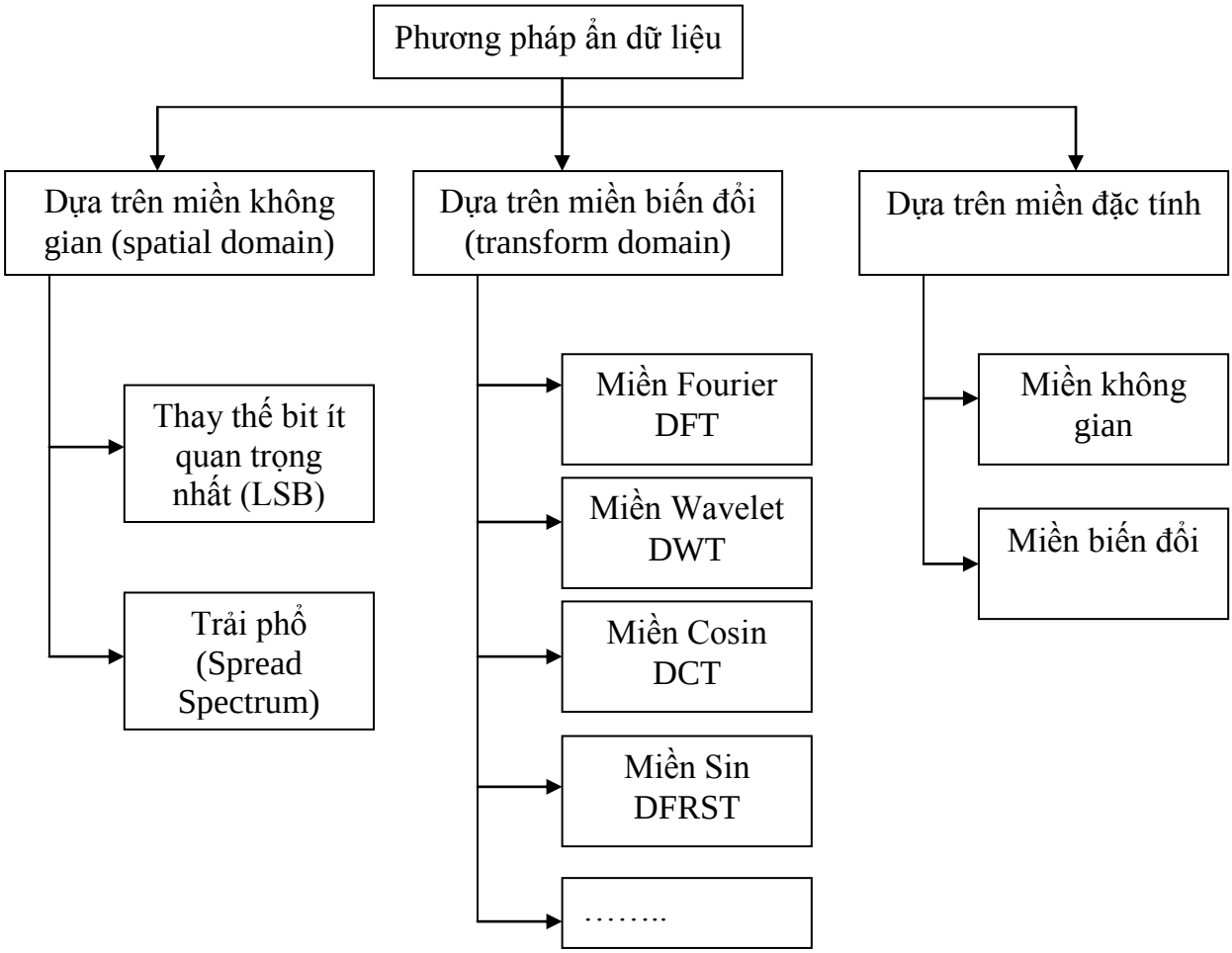
Hai kỹ thuật này đều giống nhau ở một điểm là có ba đối trượng tham gia chính đó là đối tượng chứa hay còn gọi là đối tượng trước khi nhúng (C), đối tượng cần được nhúng (M), và đối tượng cuối là đối tượng sau khi nhúng (S). Sự khác biệt của hai kỹ thuật này cũng dựa vào ba đối tượng này chính là

Bảng 1. 1*khác biệt cơ bản giữa Steganography và Watermarking*

| | *Steganography* | *Watermarking* |
|---|---|---|
| *Đối tượng (C)* | Là trung chuyển hay vật che chắn | Là vật chủ hay đối tượng chưa đánh dấu cần được bảo vệ |
| *Đối tượng (M)* | Là đối tượng thông tin mật cần được bảo vệ | Là tín hiệu vân hay thông tin bản quyền |
| *Đối tượng (S)* | Là vật chứa thông tin mật | Là tín hiệu đã thủy vân hay đối tượng đã mang thông tin bản quyền. |

***Phân loại kỹ thuật giấu tin***

Có nhiều cách để tiến hành phân loại các phương pháp giấu thông tin theo các tiêu chí khác nhau*[2]* như theo các phương tiện chứa tin, các phương pháp tác động lên phương tiện chứa tin, hay phân loại theo các ứng dụng cụ thể v.v…




Hình 1. 1 Mô hình phân loại theo miền nhúng

*a. Phân loại theo miền nhúng*

Với nhóm phương pháp làm việc trên miền không gian, các thao tác được thực hiện trực tiếp trên các pixel hay samples. Khi đó thông tin cần nhúng *M* sẽ đưa vào đối tượng chứa *C* bằng cách thay đổi trực tiếp các giá trị của các pixel hay samples. Với phương pháp này dễ thực hiện, cho dung lượng cao nhưng không bền vững với nhiều thao tác tấn công. Vơi nhóm các phương pháp tiến hành trong miền biế đổi cho khả năng chống tấn công tốt cũng như đảm bảo tính vô hình, tuy nhiên độ phứ tạp của thuật toán nhúng và trích cao.

*b.Phân loại theo phương tiện chứa tin*

- Giấu thông tin trong ảnh:

Hiện nay giấu thông tin trong ảnh là một bộ phận chiếm tỷ lệ lớn nhất trong các chương trình ứng dụng, các phần mềm, hệ thống giấu tin trong đa phương tiện bởi lượng thông tin được trao đổi bằng ảnh là rất lớn và hơn nữa giấu thông tin trong ảnh cũng 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…Thông tin sẽ được giấu cùng với dữ liệu ảnh nhưng chất lượng ảnh ít thay đổi và chẳng ai biết được đằng sau ảnh đó mang những thông tin có ý nghĩa. Ngày nay khi ảnh số đã được sử dụng rất phổ biến thì giấu thông tin trong ảnh đã đem lại nhiều những ứng dụng quan trọng trên các lĩnh vực trong đời sống xã hội. Ví dụ như ở các nước phát triển chữ ký tay đã được số hoá và lưu trữ sử dụng như là hồ sơ cá nhân của các dịch vụ ngân hàng tài chính.

Phần mềm WinWord của Microsoft cũng cho phép người dùng lưu trữ chữ ký trong ảnh nhị phân rồi gắn vào vị trí nào đó trong file văn bản để đảm bảo tính an toàn của thông tin.

- Giấu thông tin trong các file âm thanh:

Giấu thông tin trong audio mang những đặc điểm riêng khác với giấu thông tin trong các đối tượng đa phương tiện khác. Một trong những yêu cầu cơ bản của giấu tin

là đảm bảo tính chất ẩn của thông tin được giấu đồng thời không làm ảnh hưởng đến chất lượng của dữ liệu. Để đảm bảo yêu cầu này ta lưu ý rằng kỹ thuật giấu thông tin trong ảnh phụ thuộc vào hệ thống thị giác của con người – HSV (Human Vision System) còn kỹ thuật giấu thông tin trong audio lại phụ thuộc vào hệ thống thính giác HAS (Human Auditory System).

Một vấn đề khó khăn ở đây là hệ thống thính giác của con người nghe được các tín hiệu ở các giải tần rộng và công suất lớn nên đã gây khó dễ đối với các phương pháp giấu tin trong audio. Nhưng tai con người lại kém trong việc phát hiện sự khác biệt của các giải tần và công suất, có nghĩa là các âm thanh to, cao tần có thể che giấu được các âm thanh nhỏ thấp một cách dễ dàng.

Vấn đề khó khăn thứ hai đối với giấu tin trong audio là kênh truyền tin, kênh truyền hay băng thông chậm sẽ ảnh hưởng đến chất lượng thông tin sau khi giấu. Giấu thông tin trong audio đòi hỏi yêu cầu rất cao về tính đồng bộ và tính an toàn của thông tin. Các phương pháp giấu thông tin trong audio đều lợi dụng điểm yếu trong hệ thống thính giác của con người.

- Giấu thông 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, nhận thức thông tin, bản quyền tác giả…

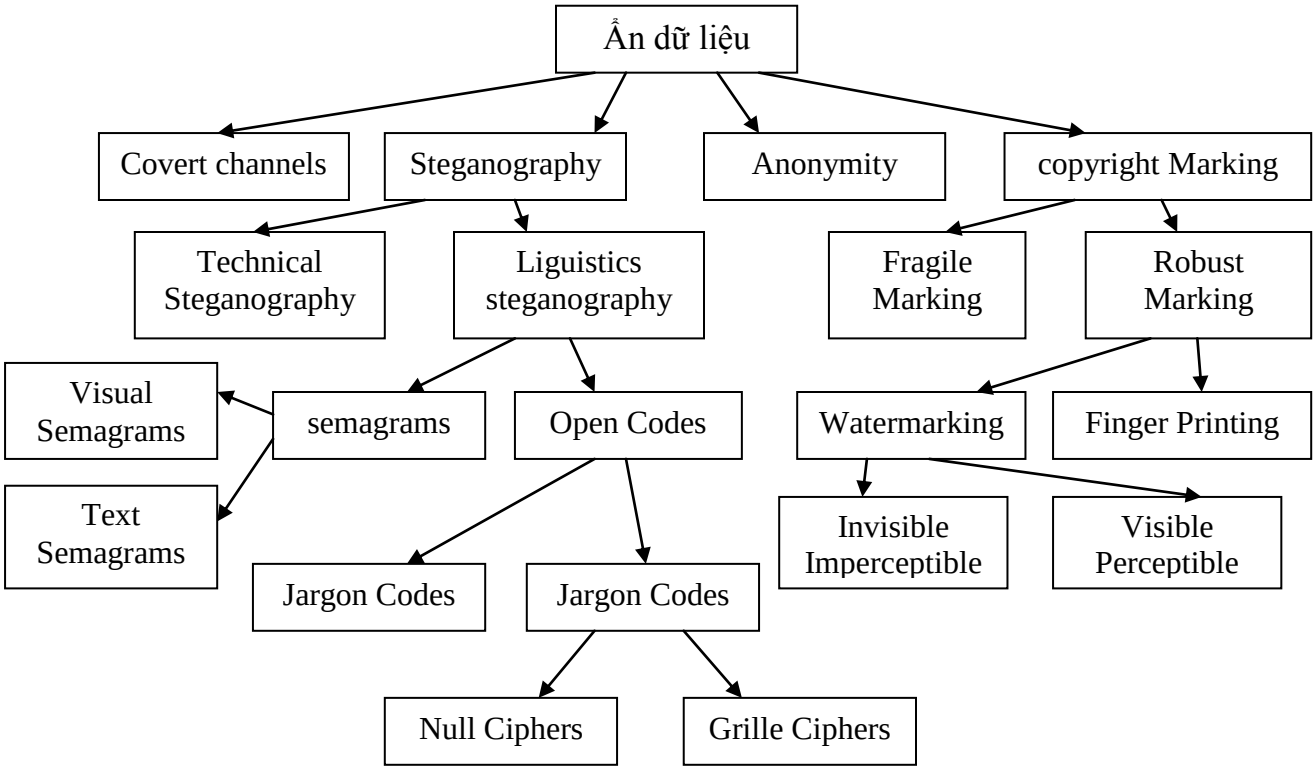
Một phương pháp giấu tin trong video được đưa ra bởi Cox là phương pháp phân bố đều. Ý tưởng cơ bản của phương pháp là phân phối thông tin giấu dàn trải theo tần số của dữ liệu gốc. Nhiều nhà nghiên cứu đã dùng những hàm cosin riêng và các hệ số truyền sóng riêng để giấu tin. Trong các thuật toán khởi nguồn thì thường các kỹ thuật cho phép giấu các ảnh vào trong video nhưng thời gian gần đây các kỹ thuật cho phép giấu cả âm thanh và hình ảnh vào video.

- Giấu thông 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 gì dữ liệu đa phương tiện như ảnh, audio, video. Gần đây đã có một số nghiên cứu giấu tin trong cơ sở dữ liệu quan hệ, các gói IP truyền trên mạng chắc chắn sau này còn tiếp tục phát triển tiếp cho các môi trường dữ liệu số khác.

*b. Phân loại theo kỹ thuật:*




Hình 1. 2 Mô hình phân loại theo kỹ thuật

- Che dấu kênh truyền (Covert channels): những kênh truyền thông được ẩn bên trong những kênh truyền thông trung gian hợp lệ khác. Nó sẽ lấy băng thông của các kênh truyền thông trung gian này để thực hiện việc truyền tải thông tin một cách bí mật, mà không cần sự cho phép của kênh trung gian.

- Che dấu định danh: là kỹ thuật dùng để che giấu nội dụng meta của thông điệp như thông tin về người gửi và người nhận thông điệp. Kỹ thuật này thường được dùng rộng rãi trên Internet nhằm bảo vệ quyền người dùng.

- Giấu thông tin bí mật (Stegnography): là kỹ thuật dùng để che dấu sự tồn tại của những thông tin bí mật trong quá trình truyền thông giữa người gửi và người nhận, sao cho tác nhân thứ ba không cảm nhận được sự tồn tại của thông tin mật.

- Đánh dấu bản quyền (Copyright marking): là kỹ thuật nhúng một dấu hiệu bản quyền vào trong các đối tượng chứa, nhằm chống lại các sự xâm phạm trên đối tượng chứa cũng như xác nhận quyền sở hữu hợp pháp của tác giả.

*c. Phân loại theo Ứng dụng:*

Mô hình phân loại này được thực hiện dựa trên các ứng dụng thực tế của hệ thống. – Bảo vệ bản quyền:

tín hiệu *C* sẽ được bảo vệ quyền tác giả, tác phẩm trong quá trình trao đổi như mua bán chuyển tải, phân phối. Ngoài ra, một số phương pháp cho phép lần vết và kiểm soát đối tượng trong quá trình nhân bản tác phẩm.

- Chứng thực thông tin ảnh:

Rất phổ biến và hữu dụng khi có tranh chấp về quyền sở hữu hợp pháp xảy ra. Một đăc trưng nữa của nó chính là hầu hết đều là hệ thống không mờ. Có nghĩa là, trong quá trình chứng thực, thông tin của người sở hữu hợp pháp sẽ được đem ra để so sánh với thông tin đã được nhúng trước đó (*M*)

- Giấu tin:

Trong thực tế nó chính là việc đưa thông tin cá nhân của bệnh nhân vào các ảnh X-Quang hay film hay bệnh án của chính người bệnh đó. Vì ảnh y khoa có những đặc trưng riêng nên các phương pháp thuộc nhóm này thường không nhiều và khá chuyên biệt. Ngoài ra, điều kiện broadcast cũng là ứng dụng khá đặc biệt của lĩnh vực giấu tin do các đặc điểm về tín hiệu truyền cũng như thông tin nhúng vào.

- Truyền thông mật:

Là lĩnh vực lớn trong ngành an ninh quốc gia. Thông tin *M* được giấu trong tín hiệu chứa *C* thường là các thông tin tuyệt mật, nên nhóm các phương pháp thuộc ứng dụng này yêu cầu khả năng vô hình và dung lượng cao. Ngoài ra các hệ thống này rất phức tạp và thao tác ẩn dữ liệu chỉ là một giai đoạn nhỏ trong nguyên quá trình truyền thông mật.

### 1.1.1 Mô hình kỹ thuật giấu thông tin cơ bản

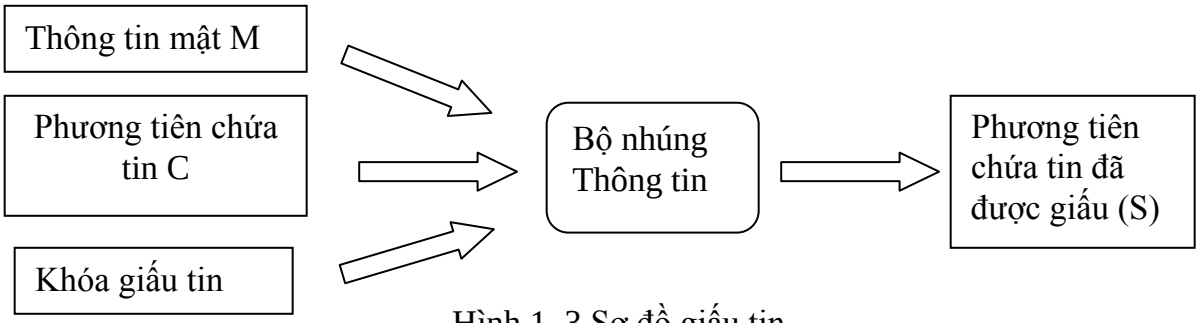
Giấu tin và tách thông tin là hai quá trình cơ bản của các bài toán ẩn dữ liệu.

*a) Mô hình giấu tin vào phương tiện chứa:*

**Đầu vào:**

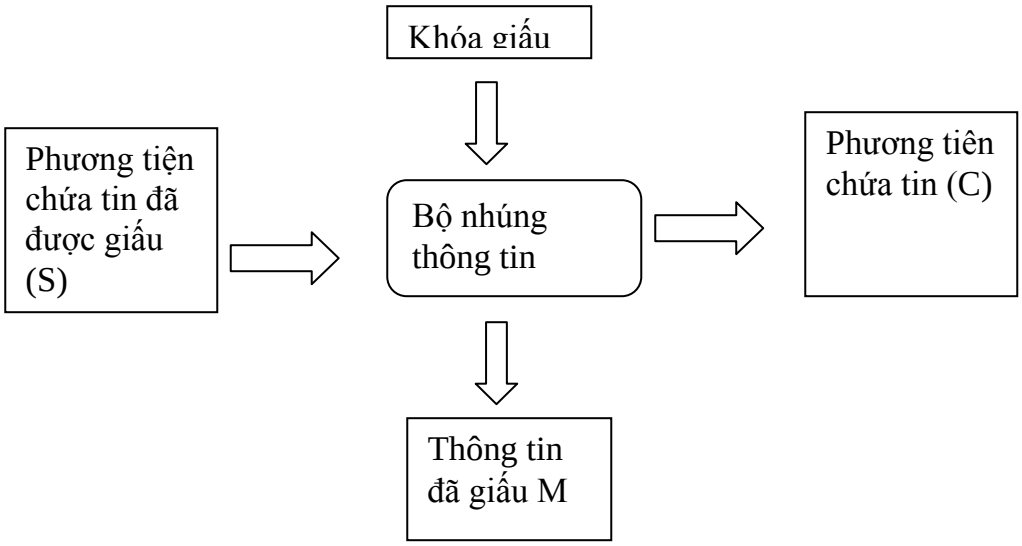
- Thông tin cần giấu: Tuỳ theo mục đích của người dùng, nó có thể là thông điệp (với giấu tin bí mật) hay là các logo, hình ảnh bản quyền.
- Phương tiện chứa: các file ảnh, text, audio…là môi trường để giấu tin.
- Khoá: thành phần để góp phần làm tăng độ bảo mật.

Bộ nhúng thông tin: là chương trình thực hiện việc giấu tin.




Hình 1. 3 Sơ đồ giấu tin

**Đầu ra:** là phương tiện chứa, đã có tin giấu trong đó.




Hình 1. 4 Sơ đồ tách tin

*a) Mô hình tách tin từ phương tiện chứa:*

Diễn ra theo quy trình ngược lại với giấu tin: đầu ra là các thông tin được giấu và phương tiện chứa

**1.1.2 Các giao thức giấu tin**

Khi một thuật toán giấu tin được sử dụng, thuật toán đó sẽ nằm trong khuôn khổ một giao thức xác định, thích hợp để xử lý dữ liệu.

Theo lý thuyết, có ba kiểu giao thức cơ bản: giấu tin thuần tuý, giấu tin với khoá bí mật, giấu tin với khoá công khai. Trong đó kiểu giấu tin sau cùng được xây dựng trên nguyên tắc mật mã khoá công khai.

**1.1.2.1 Giấu tin thuần túy**

Giấu tin thuần tuý là hệ thống giấu tin, không yêu cầu phải trao đổi trước một số thông tin bí mật. Trong hệ thống giấu tin thuần tuý, người giấu tin và người tách tin phải thực hiện cùng một thuật toán nhúng và tách thông tin, thuật toán này phải được giữ bí mật.

*Định nghĩa 1*: Giấu tin thuần tuý

Bộ bốn giá trị $\delta = (C, M, D, E)$ được gọi là Hệ giấu tin thuần tuý trong đó:

C là tập các phương tiện chứa thông tin có thể, M là tập các thông điệp cần giấu $|C| \geq |M|$.

$E: C \times M \rightarrow C$ là hàm nhúng và $D: C \rightarrow M$ là hàm tách, với tính chất $D(E(c, m)) = m$ với $m \in M$ và $c \in C$.

Trong giấu tin thuần tuý, độ bảo mật thông tin dựa trên chính thuật toán, phương tiện chứa trước và sau khi nhận tin giấu cũng phải được bảo vệ cẩn thận. Nếu đối phương tấn công vào nơi cất giữ phương tiện chứa, việc giấu thông tin sẽ không hiệu quả, khi đó đối phương không những phát hiện được việc liên lạc bí mật, mà còn lấy được cả thông tin giấu trong đó.

Phương pháp giấu tin thuần tuý phải được kết hợp với việc mã hoá thông tin. Trước tiên việc mã hoá sẽ làm tăng độ bảo mật của thông điệp, sau đó nhúng bản mã

vào trong phương tiện chứa. Cách này sẽ làm tăng độ bảo mật và vẫn đảm bảo tính vô hình của kênh liên lạc, nó thực sự khó khăn cho việc phát hiện hay tấn công các thông điệp.

**1.1.2.2 Giấu tin sử dụng khoá bí mật**

Đối với hệ thống giấu thông tin thuần tuý, độ an toàn phụ thuộc hoàn toàn vào độ bí mật của thuật toán giấu và tách thông tin.

Để cho hệ thống an toàn hơn, người ta thực hiện trao đổi một số thông tin bí mật giữa hai đối tác. Trong hệ thống giấu tin với khoá bí mật, người gửi chọn phương tiện chứa thông tin, sử dụng khoá bí mật k, tiến hành nhúng thông điệp vào phương tiện chứa tin đó. Giấu tin với khoá bí mật vẫn phải đảm bảo phương tiện chứa (trước và sau khi giấu tin) phải giống nhau về cảm nhận, để tránh kẻ giám sát phát hiện được phiên liên lạc. Đây là một tiêu chuẩn khi chọn khoá.

*Định nghĩa 2*: Giấu tin sử dụng khoá bí mật

Bộ năm giá trị $\delta = (C, M, K, Dk, Ek)$ được gọi là hệ giấu tin sử dụng khoá bí mật, trong đó:

C là tập các phương tiện chứa có thể, M là tập các thông điệp cần giấu với $|C| \geq |M|$, K là tập các khoá bí mật.

Ek: $C \times M \times K \rightarrow C$ và Dk: $C \times K \rightarrow M$ với điều kiện $Dk(Ek(c, m, k), k) = m$ với mọi $m \in M$, $c \in C$ và $k \in K$.

Giao thức truyền thông tin bằng giấu tin sử dụng khoá bí mật, yêu cầu các bên tham gia phải trao đổi khoá trước.

Có thể dùng một số đặc tính của chính phương tiện chứa làm khoá, hàm băm tính toán các giá trị này để làm khoá. Người nhận cũng tính hàm băm trên chính các giá trị này, để lấy khoá giải mã tách thông tin.

Với cách này, không phải trao đổi khoá trên kênh an toàn, nhưng vì hàm băm không phải là bí mật, nên việc liên lạc bí mật sẽ không đảm bảo.

Có thể chọn các thành phần quan trọng trong phương tiện chứa để làm khoá, các thành phần đó nếu bị thay đổi sẽ ảnh hưởng nghiêm trọng tới phương tiện chứa, và có thể nhận ra được.

#### 1.1.2.3 Giấu tin với khoá công khai

Hệ thống giấu tin với khoá công khai cũng yêu cầu có hai khoá: khóa bí mật và khóa công khai. Khóa công khai được lưu trong Cơ sở dữ liệu khoá công khai, giống như mật mã với khoá công khai, và được dùng trong quá trình nhúng thông tin. Khoá bí mật chỉ người nhận mới biết và được dùng trong quá trình tách lấy thông tin, tái tạo lại thông điệp ban đầu.

Cách dễ nhất để xây dựng hệ thống giấu tin với khoá công khai là sử dụng hệ mật mã với khoá công khai. Giả sử hai đối tác đã trao đổi khoá công khai của thuật toán mã hoá công khai.

Nguyên lý của giấu tin với khoá công khai là dùng hàm giải mã D để giải mã trên mọi phương tiện chứa thông tin C, mà không cần quan tâm việc nó chứa hay không chứa thông điệp bí mật (D là hàm trên tập C). Trong trường hợp phương tiện chứa không có thông tin thu được khi giải mã, ta chỉ thu được các phần tử ngẫu nhiên m, ta gọi là các phần tử "ngẫu nhiên tự nhiên " của phương tiện chứa.

Trong giao thức giấu tin với khoá công khai, khi cố gắng để tách tin, kẻ tấn công chỉ có thể nhận được các thông tin "ngẫu nhiên", vì không có khoá giải mã tương ứng.

### 1.1.3 Các yêu cầu của bài toán Ẩn dữ liệu

*a. tính bền vững:* thể hiện ở khả năng ít thay đổi trước các tấn công bên ngoài như:

+ Đối vói tín hiệu âm thanh: thay đổi tính chất ( tần số lấy mẫu, số bít lấy mẫu, thây đổi độ lớn biên độ….), thay đổi định dạng (WAV –MP3, MP3-MIDI,…)

+ Đối với tín hiệu ảnh: các phép biến đổi affine (dịch, quay, tỉ lệ,...), thay đổi chất lượng ảnh (thay đổi hệ màu, độ sáng,…), chuyển đổi định dạng dữ liệu ( JPG-BMP,GIF –PCX,…)

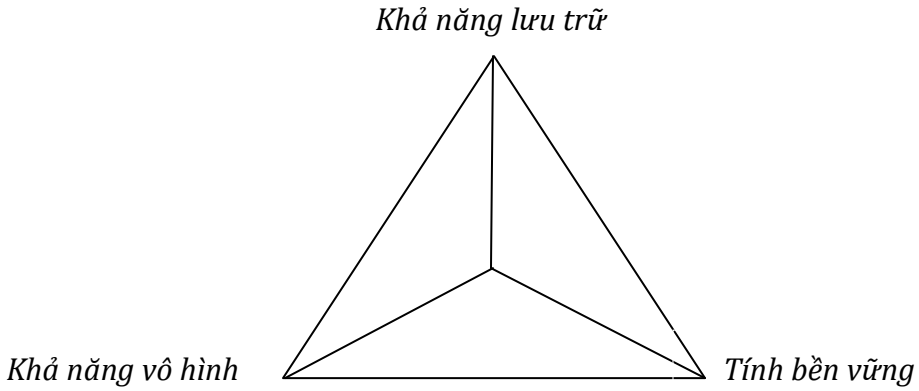
Hiện nay chưa có phương pháp nào có thể đảm bảo được tính chất này một cách tuyệt đối. Mỗi phương  pháp chỉ có thể bền vững với một vài các phép biến đổi nhất định.

*b. tính trong suốt (vô hình):* tính chất này thể hiện khả năng ngụy trang, che giấu thông điệp M trong phương tiện chứa C. Có nghĩa là đối tượng sau khi nhúng S trông "rất giống" đối tượng C ở cả 2 khía cạnh cảm nhận bằng mắt và qua phép đo. Tính vô hình càng cao thì khả năng bị phát hiện càng thấp. Có nghĩa là việc xác định một đối tượng có chứa thông tin mật hay không càng khó. Sự khác biệt giữa 2 đối tượng chứa thông tin mật và đối tượng không chứa thông tin mật được đánh giá qua độ đo PSNR.

*c. khả năng lưu trữ:* Thể hiện ở dung lượng thông tin nhúng tối đa có thể được đưa vào trong phương tiện chứa. Tính chất này thường không là vấn đề quan trọng trong các bài toán giấu tin bên trong các bài toán thủy vân, nhưng nó là tiêu chí rất quan trọng trong các bài toán giấu tin bên cạnh tính vô hình. Khả năng lưu trữ được tính bằng số bit có thể nhúng trên một pixel của đối tượng ảnh C hoặc trên một sample của đối tương âm thanh C.

trong thực tế, khi quyết định chọn phương pháp nhúng nào, ta thường lấy 3 tiêu chí trên làm cơ sở. Tùy thuộc vào từng ứng dụng mà người ta sẽ ưu tiên cho tiêu chuẩn nào hơn. Mối quan hệ giữa 3 tiêu chí này có thể biểu diễn thông qua mộ hình

Rõ ràng 3 tiêu chí này tỉ lệ nghịch nhau. Một cách đơn giản là tại một thời điểm, tổng của 3 tiêu chí này không đổi. Có nghĩa là nếu một hệ thống ẩn dữ liệu đáp ứng tuyệt đối 3 tiêu chí này cùng lúc. Ngoài 3 yêu cầu quan trọng này, một bài toán ẩn dữ liệu cần phải xem xét them 3 yếu tố sau: tính bảo mật, độ phức tạp trong quá trình nhúng, độ phức tạp trong quá trình trích.




Hình 1. 5 Mỗi tương quan giữa ba tiêu chí.

### 1.1.4 Ứng dụng của ẩn dữ liệu

*Theo dõi phát sóng :*

Dùng phương pháp thủy vân đặt một vân duy nhất cho mỗi đoạn phim hoặc đoạn âm thanh trước khi phát sóng. Những trạm theo dõi tự động có thể nhận được chương trình phát và tìm những vân này, xác định xem mỗi đoạn phát xuất hiện ở đâu và vào lúc nào.

*Xác định chủ sở hữu:*

Đưa một vân số vào các tác phẩm này dưới dạng vô hình. Thông tin vân chính là thông tin của chủ sở hưu.

*Chứng thực nội dụng:*

Chứng thực là bài toán lớn trong lĩnh vực mã hóa (Cryptography). Nhiều nhà khoa học đã bàn luận về một ứng dụng của mã hóa trong việc tạo nên mộc chiếc "máy ảnh đáng tin cậy" bằng cách tính toán một chữ ký mật mã tương ứng với một bức ảnh. Chỉ cần một bít của một pixel trên bức ảnh bị thay đổi, nó sẽ không còn khớp với chữ ký mật nữa.

Một phương pháp khác được đề ra là nhúng chữ ký đó trực tiếp vào ảnh bằng cách sử dụng phương pháp thủy vân. Nhờ vậy, chữ ký luôn đi kèm với ảnh. Tất nhiên chữ ký được xem là tín hiệu cần nhúng M và được đặt ở đâu, như thế nào vẫn đang là bài toán mở. Lưu ý rằng bài toán trong trường hợp này là làm thế nào để hệ thống

nhúng vân đạt được yêu cầu của một mô hình thủy vân dễ vỡ tốt nhất. Hơn thế nữa giải pháp này xũng mở ra khả năng mới giúp nhận biết những phép tấn công nào đã được thực hiện trên tín hiệu.

*Kiểm soát sao chép*

Mặc dù có rất nhiều phương pháp được đề xuất cho bài toán theo dõi, định danh và chứng minh quyền sở hữu hợp pháp nhưng thật khó tránh được tình trạng sao chép bất hợp pháp. Do đó, các hệ thống kiểm soát sao chép được thiết kế như một công cụ phục vụ cho mục đích ngăn chặn và điều tra. Hơn thế nữa, việc sử dụng những thiết bị thu có thể ngăn cấm việc thu lại một tín hiệu nếu nó dò thấy một Watermark xác định việc thu này là bất hợp pháp. Tất nhiên để hệ thống như vậy hoạt động thì tất cả các máy thu được sản xuất phải kèm theo 1 bộ dò tìm Watermark trong mạch điện. Những hệ thống như vậy hiện đang được phát triển cho đầu máy DVD và cho những ngành phân phối âm nhạc.

*Truyền thông ngầm*

Truyền thông ngầm là một trong những ứng dụng sớm nhất của công nghệ giấu tin, hay chính xác hơn là của ẩn dữ liệu. Các hệ thống thuộc ứng dụng này cho phép truyền đi những thông điệp bí mật. ứng dụng này được mô tả thông qua bài toán kinh điển " bài toán người tù". Có 2 phạm nhân ở 2 phòng khác nhau đang cố truyền tin cho nhau. Vấn đề của họ là không thể truyền thông tin trực tiếp mà phải thông qua người cai tù. Người cai tù sẵn sang truyền giúp họ những thông điệp không có hại gì, nhưng sẽ trừng phạt nặng nếu anh ta tìm ra các thông tin đáng ngờ chứa các nội dung không được phép ở đây là giấu thông điệp về kế hoạch vượt ngục vào trong thông điệp tưởng chừng vô hại.

*Ứng dựng ở Việt Nam:*

Ở Việt Nam đã có một số cơ quan, đơn vị tập trung nghiên cứu về vấn đề này như:

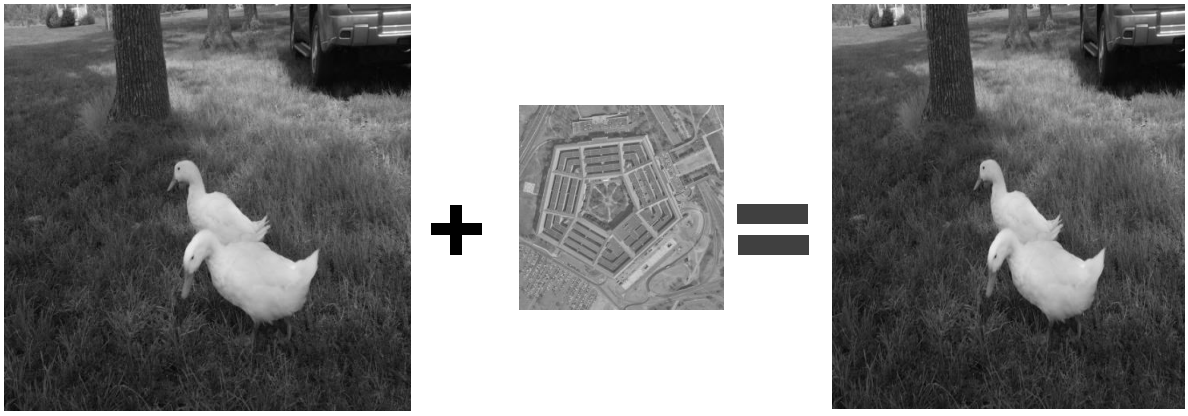
- Viện Công Nghệ Thông Tin - Viện khoa học Quốc Gia,

- Đại Học Công Nghệ - Thuộc Đại Học Quốc Gia Hà Nội,
- Đại Học Bách Khoa Thành phố Hồ Chí Minh,
- Tổng cục I, Tổng Cục V Bộ Công An
- ...

Các công trình nghiên cứu này đã thu được một số kết quả tốt. Đa số tập trung chủ yếu nghiên cứu các kỹ thuật giấu tin và cách thức giấu tin tối ưu nhất mà không làm thay đổi bản chất của đối tượng được giấu tin, tức là các nội dung thông tin được giấu trong đa phương tiện khó bị phát hiện bởi kỹ thuật thông thường. Còn các kỹ thuật phát hiện cũng đang được tập trung rất cao xong chưa có kết quả nào đáng ghi nhận. Trong khi đó trên thế giới các kỹ thuật này đang phát triển rất sôi nổi và đã có nhiều công trình công bố trên các tạp trí và khoa học quốc tế.

## 1.2 Thuật toán Least Significant Bit LSB và LSB matching revisited

Bit LSB trong ảnh là bit có ảnh hưởng ít nhất tới việc quyết định tới màu sắc của mỗi điểm ảnh, vì vậy khi ta thay đổi bit ít quan trọng của một điểm ảnh thì màu sắc của mỗi điểm ảnh mới sẽ tương đối gần với điểm ảnh cũ như hình 1.6. 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 để giấu tin… 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.



(a) (b) (c)

Hình 1. 6 (a) đối tượng chứa (b) đối tượng thông tin mật (c) đối tượng sau khi nhúng

### 1.2.1 Thuật toán LSB

Thuật toán Least Significant Bit (LSB) [3] là một trong những kỹ thuật đơn giản của những phương pháp phổ biến của ẩn dữ liệu.

Gọi C là ảnh chứa- ảnh xám 8 bit kích thước $M_c x N_c$ được biểu diễn như sau:

$$c = \ x_{i,j}\ 0 \le i \le M_c, 0 \le j \le N_c \in \ 0,1, \dots ,255 \quad (1)$$

Gọi M là n-bit mật được biểu diễn như sau:

$$M = \ m_i\ 0 \le i \le n, m_i \in \ 0,1 \quad (2)$$

Giả sử nhúng chuỗi bit mật *M* vào k-bit-phải nhất của ảnh C. Đầu tiên, chuỗi M cần được sắp xếp lại thành ảnh k-bit $M'$ như sau:

$$\mathrm{M}' = \{m_i' \mid 0 \le i \le n, m_i \in \{0,1,\dots,2^k - 1\}\}(3)$$

Với $n' < M_c x N_c$ . Ánh xạ giữa chuỗi n-bit $M=\{m_i\}$ và chuỗi nhúng được định nghĩa như sau:

$$m_i' = \sum_{j=0}^{k-1} m_{i*k+j} * 2^{k-1-j} \ (4)$$

Tiếp theo, một tập con *n'* pixel $\{x_{l_1}, x_{l_2}, \dots, x_{l_n},\}$ được chọn ra từ ảnh chứa *C* theo một thứ tự cho trước. Quá trình nhúng sẽ được thực hiện bằng cách thay thế k-bit-phải-nhất của $x_{l_i}$ băng $m_i'$ . Về mặt toán học, giá trị $x_{l_i}$ của sẽ lưu k-bit $m_i'$ bằng cách thay đổi giá trị để hình thành giá trị sau khi nhúng$x'_{l_i}$ như sau:

$$x'_{l_i} = x_{l_i} - x_{l_i} \bmod 2^{\mathrm{k}} + m_i'(5)$$

Ở quá trình rút trích, cho ảnh đã nhúng *S*, chuỗi bit mật sẽ được rút trích dựa vào chuỗi các pixel đã được nhúng. Cụ thể chuỗi mật $m_i'$sẽ được phục hồi như sau:

$$m_i' = x'_{l_i} \bmod 2^{\mathrm{k}}(6)$$

Giả sử tất cả các pixel trong ảnh đều được dùng để nhúng bít mật bằng phương pháp LSB. Trong tình huống xấu nhất, giá trị PSNR (độ do sự khác biệt giữa 2 ảnh) của ảnh sau khi nhúng so với ảnh chứa sẽ như bảng 1.2

Bảng 1. 2 PSNR trong trường hợp xấu nhất

| K | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| PNSR | 48.13 | 38.59 | 31.23 | 24.61 | 18.30 |

PNSR trong trường hợp xấu nhất được tính bằng công thức [2]:

$$PNSR_{worst} = 10 * log_{10} \frac{255^2}{(2^k - 1)^2} dB \ (7)$$

### 1.2.1 Thuật toán LSB matching revisited

LSB matching [4] là thuật toán cộng hoặc trừ giá trị của pixel trong ảnh chứa một lượng tương ứng để LSB của pixel khớp với chuỗi bit mật. Và để cho số lượng pixel được trừ và cộng bằng nhau giúp tránh hiện tượng mất cân bằng của thuật toán LSB[5]. Bên cạnh đó, thuật toán cũng sẽ kiểm tra để đảm bảo giá trị pixel vẫn nằm trong ngưỡng màu.

LSB matching revisited (LSB-MR) [4] là một cải tiến của LSB matching, cho phép nhúng cùng một lượng như LSB matching nhưng LSB-MR có tính vô hình cao hơn.

LSB-MR sẽ thực hiện nhúng trên 2 pixel cùng lúc. Gọi giá trị mức xám của 2 pixel là $x_i$ và $x_{i+1}$. Sau khi nhúng, giá trị của bit mật thứ $i$ ($m_i$) sẽ tương ứng với LSB của pixel thứ $i$ ($y_i$) trong ảnh sau khi nhúng. Giá trị của bit mật thứ $i+1$ ($m_{i+1}$) sẽ được tính dựa vào $y_i$ và $y_{i+1}$ theo hàm nhị phân sau:

$$f(y_i, y_{i+1}) = LSB\left(\frac{y_i}{2} + y_{i+1}\right) \quad 8$$

Hàm $f(y_i,y_{i+1})$ có 2 tính chất sau:

$$f(l-1,n) \neq f(l+1,n), \forall l,n \in Z \quad 9$$

$$f(l,n) \neq f(l,n+1), \forall l,n \in Z \quad 10$$

Nhờ 2 tính chất (9)(10) LSB-MR sẽ thiết lập $y_i$và $y_{i+1}$ để $f(y_i,y_{i+1})$ cho ra kết quả mong muốn. Thuật toán nhúng được thực hiện trên từng cặp pixel của ảnh chứa như sau:

**Input:** một cặp pixel $x_i$ , $x_{i+1}$ và 2 bit mật $m_i$ , $m_{i+1}$

**Output:** cặp pixel đã nhúng $y_i$ và $y_{i+1}$

$if\ \mathrm{m_i} = \mathrm{LSB}\ \ \mathrm{x_i}$

$if\ \mathrm{m_{i+1}} \neq \mathrm{f}\ \mathrm{x_i}, \mathrm{x_{i+1}}$

$y_{i+1} = x_{i+1} \pm 1$

Else

$y_{i+1} = x_{i+1}$

$end$

$y_i = x_i$

Else

$if\ \mathrm{m_{i+1}} = \mathrm{f}\ \mathrm{x_i}, \mathrm{x_{i+1}}$

$y_i = x_i - 1$

Else

$y_i = x_i + 1$

$end$

$y_{i+1} = x_{i+1}$

$end$

Thuật toán không áp dựng trên những điểm bảo hào (những điểm ảnh có giá trị cực tiểu hoặc cực đại). Bảng 1.3 thể hiện ví dụ của việc nhúng bit mật vào ảnh số lượng kỳ vọng bit bị thay đổi trên mỗi pixel được tính theo công thức:

$$\frac{P\ x_i \neq m_i\ + P\ x_i = m_i\ P(m_{i+1} \neq f\ x_i, x_{i+1}\ )}{2} \quad (11)$$

ở đây P() là hàm xác suất. Nếu số lượng bit 0 và 1 gần bằng nhau trong chuỗi mật và trong ảnh chứa và chúng độc lập với nhau thì $P\ x_i \neq m_i\ = P\ x_i = m_i\ = P(m_{i+1} \neq f\ x_i, x_{i+1}\ = 0.5$ . Do đó, số lượng kỳ vọng bit bị thay đổi trên mỗi pixel là 0.375 [6].

Bảng 1. 3ví dụ nhúng LSB-MR

| $x_i$ | $x_{i+1}$ | $m_i$ | $m_{i+1}$ | $y_i$ | $y_{i+1}$ |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 2 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 hoặc 2 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 0 | 0 | 0 | 2 |
| 1 | 2 | 0 | 1 | 2 | 2 |
| 1 | 2 | 1 | 0 | 1 | 2 |
| 1 | 2 | 1 | 1 | 1 | 1 hoặc 3 |
| 2 | 1 | 0 | 0 | 2 | 1 |
| 2 | 1 | 0 | 1 | 2 | 0 hoặc 2 |
| 2 | 1 | 1 | 0 | 3 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 0 | 0 | 2 | 1 hoặc 3 |
| 2 | 2 | 0 | 1 | 2 | 2 |
| 2 | 2 | 1 | 0 | 1 | 2 |
| 2 | 2 | 1 | 1 | 3 | 2 |

Với thuật toán phát hiện dữ liệu ẩn HCF COM – phương pháp nổi tiếngnhất đối với thuật toán họ LSB[7] cho thấy LSB-MR cho hiệu quả tốt hơn LSB matching, vì số lượng bit bị thay dổi trên mỗi pixel của LSB matching là 0.5 cao hơn LSB-MR 0.375.

### 1.3 Nén Ảnh

Các tập tin ảnh số hóa thường có rất nhiều thông tin dư thừa, và nén ảnh là một kỹ thuật mã hóa để giảm số lượng các bit dữ liệu cần thiết để biểu diễn ảnh cũng như loại bỏ các thông tin dư thừa. Nén ảnh thực hiện được là do một thực tế: thông tin trong bức ảnh không phải là ngẫu nhiên mà có trật tự , tổ chức.Vì thế nếu bóc tách được tính trật tự, cấu trúc đó thì sẽ biết phần thông tin nào quan trọng nhất trong bức ảnh để biểu diễn với số lượng ít bit hơn so với ảnh gốc mà vẫn đảm bảo tính đầy đủ của thông tin. Ở bên nhận quá trình giải mã sẽ tổ chức, sắp xếp lại được bức ảnh xấp xỉ gần chính xác so với ảnh gốc nhưng vẫn thỏa mãn chất lượng yêu cầu.

Ví dụ:

- Ảnh đa cấp xám hay ảnh 256 màu(8bit) có kích thước 256 x 256, cần 524288 bit lưu trữ.
- Ảnh màu RGB (24 bit/điểm ảnh ) cùng độ phân giải như vậy cần hơn 1 triệu bit để lưu trữ .
- Một phim âm bản có kích thước 24 × 36 mm (35 mm) chia bằng các khoảng cách nhau 12 μm, vào khoảng 3000 × 2000 điểm, 8 bit / pixel, yêu cầu 48 triệu bit cho lưu giữ ảnh và 83 phút để truyền.

Nén ảnh đạt được bằng cách loại bỏ các phần dư thừa trong ảnh đã được số hoá. Dư thừa có thể là dư thừa thông tin về không gian, dư thừa về cấp xám hay dư thừa về thời gian:

- Dư thừa thông tin về không gian : trong một bức ảnh luôn tồn tại sự tương quan giữa các điểm ảnh cạnh nhau.
- Dư thừa thông tin về cấp xám :là dư thừa dựa vào sự tương quan giữa các màu sắc cạnh nhau.
- Dư thừa thông tin về thời gian : Trong một chuỗi ảnh video, tồn tại sự tương quan giữa các điểm ảnh của các frame khác nhau .

Trong các hệ thống nén, tỉ số nén ($H_c$) chính là tham số quan trọng đánh giá khả năng nén của hệ thống công thức được tính như sau:

$$H_c = \frac{1}{r * \%}$$

với $r$ là tỷ số nén được định nghĩa:

$$r = \frac{kích\ thước\ dữ\ liệu\ gốc}{kích\ thước\ dữ\ liệu\ nén}$$

Như vậy hiệu suất nén = (1- tỷ lệ nén)*100%.

Đối với ảnh tĩnh, kích thước chính là số bit biểu diễn toàn bộ bức ảnh.

Đối với ảnh video, kích thước chính là số bit để biểu diễn một khung hình video (video frame).

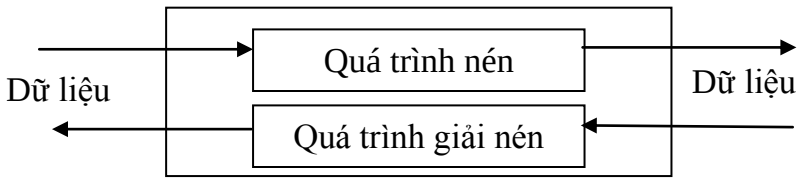
### 1.3.1 Quá trình nén và giải nén:

Gồm 2 công đoạn :

**Nén** : dữ liệu gốc qua bộ mã hoá dữ liệu , bộ mã hoá này thực hiện nén dữ liệu đến một mức thích hợp cho việc lưu trữ và truyền dẫn thông tin. Quá trình này sẽ thực hiện việc loại bỏ hay cắt bớt những dư thừa của ảnh để thu được thông tin cần thiết nhưng vẫn đảm bảo được chất lượng ảnh.

**Giải nén** : dữ liệu nén đi qua bộ giải mã dữ liệu, bộ giải mã sẽ thực hiện giải nén để thu được dữ liệu gốc ban đầu.Việc giải nén này thường phải dựa vào các thông tin đi kèm theo dữ liệu nén ,tuỳ thuộc vào kiểu nén hay phương pháp nén mà dữ liệu giải nén có được hoàn toàn giống với dữ liệu gốc ban đầu hay không.

Tóm lại quá trình nén và giải nén dữ liệu có thể mô tả một cách tóm tắt theo sơ đồ dưới đây:




Hình 1. 7Quá trình nén và giải nén

### 1.3.2 Phân loại các phương pháp nén

**a) Theo nguyên lý nén**

- *Nén bảo toàn thông tin (losses compression):* bao gồm các phương pháp nén mà sau khi giải nén sẽ thu được chính xác dữ liệu gốc.Tuy nhiên nén bảo toàn thông tin chỉ đạt hiệu quả nhỏ so với phương pháp nén không bảo toàn thông tin.

- *Nén không bảo toàn thông tin (lossy compression):* bao gồm các phương pháp nén sau khi giải nén sẽ không thu được dữ liệu như bản gốc. Các phương pháp này được gọi là "tâm lý thị giác" đó là lợi dụng tính chất của mắt người chấp nhận một số vặn xoắn trong ảnh khi khôi phục lại.Phương pháp này luôn đem lại hiệu quả cao do loại bỏ đi những thông tin dư thừa không cần thiết.

**b) Theo cách thức thực hiện nén**

- *Phương pháp không gian(Spatial Data Compression ):* các phương pháp này thực hiện nén bằng cách tác động trực tiếp lên việc lấy mẫu của ảnh trong miền không gian.

- *Phương pháp sử dụng biến đổi (Transform Coding):* gồm các phương pháp tác động lên sự biến đổi của ảnh gốc chứ không tác động trực tiếp.

**c) Phân loại dựa vào lý thuyết mã hóa**

- *Các phương pháp nén thế hệ thứ nhất:* gồm các phương pháp có mức độ tính toán đơn giản như lấy mẫu , gán từ mã,….

- *Các phương pháp nén thế hệ thứ hai:* gồm các phương pháp dựa vào mức độ bão hoà của tỷ lệ nén bằng cách sử dụng các phép toán tổ hợp đầu ra một cách hợp lý hoặc sử dụng biểu diễn ảnh như : phương pháp kim tự tháp Laplace, phương pháp dựa vào vùng gia tăng, phương pháp tách hợp.

**d) Cách phân loại của Anik.k.jain**

Theo cách của Jain, các phương pháp nén gồm 4 họ chính:

Phương pháp điểm.

Phương pháp dự đoán.

Phương pháp dựa vào biến đổi.

Các phương pháp tổ hợp (Hybrid).

Thực ra cách phân loại này là chia nhỏ của cách phân loại thứ ba và dựa vào cơ chế thực hiện nén. Xét một cách kỹ lưỡng nó cũng tương đương cách phân loại thứ ba.

### 1.3.3 Một số phương pháp nén thông tin

#### 1.3.3.1 Phương pháp mã hoá độ dài loạt (Run-Length Encoding)

Loại dư thừa đơn giản nhất trong một tập tin là các đường chạy dài gồm các kí tự lặp lại, điều này thường thấy trong các tập tin đồ hoạ bitmap, các vùng dữ liệu hằng của các tập tin chương trình, một số tập tin văn bản...

Ví dụ, xét chuỗi sau

*AAAABBBAABBBBBCCCCCCCCDABCBAAABBBBCCCD*

Chuỗi này có thể được mã hoá một cách cô đọng hơn bằng cách thay thế chuỗi kí tự lặp lại bằng một thể hiện duy nhất của kí tự lặp lại cùng với một biến đếm số lần kí tự đó được lặp lại. Ta muốn nói rằng chuỗi này gồm bốn chữ A theo sau bởi ba chữ B rồi lại theo sau bởi hai chữ A, rồi lại theo sau bởi năm chữ B... Việc nén một chuỗi theo phương pháp này được gọi là mã hoá độ dài loạt. Khi có những loạt dài, việc tiết kiệm có thể là đáng kể. Có nhiều cách để thực hiện ý tưởng này, tuỳ thuộc vào các đặc trưng của ứng dụng (các loạt chạy có khuynh hướng tương đối dài hay không ? Có bao nhiêu bit được dùng để mã hoá các kí tự đang được mã ?).

Nếu ta biết rằng chuỗi của chúng ta chỉ chứa các chữ cái, thì ta có thể mã hoá biến đếm một cách đơn giản bằng cách xen kẽ các con số với các chữ cái. Vì vậy chuỗi kí tự trên được mã hoá lại như sau:

*4A3BAA5B8CDABCB3A4B3CD*

Với "4A" có nghĩa là "bốn chữ A"... Chú ý là không đáng để mã hoá các loạt chạy có độ dài 1 hoặc 2 vì cần đến hai kí tự để mã hoá.

Đối với các tập tin nhị phân một phiên bản được tinh chế của phương pháp này được dùng để thu được sự tiết kiệm ĐÁNG KỂ. Ý tưởng ở đây là lưu lại các độ dài

loạt, tận dụng sự kiện các loạt chạy thay đổi giữa 0 và 1 để tránh phải lưu chính các số 0 và 1 đó. Điều này giả định rằng có một vài loạt chạy ngắn (Ta tiết kiệm các bit trên một loạt chạy chỉ khi độ dài của đường chạy là lớn hơn số bit cần để biểu diễn chính nó trong dạng nhị phân), nhưng khó có phương pháp mã hoá độ dài loạt nào hoạt động thật tốt trừ khi hầu hết các loạt chạy đều dài. Việc mã hoá độ dài loạt cần đến các biểu diễn riêng biệt cho tập tin và cho bản đã được mã hoá của nó, vì vậy nó không thể dùng cho mọi tập tin, điều này có thể hoàn toàn bất lợi, ví dụ, phương pháp nén tập tin kí tự đã được đề nghị ở trên sẽ không dùng được đối với các chuỗi kí tự có chứa số. Nếu những kí tự khác được sử dụng để mã hoá các số đếm, thì nó sẽ không làm việc với các chuỗi chứa các kí tự đó. Giả sử ta phải mã hoá bất kì kí tự nào từ một bảng chữ cái cố định bằng cách chỉ dùng các kí tự từ bảng chữ cái đó. Để minh hoạ, giả sử ta phải mã hoá bất kì một chuỗi nào từ một chữ cái đó, ta sẽ giả định rằng ta chỉ có 26 chữ cái trong bảng chữ cái (và cả khoảng trống) để làm việc.

Để có thể dùng vài chữ cái để biểu diễn các số và các kí tự khác biểu diễn các phần tử của chuỗi sẽ được mã hoá, ta phải chọn một kí tự được gọi là kí tự "Escape". Mỗi một sự xuất hiện của kí tự đó báo hiệu rằng hai chữ cái tiếp theo sẽ tạo thành một cặp (số đếm, kí tự) với các số đếm được biểu diễn bằng cách dùng kí tự thứ i của bảng chữ cái để biểu diễn số i. Vì vậy, chuỗi ví dụ của chúng ta sẽ được biểu diễn như sau với Q được xem là các kí tự "Escape"

*QDABBBAABQHCDABCBAAAQDBCCCD*

Tổ hợp của kí tự "Escape", số đếm và một kí tự lặp lại được gọi là một dãy Escape. Chú ý rằng không đáng để mã hoá các đường chạy có chiều dài ít hơn bốn kí tự, vì ít nhất là cần đến ba kí tự để mã hoá bất kì một loạt chạy nào. Trong trường hợp bản thân kí tự "Escape" xuất hiện trong dãy kí tự cần mã hoá ta sử dụng một dãy "Escape" với số đếm là 0 (kí tự space) để biểu diễn kí tự "Escape". Như vậy trong trường hợp kí tự "Escape" xuất hiện nhiều thì có thể làm cho tập tin nén phình to hơn trước.

Các loạt chạy dài có thể được cắt ra để mã hoá bằng nhiều dãy Escape, ví dụ, một loạt chạy gồm 51 chữ A sẽ được mã hoá như *QZAQYA* bằng cách dùng trên.
Phương pháp mã hoá độ dài loạt thường được áp dụng cho các tập tin đồ hoạ bitmap vì ở đó thường có các mảng lớn cùng màu được biểu diễn dưới dạng bitmap là các chuỗi bit có đường chạy dài. Trên thực tế, nó được dùng trong các tập tin .PCX, .RLE.

#### *1.3.3.2* Phương pháp mã hoá Huffman

Các tập tin của máy tính được lưu dưới dạng các kí tự có chiều dài không đổi là 8 bits. Trong nhiều tập tin, xác suất xuất hiện các kí tự này là nhiều hơn các kí tự khác, từ đó ta thấy ngay rằng nếu chỉ dùng một vài bit để biểu diễn cho các kí tự có xác suất xuất hiện lớn và dùng nhiều bit hơn để biểu diễn cho các kí tự có xác suất xuất hiện nhỏ thì có thể tiết kiệm được độ dài tập tin một cách đáng kể. Ví dụ, để mã hoá một chuỗi như sau:

*"ABRACADABRA"*

Nếu mã hoá chuỗi trên trong dạng mã nhị phân 5 bit ta sẽ có dãy bit sau:

00001000101001000001000110000100100000100010100100000 1

Để giải mã thông điệp này, chỉ đơn giản là đọc ra 5 bits ở từng thời điểm và chuyển đổi nó tương ứng với việc mã hoá nhị phân đã được định nghĩa ở trên. Trong mã chuẩn này, chữ D xuất hiện chỉ một lần sẽ cần số lượng bit giống chữ A xuất hiện nhiều lần.

Ta có thể gán các chuỗi bit ngắn nhất cho các kí tự được dùng phổ biến nhất, giả sử ta gán: A là 0, B là 1, R là 01, C là 10 và D là 11 thì chuỗi trên được biểu diễn như sau:

0 1 01 0 10 0 11 0 1 01 0

Ví dụ này chỉ dùng 15 bits so với 55 bits như ở trên, nhưng nó không thực sự là một mã vì phải lệ thuộc vào khoảng trống để phân cách các kí tự. Nếu không có dấu phân cách thì ta không thể giải mã được thông điệp này. Ta cũng có thể chọn các từ mã sao cho thông điệp có thể được giải mã mà không cần dấu phân cách, ví dụ như: A là

11, B là 00, C là 010, D là 10 và R là 011, các từ mã này gọi là các từ mã có tính prefix (Không có từ mã nào là tiền tố của từ mã khác). Với các từ mã này ta có thể mã hoá thông điệp trên như sau:

110001111010111010110001111

Với chuỗi đã mã hoá này ta hoàn toàn có thể giải mã được mà không cần dấu phân cách. Nhưng bằng cách nào để tìm ra bảng mã một cách tốt nhất ? Vào năm 1952, D.Huffman đã phát minh ra một cách tổng quát để tìm ra bảng mã này một cách tốt nhất.

Bước đầu tiên trong việc xây dựng mã Huffman là đếm số lần xuất hiện của mỗi kí tự trong tập tin sẽ được mã hoá.

Bước tiếp theo là xây dựng một cây nhị phân với các tần số được chứa trong các nút. Hai nút có tần suất bé nhất được tìm thấy và một nút mới được tạo ra với hai nút con là các nút đó với giá trị tần số của nút mới bằng tổng tần suất của hai nút con. Tiếp theo hai nút mới với tần số nhỏ nhất lại được tìm thấy và một nút mới nữa lại được tạo ra theo cách trên. Lặp lại như vậy cho đến khi tất cả các nút được tổ hợp thành một cây duy nhất.

Sau khi có cây nhị phân, bảng mã Huffman được phát sinh bằng cách thay thế các tần số ở nút đáy bằng các kí tự tương ứng.

*Ưu điểm:* của phương pháp mã hoá Huffman là đạt được hệ số nén cao (Hệ số nén tuỳ thuộc vào cấu trúc của các tập tin). Phương pháp thực hiện tương đối đơn giản, đòi hỏi ít bộ nhớ, có thể xây dựng dựa trên các mảng bé hơn 64KB.

*Nhược điểm*: của nó là phải chứa cả bảng mã vào tập tin nén thì phía nhận mới có thể giải mã được do đó hiệu suất nén chỉ cao khi ta thực hiện nén các tập tin lớn. Nhược điểm của phương pháp nén này có thể khắc phục bằng cách thực hiện nén một lần nhiều tập tin chuẩn bị truyền, làm như vậy coi như chúng ta đang thực hiện nén một tập tin lớn.

### 1.3.3.3Phương pháp nén LZW

Phương pháp nén LZW được phát minh bởi Lempel - Zip và Welch. Nó hoạt động dựa trên một ý tưởng rất đơn giản là người mã hoá và người giải mã cùng xây dựng bản mã.

Nguyên tắc hoạt động của nó như sau:

+ Một xâu kí tự là một tập hợp từ hai kí tự trở lên.

+ Nhớ tất cả các xâu kí tự đã gặp và gán cho nó một dấu hiệu (token) riêng.

+ Nếu lần sau gặp lại xâu kí tự đó, xâu kí tự sẽ được thay thế bằng dấu hiệu của nó.

Phần quan trọng nhất của phương pháp nén này là phải tạo một mảng rất lớn dùng để lưu giữ các xâu kí tự đã gặp (Mảng này được gọi là "Từ điển"). Khi các byte dữ liệu cần nén được đem đến, chúng liền được giữ lại trong một bộ đệm chứa (Accumulator) và đem so sánh với các chuỗi đã có trong "từ điển". Nếu chuỗi dữ liệu trong bộ đệm chứa không có trong "từ điển" thì nó được bổ sung thêm vào "từ điển" và chỉ số của chuỗi ở trong "từ điển" chính là dấu hiệu của chuỗi. Nếu chuỗi trong bộ đệm chứa đã có trong "từ điển" thì dấu hiệu của chuỗi được đem ra thay cho chuỗi ở dòng dữ liệu ra. Có bốn qui tắc để thực hiện việc nén dữ liệu theo thuật toán LZW là:

Qui tắc 1: 256 dấu hiệu đầu tiên được dành cho các kí tự đơn (0 - 0ffh).

Qui tắc 2: Cố gắng so sánh với "từ điển" khi trong bộ đệm chứa đã có nhiều hơn hai kí tự.

Qui tắc 3: Các kí tự ở đầu vào (Nhận từ tập tin sẽ được nén) được bổ sung vào bộ đệm chứa đến khi chuỗi kí tự trong bộ đệm chứa không có trong "từ điển".

Qui tắc 4: Khi bộ đệm chứa có một chuỗi mà trong "từ điển" không có thì chuỗi trong bộ đệm chứa được đem vào "từ điển". Kí tự cuối cùng của chuỗi kí tự trong bộ đệm chứa phải ở lại trong bộ đệm chứa để tiếp tục tạo thành chuỗi mới

Ví dụ: Các bước để mã hoá chuỗi "!BAN!BA!BAA!BAR!" như sau (Bảng 1. 4):
- bước 1: Kí tự thứ nhất ‘!’ được cất vào bộ đệm chứa để chuẩn bị tạo nên một chuỗi.

- bước 2: Kí tự thứ hai ‘B’ nối thêm vào sau kí tự !. Vì trong "từ điển" chưa có chuỗi "!B" nên chuỗi này được thêm vào "từ điển" và được gán dấu hiệu là 100h (Vì từ 000h đến 0ffh được dành riêng cho các kí tự đơn: Qui tắc 1). ‘!’ được gửi ra còn ‘B’ phải ở lại trong bộ đệm chứa.

Bảng 1. 4 Các bước để mã hóa chuỗi

| STT | Bộ đệm chứa | Dữ liệu vào | Dữ liệu ra | Từ điển |
|---|---|---|---|---|
| 1 | - | ! | - | - |
| 2 | ! | B | ! | 100h=!B |
| 3 | B | A | B | 101h=BA |
| 4 | A | N | A | 102h=AN |
| 5 | N | ! | N | 103h=N! |
| 6 | ! | B | - | - |
| 7 | !B | A | <100h> | 104h=!BA |
| 8 | A | ! | A | 105h=A! |
| 9 | ! | B | - | - |

- bước 3: Kí tự thứ ba ‘A’ thêm vào sau ‘B’. Chuỗi "BA" cũng chưa có trong "từ điển" nên nó được thêm vào "từ điển" và gán dấu hiệu là 101h. ‘A’ ở lại trong bộ đệm chứa còn ‘B’ được gửi ra.

- bước 4: Kí tự thứ tư ‘N’ thêm vào sau ‘A’ tạo thành chuỗi "AN" cũng chưa có trong "từ điển" nên được thêm vào "từ điển" và có dấu hiệu là 102h. ‘N’ ở lại trong bộ đệm chứa còn ‘A’ được gửi ra.

- bước 5: Kí tự thứ năm ‘!’ thêm vào sau ‘N’ để tạo thành chuỗi "N!", "N!" được thêm vào "từ điển" với dấu hiệu là 103h. ‘!’ ở lại còn ‘N’ được gửi ra.

- bước 6: Kí tự thứ sáu ‘B’ thêm vào sau ‘!’. Lần này thì chuỗi "B!" đã có trong "từ điển" nên không có kí tự nào được gửi ra. "B!" tiếp tục ở lại trong "từ điển" để tạo ra chuỗi mới.

- bước 7: Kí tự thứ bảy ‘A’ thêm vào sau ‘B’ để tạo thành chuỗi "B!A", do "B!A" không có trong "từ điển" nên nó được thêm vào "từ điển" và gán dấu hiệu là 104h đồng thời dấu hiệu 100h được gửi ra thay cho "B!" (Qui tắc 4). A tiếp tục ở lại trong bộ đệm chứa để tạo thành chuỗi mới.

Các bước trên cứ thế tiếp tục cho đến khi hết tập tin cần nén. Việc giảm kích thước chỉ thực sự bắt đầu tại bước 7 khi mà một dấu hiệu 12 bits là <100h> được gửi ra thay cho hai byte "B!".

Trong thuật toán nén này, phần lớn thời gian khi bắt đầu nén chủ yếu mất vào việc tạo "từ điển". Khi "từ điển" đủ lớn, xác suất gặp chuỗi ở bộ đệm chứa trong "từ điển" tăng lên và càng nén được nhiều hơn. Một điều cần chú ý ở đây là mỗi một dấu hiệu, ta phải lưu một chuỗi trong "từ điển" để so sánh. Vì dấu hiệu được biểu diễn bằng một số 12 bits nên "từ điển" sẽ có 4096 lối vào, khi tăng số bit để biểu diễn dấu hiệu lên thì hiệu quả nén sẽ tốt hơn nhưng lại bị giới hạn bởi bộ nhớ của máy tính. Vì dụ, khi dùng 16 bits để biểu diễn một dấu hiệu thì "từ điển" phải có đến 65536 lối vào, nếu mỗi lối vào có khoảng 20 kí tự thì "từ điển" phải lớn khoảng 1,2 MB. Với một từ điển có dung lượng như vậy rất khó có thể thực hiện trên các máy tính PC hoạt động dưới hệ điều hành DOS vì giới hạn của một đoạn (Segment) là 64KB. Ưu điểm của phương pháp nén LZW là bên nhận có thể tự xây dựng bảng mã mà không cần bên gửi phải gửi kèm theo bản tin nén.

*Ưu điểm:* là Thuật toán nén LZW có hệ số nén tương đối cao, trong tập tin nén không cần phải chứa bảng mã.

*Nhược điểm:* của thuật toán này là tốn nhiều bộ nhớ, khó thực hiện dựa trên các mảng đơn giản (bé hơn 64KB).

### 1.3.4 Thuật toán nén Fractal

Thuật toán nén Fractal [8] là kỹ thuật nén ảnh dựa trên lý thuyết các hệ lặp Iterated Function System (IFS) [9] của hình học Fracal, được đưa ra bởi Barnsley [10] and

Jacquin [11]. Phương pháp nén này cho phép thu được các tỉ số nén cao và ảnh thu được độc lập về độ phân giải.

Sự ra đời của lý thuyết hình học fractal là kết quả của nhiều thập kỷ nỗ lực giải quyết các vấn đề nan giải trong nhiều ngành khoa học chính xác, đặc biệt là vật lý và toán học. Một cách cụ thể, lý thuyết hình học fractal được xây dựng dựa trên 2 vấn đề lớn được quan tâm ở những thập niên đầu thế kỷ 20. Các vấn đề đó bao gồm:

· Tính hỗn độn của các quá trình phát triển có quy luật trong tự nhiên.

· Sự mở rộng khái niệm số chiều và độ đo trong lý thuyết hình học Euclide cổ điển.

Năm 1979, nhà toán học Benoit Mandelbrot áp dụng tập Mandelbrot đầy kì ảo lên máy tính. Ông đã khám phá ra một lãnh vực hình học mới đầy thú vị cho phép phản ánh thế giới thực một cách tự nhiên hơn so với hình học Euclid. Tất cả những hình ảnh mà ta thường gặp trong tự nhiên như : núi, mây, sông, nước… nay máy tính đã có khả năng mô tả được bằng phương pháp fractal. Để thấy rõ hơn sức mạnh của fractal trong mô tả tự nhiên bạn có thể xem thêm bộ sưu tập ảnh fractal kèm theo.

Trong giai đoạn này B. Mandelbrot và các nhà toán hoc khác như A.Douady và J.Hubbard đã đặt nền móng và phát triển lí thuyết cho hình học Fractal. Các kết quả đạt được chủ yếu tập trung ở các tính chất của các cấu trúc fractal cơ sở như tập Maldenbrot và tập Julia. Ngoài ra các ngiên cứu khác cũng cố gắng tìm kiếm mối quan hệ giữa các cấu trúc này, ví dụ như mối quan hệ giữa Maldenbrot và Julia.

Dựa trên các công trình của Maldenbrot (trong những năm 1976, 1979, 1982) và Hutchinson(1981), vào các năm 1986,1988 Michael F.Barnsley và M.Begger đã phát triển lý thuyết biểu diễn các đối tượng tự nhiên dựa trên cơ sở lý thuyết về các hệ hàm lặp IFS. Các hệ hàm lặp này bao gồm một bộ hữu hạn các phép biến đổi affine cho phép với sự giúp đỡ của máy tính tạo nên hình ảnh của các đối tượng trong tự nhiên. Theo lý thuyết này hình học Euclide cổ điển rất có hiệu lực trong việc biểu diễn các đối tượng nhân tạo như một tòa nhà, một cổ máy nhưng lại hoàn toàn không thích hợp cho

việc biểu diễn các đối tượng của thế giới thực vì đòi hỏi một lượng quá lớn các đặc tả cần có. Nếu như trong hình học Euclide các yếu tố cơ sở là đường thẳng, đường tròn, hình vuông,… thì lý thuyết IFS mởrộng hình học cổ điển với các yếu tố cơ sở mới là vô số thuật toán để vẽ nên các fractal củatự nhiên.

Ngoài các công trình có tính chất lý thuyết, hình học fractal còn được bổ sung bởi nhiềunghiên cứu ứng dụng lý thuyết vào khoa học máy tính và các khoa học chính xác khác, vídụ như dựa trên lý thuyết IFS, Barnsley đã phát triển lý thuyết biến đổi fractal áp dụng vàocông nghệ nén ảnh tự động trên máy tính, là một lĩnh vực đòi hỏi những kỹ thuật tiên tiếnnhất của tin học hiện đại.

Hiện nay nhiều vấn đề, về lý thuyết fractal vẫn đang được tiếp tục nghiên cứu. Một trongnhững vấn đề lớn đang được quan tâm là bài toán về các độ đo đa fractal(multifractalmeasurement) có liên quan đến sự mở rộng các khái niệm số chiều fractal với đối tượng fractaltrong tự nhiên, đồng thời cũng liên quan đến việc áp dụng các độ đo fractal trong các ngànhkhoa học tự nhiên

***Cơ sở lý thuyết:***

Đầu tiên ta xem xét 1 ánh xạ co $W_i$ mỗi hệ số ánh xạ $s<1$, và áp dụng với không gian ma trận $X$ vào chính nó với $i= 1,2, .... ,N$. Thiết lập như vậy gọi là hệ thống lập IFS Sử dụng IFS này để xây dựng một bản đồ $W$ từ không gian $H$ tập con khác rỗng của $X$ vào chính theo định nghĩa sau:

$$W\ B\ = \bigcup_{i}^{N} w_i\ B \quad với i\ \forall\ B \in H$$

Trong đó W là một hình ảnh được co lại với hệ số co lại s <1, và ma trận h trong H được định nghĩa là:

$$h\ (A, B) = \text{Max}\ \{d\ (A, B), d\ (B, A)\} \quad với\ \forall A, B \in H,$$

và

$$d\ (A, B) = \text{Max}\ \{d\ (x, B): x \in A\} \qquad \text{với}\ \forall A, B \in H$$

*với*

$$d\,(x, B) = \text{Min}\,\{d\,(x, y): y \in B\} \quad với \quad \forall x \in X\,,\ B \in H.$$

Các khái niệm cơ bản sử dụng trong nén Fractal là phép biến đổi Affine, biến đổi co lại:

- *Phép biến đổi affine*:

Là phép biến đổi thông dụng trong đồ họa máy tính, một biến đổi affine trong không gian $R^n$ là chức năng bao gồm chuyển đổi tuyến tính và chuyển đổi dịch trong không gian $R^n$. Ví dụ trong không gian 2 chiều $R^2$ có dạng :

$$w_i \begin{matrix} x \\ y \end{matrix} = \begin{matrix} a_i & b_i \\ c_i & d_i \end{matrix} \begin{matrix} x \\ y \end{matrix} + \begin{matrix} e_i \\ f_i \end{matrix}$$

Với các tham số a,b,c,d tạo thành chuyển đổi tuyến tính, còn e,f là khoảng các dịch chuyển của x và y tương ứng.

- *Biến đổi co lại:*

Một *w* chuyển đổi được cho là co lại nếu đối với bất kỳ hai điểm *P1, P2,* khoảng cách

$$d\,(w\,(P1), w\,(P2)) < s\,d\,(P1, P2)$$

Đối với một số *s*<1, trong đó d = khoảng cách.Công thức này cho biết việc áp dụng một bản đồ luôn mang đến điểm gần nhau hơn (theo một số yếu tố nhỏ hơn 1).

- *Định lý các bản đồ co lại cố định*

Một sự biến đổi là co lại khi được áp dụng lặp đi lặp lại với bất kỳ điểm ban đầu, chúng tôi hội tụ về một điểm cố định duy nhất.

Nếu *X* là một không gian ma trận và *W: X ->X* là co lại, lúc đó W có điểm duy nhất cố định |W|

Với các khái niệm trên cho chúng ta thấy một hình ảnh có thể được biểu diễn bằng bộ sưu tập các biến đổi W

- *Các cách phân hoạch ảnh :*

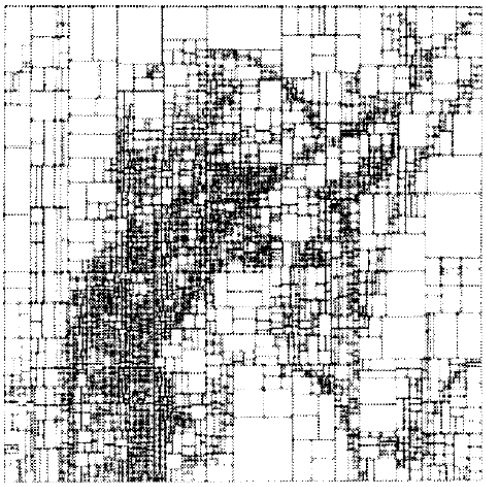
*Phân hoạch quadtree*: trong một phân hoạch quadtree, một hình vuông trong ảnh được chia thành bốn hình vuông con có kích thước bằng nhau khi nó không đủ phủ

tốt bởi domain nào đó. Quá trình lặp lại đệ quy bắt đầu từ toàn bộ ảnh và tiếp tục cho đến khi các hình vuông đủ nhỏ để được phủ trong phạm vi chỉ định cho phép RMS nào đó.

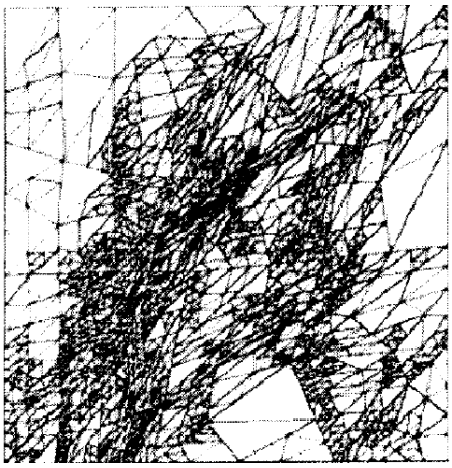
*Phân hoạch HV:* trong một phân hoạch HV một ảnh hình chữ nhật là phân hoạch đệ quy hoặc theo chiều ngang hoặc theo chiều dọc để tạo các hình chữ nhật mới. Việc phân hoạch lặp lại đệ quy cho tới khi việc phủ được chấp nhận như trong sơ đồ phân hoạch quadtree.

*Phân hoạch tam giác:* trong một phân hoạch HV một ảnh hình chữ nhật là phân hoạch đệ quy hoặc theo chiều ngang hoặc theo chiều dọc để tạo các hình chữ nhật mới. Việc phân hoạch lặp lại đệ quy.

*Hình phân hoạch Quadtree.*



*Hình phân hoạch Quadtree.*



*Hình phân hoạch tam giác*

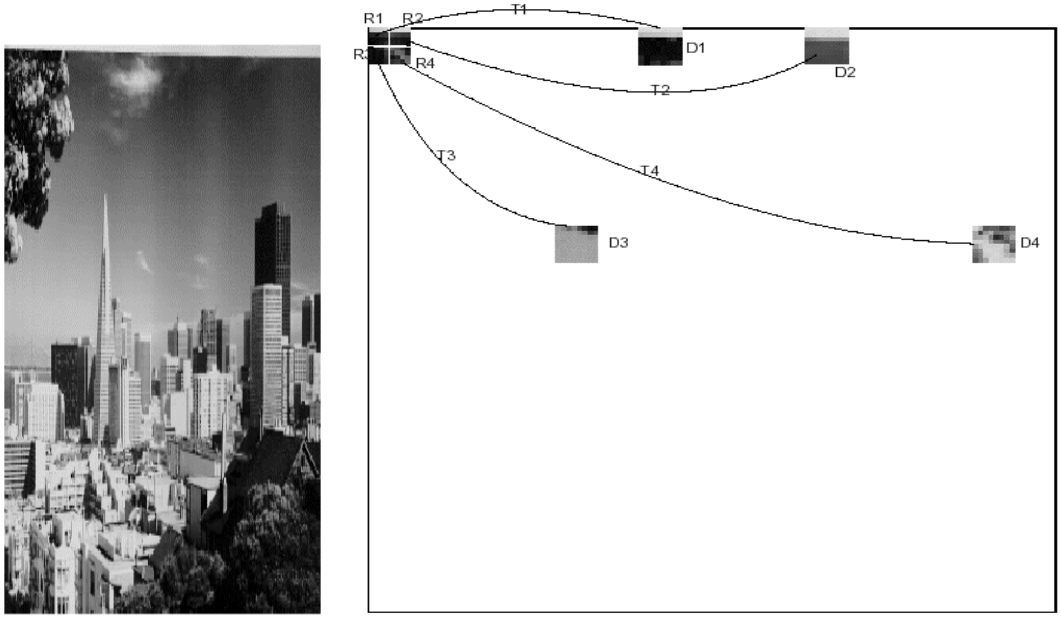
Hình 1. 8 Các loại phân hoạch

**1.3.4.1Nén Fractal**

Hình ảnh cần nén được phân hoạch thành các khối không lấp nhau gọi là khối range. Tập hợp các khối range này có thể bao gồm các khối có cùng kích thước và hình dạng bất kỳ, nhưng các range này phải phủ toàn bộ ảnh. Sau đó xác định một tập các khối domain. Các khối domain này được chọn với kích thước lớn hơn kích thước các khối range để bảo đảm ánh xạ đi từ khối này sang khối range tương ứng là một ánh xạ co lại. ngoài ra các khối domain này có thể phủ lên nhau và không nhất thiết phải phủ kín toàn bộ ảnh cần nén.

Với mỗi khối range cần phải chọn được khối domain thích hợp sao cho khi áp dụng một phép ánh xạ affine co 3 chiều (về vị trí và độ xám), ảnh thu được có thể đối xứng tốt với khối range đang xét. Sau đó các hệ số của phép ánh xạ này được lưu lại. Quá trình này tiếp tục cho đến khi tất cả các khối range của ảnh nén đều đã được duyệt qua.

Sau cùng nó sẽ tạo file ảnh dạng Fractal với các thông tin tương ứng với cá khổi ragne theo sau là danh sách các hệ số afine được chọn đáp ứng khối range.




Hình 1. 9 *Minh họa các khôi Domain(D) , khối Range(R) và phép biến (T)*

### 1.3.4.2 Giải nén Fractal

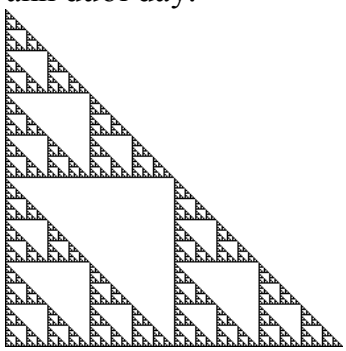
Xuất phát từ dữ liệu nén chứa trong file ảnh Fractal, chúng ta xây dựng lại hệ hàm lặp tương ứng với ảnh khởi động bất kỳ và sau đó áp dụng liên tiếp hệ hàm lặp này vào các ảnh thu được sau mỗi lần lặp cho đến khi thu được ảnh xấp xỉ tốt nhất với ảnh ban đầu.

So với các phương pháp nén thông tin hiện tại (JPEG,MPEG, LZW…) thì nén Fractal cho tỉ số nén vượt trội có thể đến 100:1 bên cạnh đó hình ảnh sau khi giải nén và hình ảnh trước khi nén không sai khác gì nhiều. Vì vậy tôi đã sữ dụng nén Fractal để giải quyết bài toán DH.

# CHƯƠNG II. THUẬT TOÁN ĐỀ XUẤT

## 2.1 Hướng tiếp cận

Với một hình ảnh như  hình ảnh dưới đây:



Hình 2. 1 ảnh cần nén

Chúng ta phải lưu trữ hình xám kích thước $256x256$ này như là một tập hợp các điểm ảnh, và cần ít nhất là 524288 bit cho độ phân giải hiển thị hình ảnh. Nhưng đối với nén Fractal thì hình ảnh phức tạp này được tạo ra từ chỉ có 4 biến đổi affine Một biến đối  được định nghĩa từ 6 con số a,b,c,d,e và f trong công thứ biến đổi của affine, mà và nó chỉ cần768 bit (4 x 6 biến x 32 bit)  để lưu trữ hình ảnh này.

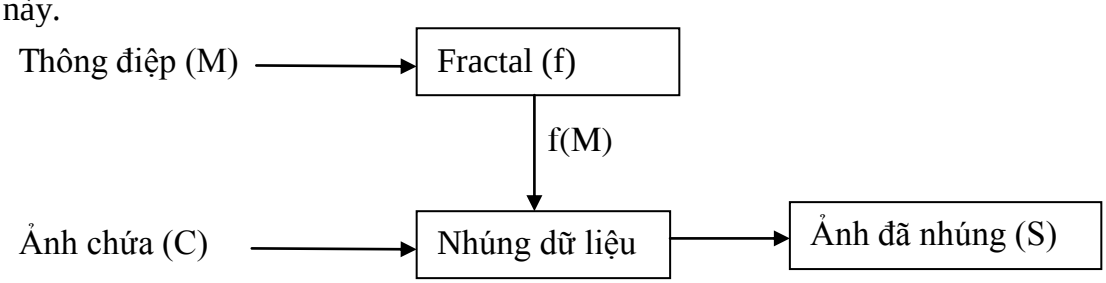
Như vậy thay vì nhúng toàn bộ bit của thông điệp, thì chúng ta sẽ dùng thuật toán nén Fractal để giảm đi dung lượng của ảnh cần được nhúng.

Kỹ thuật đề xuất có ưu điểm chính đó là chỉ tác động đến thông điệp trước khi nhúng và sau khi nhúng, do đó khi áp dụng với các thuật toán ẩn dữ liệu hiện có sẽ làm tăng khả năng chứa cũng như khả năng vô hình và khả năng lưu trữ; mặt khác thông điệp trước khi nhúng đã được nén với tỉ số nén cực cao và sau khi nhúng phục hồi lại chất lượng ảnh gần như không thay đổi.

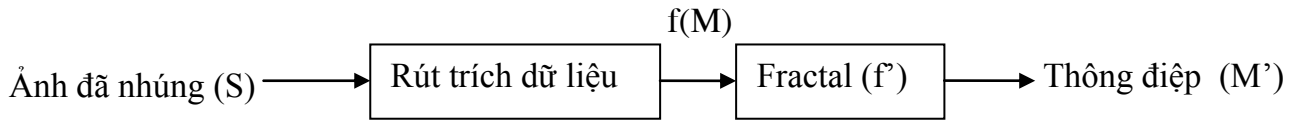
## 2.2 Thuật toán đề xuất:

Thuật toán áp dụng cho thông điệp mật dạng ảnh xám. Qui trình nhúng như hình 2.2. Thuật toán xem thông điệp cần nhúng (M) sau khi được đưa vào hàm nén Fractal ta được hình ảnh nén Fractal (M’). Trước khi nhúng thông điệp (M’) được xem như là một chuỗi bít và các bit này được ẩn vào trong ảnh chứa bằng thuật toán nhúng. Để tiện

cho việc trình bày và so sánh, thuật toán nhúng LSB-MR sẽ được áp dụng trong phần này.



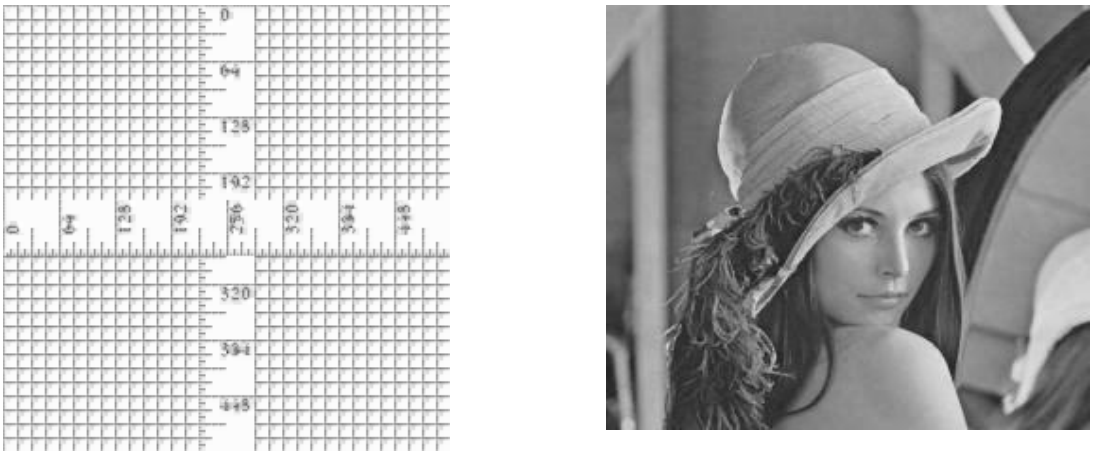

(a)




(b)

Hình 2. 2 (a) Qui trình nhúng (b) Qui trình rút trích.

Trong quá trình rút trích, thuật toán sẽ dùng ảnh sau khi nhúng như đầu vào, đâu tiên ảnh nhúng sẽ được đưa qua phần rút trích của thuật toán LSB-MR, chúng ta sẽ được file hình ảnh Fractal f(M), tiếp theo f(M) này sẽ được đưa giải nén Fractal, ta sẽ được thông điệp mật hình ảnh ban đầu, tuy nhiên ảnh sẽ bị nhiễu nhưng trong giới hạn chấp nhận được bằng mắt thường.



Hình 2. 3nhiễu của ảnh mật

**2.3 Qui trình nhúng**

Thuật toán nhúng một thông điệp mật M dạng ảnh xám có kích thước MxN (M,N là bội số của 4 vì nén) vào ảnh chứa C ảnh xám có kích thước KxZ với MxN <= KxZ

*Input:* ảnh nhị phân M và ảnh chứa C

*Output*: ảnh đã nhúng S.

Qui trình nhúng được thực hiện theo các bước sau:

1. Đưa thông điệp qua thuật toán nén Fractal được M'là file ảnh IFS
2. Nhúng *M'* vào *C* bằng thuật toán LSB-MR [4]. Đơn vị nhúng của LSB-MR là 1 cặp pixel. LSB-MR không áp dụng trên những điểm ảnh bão hòa (những điểm ảnh có giá trị màu cực đại hoặc cực tiểu). Với mỗi cặp $x_i, x_{i+1}$ thuật toán sẽ thực hiện nhúng theo 1 trong 4 trường hợp sau:

$$TH1: if\ LSB\ x_i = m_i \&\& f\ x_i, x_{i+1} = m_{i+1}$$
$$then\ x_i', x_{i+1}' = x_i, x_{i+1}$$
$$TH2: if\ LSB\ x_i = m_i \&\& f\ x_i, x_{i+1} \neq m_{i+1}$$
$$then\ x_i', x_{i+1}' = (x_i, x_{i+1} + r)$$
$$TH3: if\ LSB\ x_i \neq m_i \&\& f\ x_i, x_{i+1} = m_{i+1}$$

$$then \;\; x_i', x_{i+1}' \;=\; x_i - 1, x_{i+1}$$

$$TH4: if \; LSB \;\; x_i \;\neq m_i \&\& \; f \;\; x_i, x_{i+1} \;\neq m_{i+1}$$

$$then \;\; x_i', x_{i+1}' \;= (x_i, x_{i+1})$$

Với, $m_i$ và $m_{i+1}$ biểu diễn 2 bít cần được nhúng của M'. Hàm f được định nghĩa bởi (8); $r$ là giá trị random của -1 hoặc 1 và $(x_i', x_{i+1}')$ biểu diễn cặp ảnh sau khi nhúng *S*.

**2.4 Qui trình rút trích**

*Input:* Ảnh đã nhúng *S*

*Output:* Thông điệp được rút trích *M*

Qui trình rút trích được thực hiện theo các bước sau:

1. Rút trích phần nhúng $M'$ từ ảnh đã nhúng *S* bằng thuật toán LSB-MR với mỗi đơn vị nhúng $(x_i', x_{i+1}')$, 2 bit $(m_i, m_{i+1})$ được rút trích như sau:

$$m_i = LSB \;\; x_i' \;\; , m_{i+1} = LSB \quad \frac{x_i'}{2} \; + x_{i+1}' \quad 9$$

2. ảnh *M* được rút trích từ $M'$ dựa vào giải nén Fractal.

# CHƯƠNG III. KẾT QUẢ THỰC NGHIỆM VÀ PHÂN TÍCH.

Trong phần thực nghiệm, tập hình ảnhđược sử dụng làm ảnh chứa để đánh giá thuật toán so với LSB matching và LSB-MR. tập ảnh [15] không nén với kích thước 1600x1143 và 1143x1600. Những ảnh này được chuyển sang dạng ảnh xám và kích thước 1024x1024 để tiến hành các thuật toán ẩn dữ liệu như hình 3.1. Một tập ảnh xám có kích thước 256x256 được dùng làm thông tin mật hình 3.2.

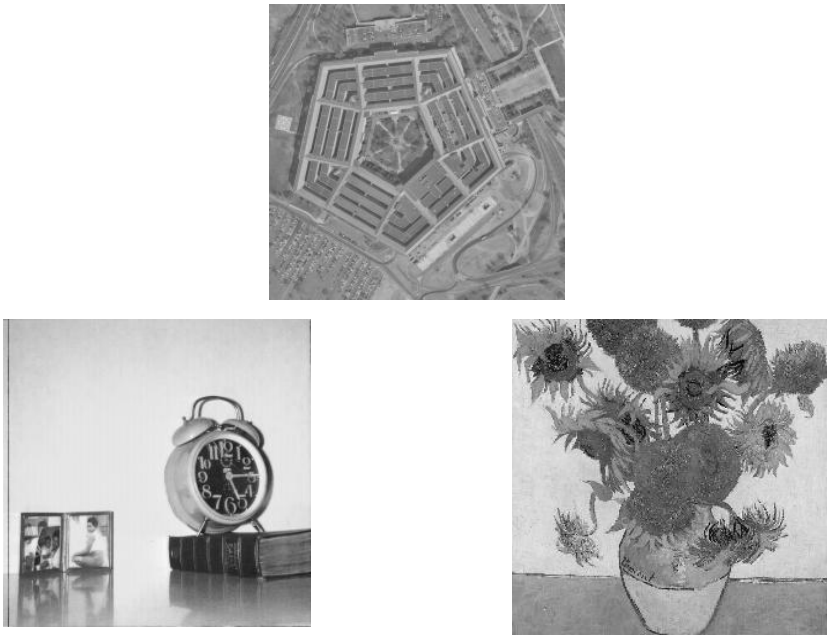
Ảnh màu 1600x1143 Ảnh xám 1024x1024

Ảnh màu 1600x1143 Ảnh xám 1024x1024

Hình 3. 1 Ảnh chưa với kích thước 1024x1024

### 3.1Đánh giá về dung lượng và tính vô hình

Hướng tiếp cận của thuật toán đề xuất thì hình ảnh thông tin mật được nhúng sau khi nén Fractal với tỉ lệ nén 2,27:1. Do đó, dung lượng nhúng của thuật toán đề xuất thì gấp 2,27 lần so với LSB-MR ở cùng tỷ lệ thay đổi trên mỗi pixel. Vì dùng thuật toán nén mất thông tin nên thông điệp được rút trích bị thiếu một số bit.



Hình 3. 2Một số ảnh ẩn 256x256

Tỉ lệ nhiễu (Peak Signal to Noise Ratio - PSNR) [2] xác định chất lượng sau khi nhúng hay nói cách khác là đánh giá độ khác biệt của ảnh sau khi nhúng và ảnh ban đầu. Đây cũng chính là thông số để đánh giá khả năng vô hình của ảnh sau khi nhúng. PSNR đo độ khác biệt giữa 2 ảnh bằng cách tính bình phương trung bình lỗi (MSE). Gia trị PSNR càng cao thì tính vô hình của ảnh sau khi nhúng càng cao, hay sự khác biệt giữa ảnh nhúng và sau khi nhúng càng ít.

Nếu C là ảnh chứa và S là ảnh sau khi nhúng có kích thước MxN thì PSNR được tính như sau

$$MSE = \frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [C(i,j) - S(i,j)]^2 \quad (14)$$

$$PSNR = 10 * \log_{10} \frac{255^2}{MSE} \quad dB \quad (15)$$

Bảng 3.1 thể hiện kết quả thực nghiệm của thuật toán đề xuất trên bộ dữ liệu so với LSB-MR và LSB matching. Bảng thể hiện tính vô hình của thuật toán đề xuất có chất lượng tốt hơn so với các thuật toán LSB, LSB matching, LSB-MR.

Bảng 3. 1 Giá trị trung bình PSNR trên 100 ảnh đã nhúng với các thuật toán ẩn dữ liệu khác nhau.

| Thuật toán | PSNR |
|---|---|
| **LSB matching** | 54.081 |
| **LSB-MR** | 55.329 |
| **Thuật toán đề xuất** | 58.987 |

### 3.2 Đánh giá về tính mạnh mẽ

Tỉ lệ bít lỗi (Bit Error Rate - BER) [2] xác định tỉ lệ bit lỗi trong quá trình rút trích thông tin M. Thông số này được áp dụng để đánh giá mức độ mạnh mẽ của hệ thống trước tấn công cũng như xác định mức độ lỗi do các tấn công gây ra.

Nếu M là tín hiệu ảnh sau khi rút trích có kích thước *MxN*

$$BER = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} N_{i,j}}{MxN} \quad (16)$$

Trong đó, $N_{i,j}$ là số lượng bit khác nhau giữa các điểm pixel tương ứng với tín hiệu mật trước khi nhúng và sau khi rút trích.

a)

b)

c)

Hình 3. 3 a)Ảnh mật ban đầu trước khi nhúng; b) Ảnh nhúng và rút trích bằng thuật toán LSB MR; c) Ảnh mật nhúng và rút trích bằng thuật toán đề xuất.

Nén Fractal là nén mất thông tin, nên ảnh sau khi được tách và giải nén khỏi ảnh nhúng thì ta có được ảnh mật sau cùng sẽ có tỉ lệ bit lỗi sẽ cao hơn với các thuật toán khác, Thực nghiệm trên 100 ảnh sau khi nhúng thì kết quả cho thấy thuật toán đề xuất có tỉ lệ bít lỗi là 27%. Tuy nhiên, như quan sát ở hình 3.3 ta thấy hình ảnh rút trích bằng thuật toán đề xuất không được rõ nét như hình ảnh ban đầu nhưng về hình ảnh thì vẫn như ban đầu.

### 3.3 Đánh khả năng chống tấn công

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

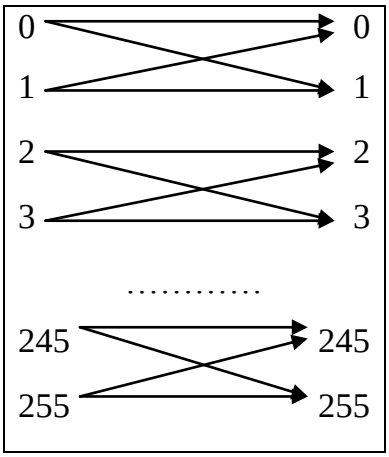
Phân tích tin ẩn giấu thường dựa vào các yếu tố sau:

- 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 điệp 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.

*a) Kỹ thuật phát hiện dữ liệu ẩn Pair of values (POV)*

Trong tài liệu [13] của hai tác giả Westfeld và Pfitzmann nói rằng, LSB trong ảnh không phải sinh ngẫu nhiên. Mà là, ý nghĩa về tần số xuất hiện của cặp hai giá trị trong POV. POV được hiểu như sau:

Nếu giá trị điểm ảnh gốc là 100, sau khi nhúng tin, nó sẽ có giá trị 101 hoặc 100. Nếu điểm ảnh gốc có giá trị là 101, sau khi nhúng tin, nó sẽ có giá trị 100 hoặc 101. Như vậy {100,101} là một cặp POV, chúng ta có một ánh xạ từ C sang S: {0, 1} $\rightarrow${0, 1}, {2, 3} $\rightarrow${2, 3}, {3, 4}$\rightarrow${3, 4}, … , {254, 255}$\rightarrow${254, 255} theo sơ đồ sau:




Hình 3. 4 Sơ đồ ánh xạ giá trị các pixel khi nhúng

Dạng tổng quát của các cặp POV là $\{2k,\ 2k+1 \mid 0 \le k \le 127\}$

Khi thông tin được nhúng vào một ảnh dùng thuật LSB nó sẽ tạo ra các cặp POV, khi đó tần số của *2k* và *2k+1* trở lên bằng hoặc gần bằng nhau. Điều này hiếm khi xẩy ra với ảnh chưa giấu tin.

Thống kê $\chi^2$ dùng để phát hiện số POV gần bằng nhau trong ảnh có nhúng tin, dựa trên việc kiểm tra sự tương quan của giá trị pixel chẵn và giá trị pixel lẻ.

Thống kê $\chi^2$ được thiết kế cho nhiều thuật toán nhúng khác nhau, nhưng nội dung cơ bản là đánh giá giống nhau về các thuật toán nhúng thông điệp trên miền LSB của ảnh. Sau đây mô tả thuật toán POV sử dụng thống kê $\chi^2$ để phát hiện cho ảnh có giấu tin.

Người dùng chọn tùy ý một ảnh, POV kiểm tra tổng thể các pixel ảnh từ 1% đến 100% kích cỡ của ảnh và trả về khả năng tính toán cho mỗi trường hợp kiểm tra ảnh. Các pixel được kiểm tra lần lượt từ trái qua phải, từ trên xuống dưới giống như trình tự nhúng.

***Thuật toán POV***

Input : Ảnh cấp xám đã giấu tin

Output : Phát hiện xem ảnh đó có giấu tin hay không

Bước 1: Đọc dữ liệu vào ma trận $M_{m\ x\ n}$

Bước 2: Khởi tạo giá trị ban đầu cho vecto $X, Y$

$$X\ k\ = 0\ ; Y\ k\ = 0\ với\ \mathrm{k} = \{0, \dots, 127\}$$

Bước 3: Tình $X\ k$ là tần số xuất hiện của các điểm ảnh có giá trị chẵn trên ảnh.

Tình $Y\ k$ là tần số xuất hiện của các điểm ảnh có giá trị lẻ trên ảnh.

Bước 4 : Giả sử ta có N cặp PoV

$$Nếu\ X\ k\ + Y\ k\ \leq 4\ \text{Thì}\ X\ k\ = Y\ k\ = 0$$

Với mọi *k*Và*N=N-1*

Bước 5: Tính $Z\ k\ = (X\ k\ + Y\ k\ )/2$

Bước 6: Giả sử ta có N cặp PoV, theo thống kê $\chi^2$ với n-1 mức tự do được tính như sau

$$X_{n-1}^{2} = \sum_{i=0}^{127} \frac{(x_i - z_i)^2}{z_i}$$

Bước 7: Tinh P là xác suất của việc giấu tin

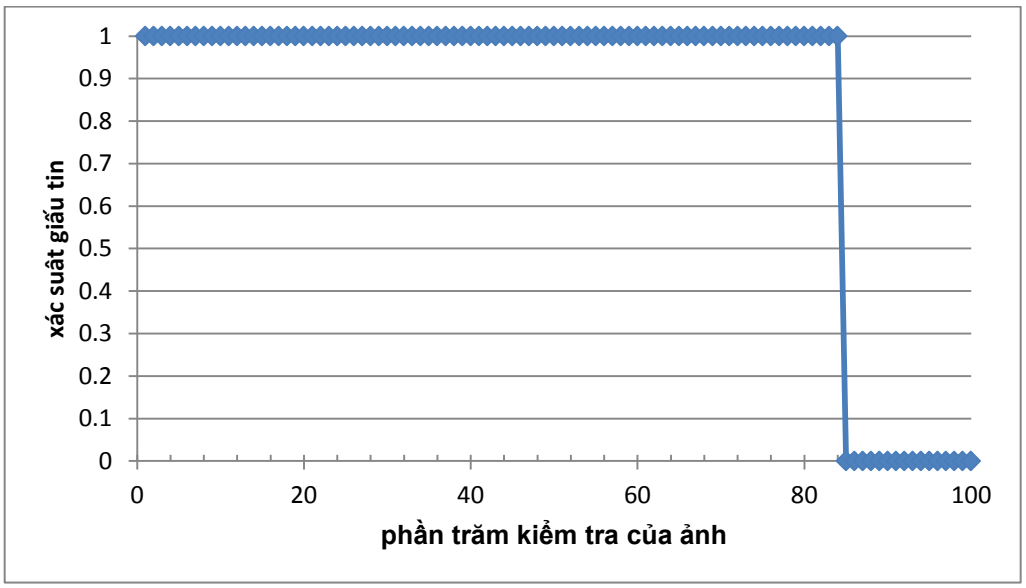
$$\mathrm{p} = 1 - \frac{1}{2^{\frac{n-1}{2}}\Gamma(\frac{n-1}{2})} \int_0^{X_{n-1}^2} e^{-\frac{u}{2}} u^{\frac{n-1}{2}-1} du$$

vậy p tiến đến 0 khi $\chi_{n-1}^2$ tiến đến vô hạn. Do đó đối với $\chi_{n-1}^2$ rất lớn thì xác suất nhúng là bằng 0. Khi $\chi_{n-1}^2$ là nhỏ so với n-1, thì 1-p gần bằng 0, p gần bằng 1. Do vậy với $\chi_{n-1}^2$ nhỏ, xác suất nhúng gần bằng 1. Thêm nữa, Westfeld và Pfitzmann cho rằng nếu dữ liệu giấu tin trong ảnh càng nhiều thì xác suất phát hiện càng chính xác.

Ví dụ: ta sử dụng thuật toán POV đối với ảnh 001.bmp sau đã được ẩn dữ liệu bằng thuật toán LSB.


Hình 3. 5 hình 001.bmp đã ẩn dữa liệu.

Ta sẽ được hình sau:




Hình 3. 6 biểu đồ mô tả phát hiện ảnh có giấu tin sử dụng thống kê POV

Đối với một ảnh thông tin giấu được nhúng liên tục (từ góc trên trái của ảnh) thì giá trị của P sẽ gần tới 1 và sau đó rơi xuống 0 khi chúng ta xét tới các vùng không giấu tin. Với kỹ thuật này không những độ phát hiện rất cao mà còn tính toán được độ dài của thông điệp giấu.

Với bảng 3.2 là kết quả ứng dụng thuật toán POV (với hệ số kiểm tra là 10% của ảnh) trên tập ảnh ẩn dữ liệu 100 kết quả ẩn của mỗi loạithuật toán LSB , thuật toán LSB-MR và thuật toán đề xuất thì số lượng phát hiện là 34, 29 và 23. Từ đây ta cũng

thấy việc xác định thông tin ẩn của cả 3 thuật toán thì thuật toán đề xuất có khả năng vô hình cao hơn.

Bảng 3. 2 Bảng kết quả xác suất giấu tin với thuật toán POV

| Tên ảnh | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Thuật toán đề xuất | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| LSB - MR | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0.87 | 1 | 1 |
| LSB | 0 | 0.78 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 1 | 0.81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 1 | 0.58 | 1 | 1 | 1 |
| | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0.23 | 0 |
| | 1 | 1 | 1 | 0.93 | 0 | 0 | 0.95 | 1 | 0.98 | 0 |
| | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 0.34 | 0.82 | 0.12 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| | 1 | 0.62 | 0.51 | 1 | 0 | 0.87 | 1 | 0 | 0 | 0 |
| | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0.97 | 0.88 | 0 | 0 | 0 | 0 | 0 | 0.52 |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 0 | 0 | 0.92 | 0.89 | 1 | 0 | 0 |
| | 0.86 | 0 | 0 | 0 | 1 | 0 | 0 | 0.99 | 1 | 1 |
| | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| | 0.46 | 0 | 1 | 0.64 | 0.71 | 0 | 0.95 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0.76 | 0 | 1 | 0 | 1 | 0 | 1 |
| | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0.99 | 0 | 0 | 0.56 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0.67 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

### *a. Kỹ thuật phát hiện dữ liệu ẩn RS*

Thuật toánRegular Singular - RS là thuật toán phát hiên tin cậy do nhóm tác giả J. Fridrich, M. Goljan, and R. Du [14] đưa ra, với ý tưởng chia miền giá trị của ảnh thành các nhóm có miền giá trị đều đặn (R-Regular), miền giá trị dị thường (S-Singular). Ta thấy rằng với một ảnh có giấu tin thì tổng số của các miền R rất gần với tổng số của miền S.

Để thực hiện việc phân dữ liệu ảnh thành miền có giá trị đều đặn và miền giá trị dị thường ta sử dụng các hàm phụ trợ:

+ Hàm Hamming xác định khoảng cách giữa các điểm trong một tập

$$f\ x_1, x_2, \dots, x_n\ = \sum_{i=1}^{x-1} |x_{i+1} - x_i|$$

+ Một ánh xạ F xác định trên tập P gọi là Flipping nếu $F^2 = I$ trong đó I là một ánh xạ đồng nhất. Ta sử dụng hai tập ánh xạ F1 và F-1 như sau:

Ta có công thức F-1(x)=F1(x+1) –1 với tất cả các x (3)

=> Nhóm G được quyết định là thuộc nhóm nào trong 3 nhóm (Regular Group (R), Singular Group (S), Unusable Group (U)) khi và chỉ khi:

$$G \in R \Leftrightarrow f(F_M(G)) > f(G)$$

$$G \in S \Leftrightarrow f(F_M(G)) < f(G)$$

$$G \in U \Leftrightarrow f(F_M(G)) = f(G)$$

+ Với M là một mặt lạ phụ trợ chứa các giá trị -1, 0, 1 để trộn các Pixel trong nhóm. $F_M(G)$ được xác định như sau:

$$F_M(G) = \{ F_{M_1}\ x_1\ , F_{M_2}\ x_2\ , \dots, F_{M_n}\ x_n\ \} \quad với \quad x_i \in G$$

***Thuật toán RS***

Input : Ảnh cấp xám đã giấu tin

Output : Phát hiện xem ảnh đó có giấu tin hay không

Bước 1: Đọc dữ liệu ảnh C với các giá trị pixel của nó thuộc$\mathrm{P} = \{0, \dots, 255\}$.

Bước 2: Sau đó chia C thành *k*nhóm G khác nhau mỗi nhóm có n pixel

$$G_k = (\mathrm{x1}, \dots, \mathrm{xn}).$$

Bước 3: Tính khoảng cách giữa các điểm trong tập $G_k$

$$f\ x_1, x_2, \dots, x_n\ = \sum_{i=1}^{x-1} |x_{i+1} - x_i|$$

Bước 4: Trộn các Pixel trong nhóm G với mặt nhạ phụ trợ M chứa các giá trị *{-1, 0, 1}* để trộn các Pixel trong nhóm. Tính $F_M(G)$ *:*

$$F_M(G) = \{\ F_{M_1}\ x_1\ , F_{M_2}\ x_2\ , \dots, F_{M_n}\ x_n\ \} \quad với i \quad x_i \in G$$

Bước 5: Tính khoảng cách giữa các điểm trong tập $F_M(G)$*:*

$$f\ F_{M_1}(x_1), F_{M_2}(x_2), \dots, F_{M_n}(x_n)\ = \sum_{i=1}^{x-1} |F_{M_i}(x_{i+1}) - F_{M_i}(x_i)|$$

Bước 6: Nhóm G được quyết định là thuộc nhóm nào trong 3 nhóm (Regular Group (R), Singular Group (S), Unusable Group (U)) khi và chỉ khi:

$$\mathrm{G} \in \mathrm{R} \Leftrightarrow \mathrm{f}(\mathrm{F_M}(\mathrm{G})) > f(G)$$

$$\mathrm{G} \in \mathrm{S} \Leftrightarrow \mathrm{f}(\mathrm{F_M}(\mathrm{G})) < \mathrm{f}(\mathrm{G})$$

$$\mathrm{G} \in \mathrm{U} \Leftrightarrow \mathrm{f}(\mathrm{F_M}(\mathrm{G})) = \mathrm{f}(\mathrm{G})$$

Bước 7: Lập lại bước 3 cho tới khi hết các nhóm $G_k$.

Bước 8: Đặt $\mathrm{R_M}$ là số nhóm của R, $\mathrm{S_M}$ là số nhóm của S, $\mathrm{U_M}$ là số nhóm của U.

Chúng ta có:

Theo giả thuyết thống kê của phương pháp này đó là trong một ảnh điển hình (chưa giấu thông tin) thì giá trị của $\mathrm{R_M}$ gần bằng giá trị của $\mathrm{R_{\_M}}$ và tương tự giá trị của $\mathrm{S_M}$ gần bằng giá trị của $\mathrm{S_{\_M}}$. (Với _M = _M)

$$R_M \cong R_{\_M} \text{ và } S_M \cong S_{\_M}$$

Sự ngẫu nhiên của LSB gây ra sự khác nhau giữa $\mathrm{R_M}$ và $\mathrm{S_M}$. Khi độ dài của thông điệp giấu trong LSB càng tăng lên thì nó làm cho $\mathrm{R_M}$và $\mathrm{R_{\_M}}$ càng rất khác nhau,

tương tự $S_M$càng khác $S_{-M}$trong khi đó $R_M$ và $S_M$ có giá trị rất gần nhau. Tức là với ảnh có giấu thông tin ẩn trong LSB của ảnh thì

$$R_M \cong S_M$$

$$R_M \neq R_{\_M} \ (R_{\_M} > R_M)$$

$$S_M \neq S_{\_M} \ (S_M > S_{\_M})$$

Ứng dụng các thuộc tính của RS ta thực nghiệm trên những tập ảnh ẩn dữ liệu 100 ảnh bằng thuật toán LSB-MR thì số lượng phát hiện là 7 và thuật toán đề xuất ta được là 5. Từ đó ta thấy thuật toán RS phát hiện ảnh chứa thông tin là thấp đối với 2 thuật toán LSB-MR và thuật toán đề xuất.

***c) Kỹ thuật phát hiện dữ liệu ẩn HCF COM***

Kỹ thuật phát hiện dữ liệu ẩn HCF COM (**H**istogram **C**haracteristic **F**unction – the **C**enter **o**f **m**ass) [12] là phương pháp nổi tiếng trong việc phát hiện dữ liệu ẩn được nhúng bằng thuật toán thuộc họ LSB. HCF COM sử dụng hàm đặc trưng histogram để phát hiện ẩn dữ liệu ẩn:

$$h_c\ n\ = \ \ i,j\ |\ p_c\ i,j\ = n \qquad (17)$$

Với $h_c\ n$ là số lượng pixel có giá trị màu n *(n= 0,…,255)* , $p_c\ i,j$ pixel tại vị trí $i,j$

Harmsen sử dụng khối trung tâm COM của HFC

$$C\ H\ k\ \ = \frac{\sum_{i=0}^{n} i\ H\ i}{\sum_{i=0}^{n} H\ i} \qquad (18)$$

Với $H\ i$ là biến đổi Fourier (DFT) của $h_c\ n$ với chiều dài $k = 0, \dots, , \frac{N}{2}$.

Mặt khác HCF- COM nó xem ẩn dữ liệu dạng LSB như là việc thêm nhiễu tăng cương

$$p'_c\ i,j\ = \sum_{u=0}^{1}\sum_{v=0}^{1} \frac{p_c\,(2i+u, 2j+v)}{4} \qquad (19)$$

Tiếp theo tính HCF và COM của $p'_c$ được $C\ H'\ k$ và nó được so sánh với $C\ H\ k$ nếu hình ảnh không có giữ liệu ẩn thì

$$C\ H'\ k \approx C\ H\ k \quad 20$$

Ứng dụng các thuộc tính của HCF COM ta thực nghiệm trên những tập ảnh ẩn dữ liệu bằng thuật toán LSB matching, LSB-MR , và thuật toán đề xuất ta được kết quả như sau:

Bảng 3. 3 Bảng kết quả ứng dụng thuật toán HCF COM trên tập ảnh ẩn dữ liệu

| Ảnh | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| LSB - MR | $C\ H\ k$ | 36.644 | 59.097 | 49.566 | 31.653 | 31.829 | 33.576 | 31.045 | 37.292 |
| | $C\ H'\ k$ | 37.776 | 59.595 | 42.942 | 20.362 | 24.902 | 38.09 | 28.942 | 17.081 |
| LSB - Matching | $C\ H\ k$ | 36.364 | 59.098 | 49.48 | 32.22 | 32.208 | 37.55 | 31.733 | 37.43 |
| | $C\ H'\ k$ | 37.673 | 59.598 | 42.923 | 21.454 | 25.439 | 37.698 | 29.169 | 17.268 |
| Đề xuất | $C\ H\ k$ | 37.36 | 59.1 | 49.303 | 32.72 | 32.832 | 33.585 | 32.259 | 37.852 |
| | $C\ H'\ k$ | 38.138 | 59.592 | 42.755 | 20.769 | 25.117 | 32.059 | 29.364 | 17.971 |
| | | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | | 46.144 | 42.891 | 41.908 | 38.274 | 36.115 | 37.007 | 43.758 | 34.991 |
| | | 41.464 | 43.145 | 32.304 | 30.713 | 33.655 | 18.834 | 40.757 | 12.915 |
| | | 46.161 | 42.954 | 42.528 | 38.939 | 37.251 | 37.463 | 43.733 | 34.284 |
| | | 41.491 | 43.169 | 32.809 | 30.812 | 33.717 | 19.732 | 40.82 | 14.487 |
| | | 46.387 | 43.067 | 42.064 | 39.121 | 36.92 | 37.663 | 44.092 | 35.126 |
| | | 41.667 | 43.174 | 32.392 | 30.951 | 33.854 | 20.05 | 40.925 | 13.761 |
| | | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| | | 36.681 | 44.191 | 34.487 | 48.109 | 39.04 | 34.619 | 33.907 | 33.557 |
| | | 25.731 | 46.092 | 33.219 | 47.561 | 35.92 | 22.301 | 32.385 | 30.007 |
| | | 36.981 | 44.508 | 34.832 | 48.103 | 38.963 | 34.962 | 34.071 | 33.843 |
| | | 26.325 | 45.785 | 33.248 | 47.53 | 35.193 | 22.715 | 32.314 | 30.289 |
| | | 37.076 | 44.714 | 35.275 | 48.259 | 39.796 | 34.243 | 34.421 | 34.295 |
| | | 26.465 | 45.91 | 33.564 | 47.676 | 36.383 | 21.726 | 32.523 | 30.558 |
| | | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| | | 34.229 | 48.052 | 34.124 | 41.087 | 34.424 | 35.99 | 51.672 | 43.376 |
| | | 26.105 | 50.044 | 35.089 | 38.73 | 39.222 | 37.383 | 51.977 | 45.756 |
| | | 37.28 | 47.98 | 34.047 | 42.998 | 34.276 | 38.086 | 51.669 | 43.291 |
| | | 28.431 | 49.963 | 34.841 | 39.936 | 39.142 | 38.189 | 51.959 | 45.768 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 34.637 | 48.314 | 34.257 | 40.859 | 34.447 | 34.982 | 51.825 | 43.826 |
| 26.419 | 50.023 | 34.921 | 38.686 | 39.191 | 36.943 | 52.018 | 45.85 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 37.998 | 46.028 | 36.802 | 34.187 | 43.311 | 60.884 | 41.693 | 53.558 |
| 40.134 | 46.515 | 29.237 | 33.317 | 46.522 | 61.649 | 32.901 | 55.721 |
| 38.57 | 46.87 | 38.494 | 34.115 | 43.651 | 50.656 | 43.193 | 42.673 |
| 40.202 | 46.546 | 32.609 | 33.178 | 47.074 | 51.056 | 35.946 | 42.924 |
| 38.589 | 45.779 | 37.114 | 34.646 | 43.496 | 48.341 | 42.624 | 43.39 |
| 40.196 | 46.568 | 26.137 | 33.374 | 47.008 | 49.187 | 32.884 | 43.042 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 53.572 | 37.095 | 34.715 | 36.094 | 32.156 | 33.551 | 35.45 | 33.848 |
| 54.34 | 40.364 | 36.45 | 22.26 | 29.802 | 21.164 | 38.764 | 27.021 |
| 53.996 | 38.174 | 34.469 | 36.271 | 34.292 | 36.561 | 37.181 | 36.391 |
| 54.413 | 40.095 | 36.183 | 22.313 | 30.341 | 23.383 | 39.111 | 27.476 |
| 53.859 | 36.435 | 34.826 | 36.346 | 33.015 | 34.009 | 36.289 | 33.708 |
| 54.61 | 39.053 | 36.327 | 22.675 | 30.002 | 21.698 | 38.963 | 26.882 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 37.293 | 37.02 | 33.579 | 40.126 | 34.909 | 45.262 | 35.732 | 37.994 |
| 33.974 | 38.535 | 32.696 | 38.96 | 27.965 | 36.8 | 25.124 | 32.37 |
| 37.244 | 37.983 | 33.786 | 38.439 | 33.944 | 45.291 | 34.309 | 39.566 |
| 34.317 | 38.338 | 32.58 | 38.327 | 27.757 | 36.843 | 22.975 | 39.277 |
| 37.882 | 37.771 | 34.89 | 39.348 | 34.525 | 45.45 | 35.02 | 38.878 |
| 33.701 | 38.656 | 33.078 | 38.625 | 27.617 | 37.172 | 23.76 | 32.811 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |
| 36.815 | 54.981 | 43.93 | 40.935 | 40.138 | 35.994 | 36.548 | 43.288 |
| 31.196 | 44.524 | 44.522 | 29.615 | 40.868 | 40.225 | 37.799 | 45.399 |
| 37.127 | 54.761 | 43.986 | 39.718 | 41.136 | 36.221 | 37.32 | 43.037 |
| 31.158 | 44.169 | 44.611 | 29.364 | 41.13 | 40.211 | 38.023 | 45.383 |
| 37.455 | 54.584 | 44.2 | 39.834 | 40.261 | 36.39 | 37.119 | 43.353 |
| 31.309 | 44.235 | 44.515 | 29.553 | 40.728 | 40.223 | 37.851 | 45.544 |
| 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 |
| 43.555 | 41.535 | 45.16 | 42.66 | 32.534 | 34.17 | 46.351 | 46.762 |
| 45.072 | 42.636 | 46.325 | 44.519 | 22.187 | 40.587 | 47.952 | 48.697 |
| 43.445 | 37.753 | 44.887 | 42.837 | 32.671 | 34.814 | 45.709 | 47.024 |
| 44.868 | 40.863 | 46.265 | 44.364 | 22.508 | 40.779 | 47.744 | 48.707 |
| 44.128 | 37.104 | 45.23 | 42.892 | 33.267 | 35.075 | 45.921 | 46.732 |
| 45.192 | 40.339 | 46.432 | 44.445 | 22.546 | 40.857 | 47.709 | 48.62 |
| 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 40.878 | 35.317 | 31.866 | 36.025 | 36.492 | 37.33 | 33.008 | 39.836 |

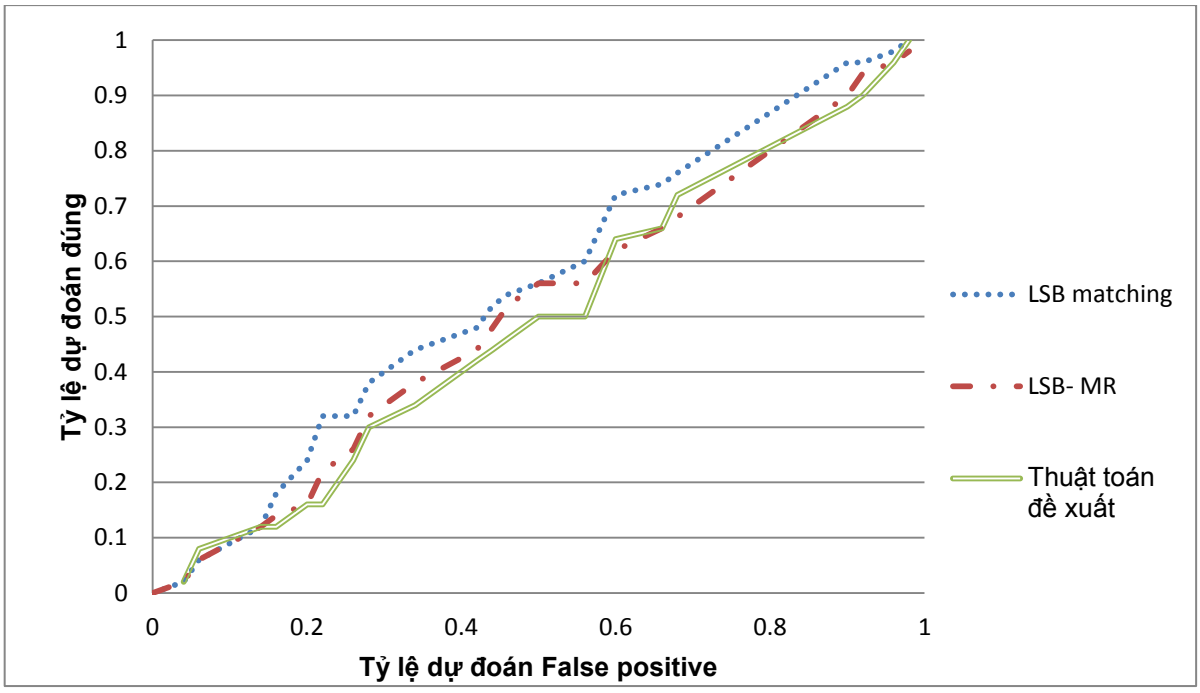
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 44.021 | 30.142 | 21.147 | 25.355 | 38.635 | 41.183 | 32.628 | 27.741 |
| 39.57 | 35.653 | 36.111 | 35.877 | 36.714 | 40.02 | 32.56 | 42.054 |
| 43.396 | 29.532 | 24.099 | 25.224 | 38.902 | 41.974 | 32.718 | 26.287 |
| 40.165 | 36.306 | 33.15 | 36.299 | 36.947 | 38.125 | 32.601 | 41.531 |
| 43.589 | 29.994 | 21.517 | 25.81 | 39.139 | 41.422 | 32.786 | 27.412 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 |
| 55.95 | 32.154 | 46.198 | 35.304 | 54.436 | 54.678 | 48.06 | 54.759 |
| 56.926 | 29.745 | 48.71 | 35.375 | 55.082 | 55.224 | 46.873 | 56.438 |
| 55.81 | 32.498 | 46.042 | 36.775 | 55.918 | 54.271 | 48.989 | 54.781 |
| 56.597 | 29.563 | 46.602 | 35.315 | 55.701 | 55.534 | 50.004 | 56.334 |
| 56.011 | 33.229 | 46.602 | 35.601 | 54.76 | 54.779 | 49.22 | 54.818 |
| 56.706 | 29.792 | 47.099 | 35.313 | 54.847 | 56.124 | 50.619 | 56.415 |
| 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| 38.712 | 40.901 | 28.839 | 35.364 | 38.298 | 33.414 | 50.503 | 38.957 |
| 40.794 | 44.639 | 37.48 | 32.713 | 38.351 | 31.885 | 50.83 | 40.942 |
| 40.258 | 40.798 | 39.013 | 38.111 | 39.241 | 33.776 | 51.836 | 41.996 |
| 41.517 | 41.768 | 36.125 | 33.97 | 39.002 | 31.885 | 51.547 | 41.779 |
| 39.587 | 41.093 | 30.741 | 36.467 | 38.965 | 34.107 | 51.348 | 40.276 |
| 41.135 | 41.649 | 34.23 | 33.119 | 38.694 | 31.927 | 51.133 | 41.322 |
| 97 | 98 | 99 | 100 | | | | |
| 33.364 | 34.695 | 41.422 | 40.483 | | | | |
| 22.001 | 31.167 | 41.254 | 35.839 | | | | |
| 34.456 | 38.406 | 41.894 | 40.532 | | | | |
| 22.93 | 32.047 | 41.722 | 35.594 | | | | |
| 34.238 | 35.685 | 41.773 | 40.652 | | | | |
| 22.644 | 31.114 | 41.467 | 36.026 | | | | |

Dựa vào bảng 3.3 ta sẽ tính được bảng 3.4 theo các tỉ lệ của $C\ H\ k\ /C\ H'\ k$ khác nhau.

Bảng 3. 4Bảng tính tỉ lệ dự đoán của đường cong ROC

| Tỉ lệ dự đoán Flase positive | Tỉ lệ dự đoán đúng | | |
|---|---|---|---|
| | LSB matching | LSB MR | Thuật toán đề xuất |
| 0.04 | 0.02 | 0.02 | 0.02 |
| 0.06 | 0.06 | 0.06 | 0.08 |
| 0.14 | 0.12 | 0.12 | 0.12 |
| 0.16 | 0.18 | 0.14 | 0.12 |
| 0.2 | 0.24 | 0.16 | 0.16 |

| | | | |
|---|---|---|---|
| 0.22 | 0.32 | 0.22 | 0.16 |
| 0.26 | 0.32 | 0.26 | 0.24 |
| 0.28 | 0.38 | 0.32 | 0.3 |
| 0.34 | 0.44 | 0.38 | 0.34 |
| 0.42 | 0.48 | 0.44 | 0.42 |
| 0.44 | 0.52 | 0.48 | 0.44 |
| 0.46 | 0.54 | 0.52 | 0.46 |
| 0.5 | 0.56 | 0.56 | 0.5 |
| 0.56 | 0.6 | 0.56 | 0.5 |
| 0.6 | 0.72 | 0.62 | 0.64 |
| 0.66 | 0.74 | 0.66 | 0.66 |
| 0.68 | 0.76 | 0.68 | 0.72 |
| 0.9 | 0.96 | 0.9 | 0.88 |
| 0.92 | 0.96 | 0.94 | 0.9 |
| 0.96 | 0.98 | 0.96 | 0.96 |
| 0.98 | 1 | 0.98 | 1 |




`Hình 3. 7 Đường cong ROC của HCF COM cho 3 thuật toán LSB matching, LSB-MR và thuật toán đề xuất.

Từ bảng 3.4 ta có hình 3.5 thể hiện các xác suất dự đoán phát hiện sai và xác suất phát hiện đúng với ngưỡng phát hiện thay đổi. Từ những đường cong ROC, ta thấy được rằng xác suất phát hiện của thuật toán tấn công HCF COM đã giảm so với thuật toán LSB-matching và LSB MR. Đạt được điều đó là vì đối với thuật toán đề xuất hình ảnh chứa thông tin mật có mức thay đổi bit trên mỗi pixel là ít hơn so với hai thuật toán trên.

# C. KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

*Những kết quả nghiên cứu đã đạt được:*

Luận văn đã đưa ra một hướng tiếp cận mới trong việc nhúng và rút trích dữ liệu và đặc biệt có thể áp dụng với các thuật toán ẩn dữ liệu hiện có như ẩn dữ liệu DFT, DCT, DWT, LSB, LSB Matching,…

Luận văn đã tiến hành thực nghiệm với phương pháp LSB-MR. Kết quả thực nghiệm cho thấy sự vượt trội của phương pháp đề xuất so với LSB-MR về dung lượng nhúng, cũng như khả năng vô hình. Mặt khác, cùng một lượng thông tin, phương pháp mới thay đổi ảnh chứa ít hơn so với LSB-MR, do đó thuật toán đề xuất làm cho tấn công HCF-COM kém hiệu quả hơn (so với tấn công trên LSB-MR). Nhưng bên cạnh đó ta thấy được rằng khả năng chịu tấn công hình ảnh (xén ảnh, biến đổi Afine) kém hơn so với thuật toán LSB-MR. Vì vậy nếu mục đích nhúng thông tin mật là hình ảnh có kích thước lơn thì việc sữ dụng thuật toán đề xuất là tốt nhất.

*Phương hướng phát triển:*

Trong phạm vi luận văn về cơ bản luận văn của em đã đạt được các yêu cầu đặt ra. Tuy nhiên do hạn chế về kiến thức và thời gian nghiên cứu nên các kết quả đạt được còn khá khiêm tốn. Đối với thuật toán đề xuất sử dụng thuật toán nén Fractal để nến và giải nén và về thực chất việc nén và giải nén đã cung cấp một bộ tạo sinh để vẽ lại ảnh gốc dựa trên một số nhỏ thông tin ban đầu. Do bộ công cụ vẽ này làm việc trên mọi độ phân giải, chúng ta có thể thu được ảnh giải nén ở bất kỳ độ phân giải nào, tức là có thể cho ra một ảnh mới có độ phân giải cao hơn và chất lượng ảnh vẫn không thay đổi mà tỉ số nén có thể đạt được rất cao. Bên cạnh đó để tăng khả năng chống tấn công hình ảnh thì ta nên áp dụng thêm phương pháp ẩn dữ liệu bằng phép biến đổi miền (DFT,DCT…).

# D. TÀI LIỆU THAM KHẢO

[1] Wang.H & Wang.S, "*Cyber warfare: Steganography vs. Steganalysis*," in communications of the ACM, vol.47, no.10, October 2004.

[2]Lê Hoài Bắc và Lê Thị Hoàng Ngân, "*Ẩn Dữ Liệu và Chia Sẻ Thông Tin*", NXB Đại Học Quốc Gia TP Hồ Chí Minh, 2011

[3] Chi- Kwong Chan, L.M. Cheng, "*Hiding data in images by siple LSB substitution*", Pattern Recognition 37 (2004) 469 -474.

[4] J. Mielikainen, "*LSB matching revisited*", IEEE Signal Process. Lett., vol. 13, no. 5, pp.285287, May 2006

[5] T. Sharp, "*An implementation of key-based digital signal steganography*," in Proc. Information Hiding Workshop, vol. 2137, Springer LNCS, 2001, pp. 1326.

[6] Shunquan Tan; Bin Li, "*Targeted steganalysis of adaptive pixel-value differencing steganography*," Image Processing (ICIP), 2012 19th IEEE International Conference on, pp. 1129 – 1132, Oct.3 2012.

[7]J.Harmsen and W.Pearlman, "*Steganalysis of additive-noise modelable information hiding*", in Proc. SPIE Security Watermarking Multimedia contents, vol. 5020, 2003, pp. 131142.

[8]L.F.Anson, "*Fractal image compression*", Bytes Magazine, 10/1993.

[9] M. F. Barnsley and S. G. Demko, *Iterated function systems and the global construction of fractals*, Proc. Roy. Soc. London A399 (1985), 2433–275.

[10] M. F. Barnsley and L. P. Hurd. "*Fractal Image Compression"* AK Peters Ltd, Wellesley, Ma, 1992

[11]A. Jacquin *A Fractal Theory of iterated Markov Operators with Applications to Digital Image Coding* . PhD thesis, Georgia Institute of Technology, August 1989.

[12] Andrew D.ker. "*Steganalysis of LSB Matching in Grayscale Image*". IEEE signal processing letter, Vol. 12, No. 6, June 2005

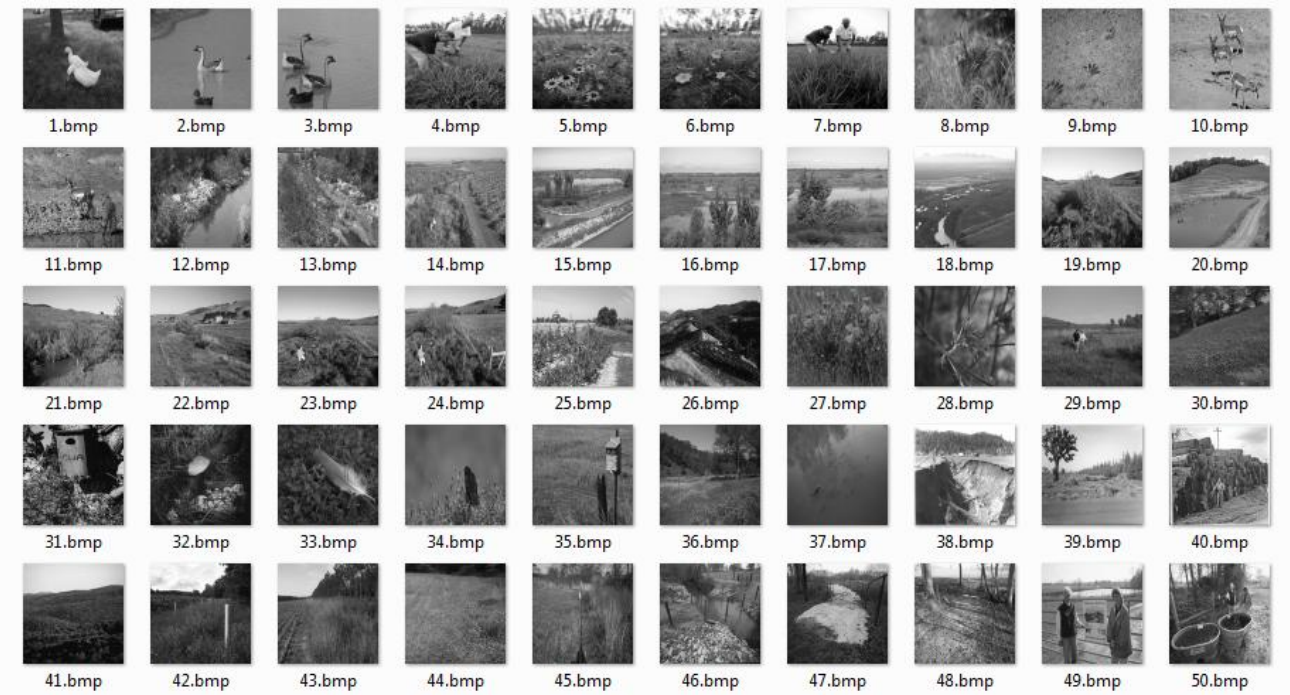
[13]. A. Westfeld and A. Pfitzmann, "*Attacks on steganographic systems*," in Proc. Information Hiding Workshop, Springer LNCS 1768, pp. 61–76, 1999.

[14]. J. Fridrich, M. Goljan, and R. Du, "*Reliable detection of LSB steganography in color and grayscale images*," Proc. ACM Workshop on Multimedia and Security, pp. 27–30, 2001.

[15]. G.Scheafer and M.Stich, "UCID: An uncompressed color image database" ,Proc. SPIE, Storage and Retrieval Methods and Appilcations for Multimedia 2004.

# PHỤ LỤC

***Phụ lục 1.*** 100 hình gray dùng làm ảnh chứa có kích thước 1024x1024, được gán tên lần lượt từ 1.bmp tới 100.bmp




***Phụ lục 2.*** 50 hình gray 8bit dùng làm ảnh ẩn có kích thước 256x256, được gán tên từ 1.bmp  tới 50.bmp, khi giấu tin ảnh chứa từ 1.bmp tới 50.bmp sẽ ẩn hình ẩn từ 1.bmp tới 50.bmp và ảnh chứa từ 51.bmp tới 100.bmp sẽ ẩn hình ẩn từ 50.bmp tới 1.bmp