MỤC LỤC

LỜI CẢM ƠN

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công nghệ thông tin - Trường ĐHDL Hải Phòng, những người đã nhiệt tình giảng dạy và truyền đạt những kiến thức cần thiết trong suốt thời gian em học tập tại trường, để em hoàn thành tốt quá trình tốt nghiệp.

Em xin tỏ lòng biết ơn sâu sắc đến cô Hồ Thị Hương Thơm, người đã trực tiếp hướng dẫn, giúp đỡ và truyền đạt cho em những kinh nghiệm để đề tài này có thể thực hiện được và hoàn thành.

Em xin cảm ơn gia đình và bạn bè đã động viên và giúp đỡ em trong suốt thời gian em làm đề tài tốt nghiệp.

Vì thời gian có hạn, trình độ hiểu biết của bản thân còn nhiều hạn chế. Cho nên trong đồ án không tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của tất cả các thầy cô giáo cũng như các bạn bè để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hải Phòng, ngày….. tháng ….. năm 2014

Sinh viên

Trần Đình Linh

LỜI MỞ ĐẦU

Sự phát triển vượt bậc của công nghệ mạng dẫn đến vấn đề an toàn thông tin trong là rất quan trọng. Có nhiều phương pháp để trao đổi thông tin mật, trong đó phương pháp mã hóa thông tin được coi là xuất hiện sớm nhất, tuy nhiên phương pháp này làm cho người ta dễ phát hiện. Do đó với một phương pháp khác giấu tin trong dữ liệu đa phương tiện được coi là "vô hình" đối với người dùng. Trong một số trường hợp để đảm bảo an toàn cho thông tin đem giấu người ta đã kết hợp cả hai phương pháp này. Trong đề tài này sẽ sử dụng phương pháp giấu tin Automata 2D-CA (two-dimensional cellular automata) để giấu thông tin vào trong ảnh.

Nôi dung báo cáo gồm 3 chương chính sau:

- Chương 1: Tổng quan về giấu tin trong ảnh.
  - Trình bày tổng quan về giấu tin trong ảnh.
- Chương 2: Phương pháp giấu tin dựa vào Automata 2D-CA.
  - Giới thiệu Automata 2D-CA.
  - Trình bày thuật toán giấu tin, tách tin Automata 2D-CA.
- Chương 3: Cài đặt và thử nghiệm.
  - Xây dựng chương trình ứng dụng và kết quả thu được.

CHƯƠNG 1: TỔNG QUAN VỀ GIẤU TIN

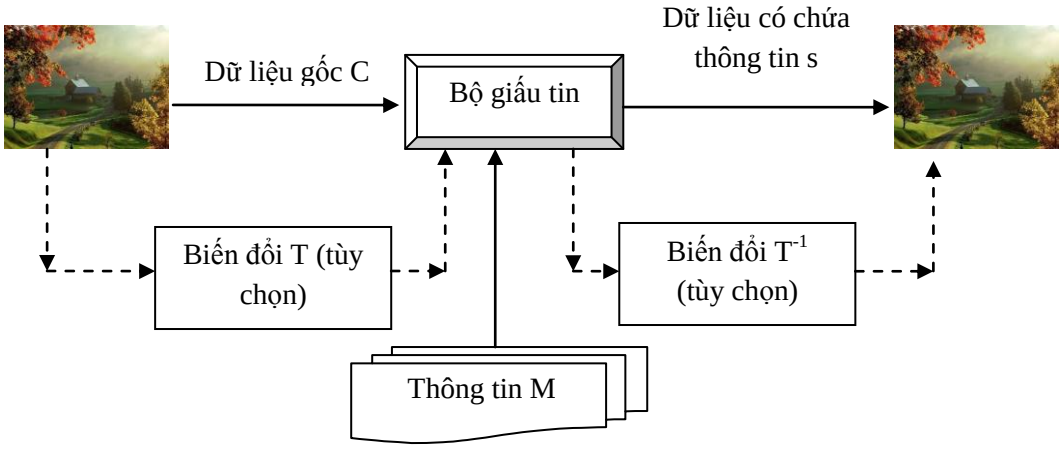
1.1 KHÁI NIỆM VỀ GIẤU TIN

Loài người đã biết đến nhiều phương pháp bảo vệ thông tin khác nhau, giải pháp được biết đến sớm nhất đó là các hệ mật mã. Với phương pháp này thông tin ban đầu được mã hóa, sau đó sẽ được giải mã nhờ khóa của hệ mã. Độ an toàn thông tin là do độ phức tạp của việc tìm ra khóa giải mã. Các hệ mật mã như RSA, DSA (Digital Signature Algorithm), NAPSACK… đã được sử dụng rất hiệu quả và phổ biến cho đến ngày nay. Một hướng nghiên cứu mới đã được thu hút sự quan tâm của nhiều người trong những năm gần đây đó là phương pháp giấu tin. Cho tới nay phương pháp giấu tin đã được ứng dụng mạnh mẽ ở nhiều nước trên thế giới. Vậy giấu tin là gì? Giấu tin là kỹ thuật nhúng (giấu) một lượng thông tin số nào đó vào một đối tượng số nào khác. Độ an toàn thông tin phương pháp này là do tính chất ẩn thông tin được giấu. Do đó yêu cầu cơ bản của giấu tin là không làm ảnh hưởng đến dữ liệu gốc.

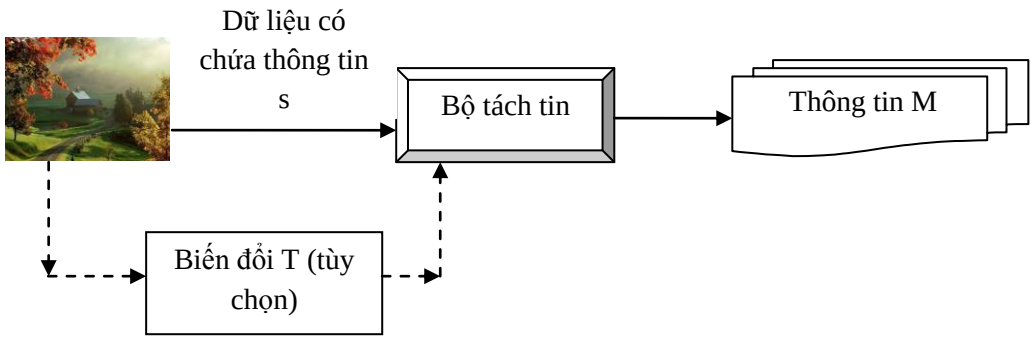
Các phương pháp giấu tin được tiến hành theo nhiều cách khác nhau tùy vào mục đích và môi trường giấu tin. Mỗi kỹ thuật giấu tin gồm :

- Thuật toán giấu tin
- Bộ giải mã thông tin

Thuật toán giấu tin được dùng để giấu thông tin vào một phương tiện mang bằng cách sử dụng một khóa bí mật được dùng chung bởi người mã và người giải mã.



Hình 1.1:Sơ đồ quá trình giấu tin trong ảnh



Hình 1.2: Sơ đồ quá trình tách tin trong ảnh

Hình 1.1 và 1.2 là sơ đồ tổng quát của quá trình giấu tin và tách tin trong ảnh, trong đó phép biến đổi T và T^{-1} là các phép biến đổi tần số cosine, wavelet, fourier rời rạc hoặc biến đổi sai phân (image difference).

Hình vẽ trên biểu diễn quá trình giấu tin cơ bản. Phương tiện chứa bao gồm các đối tượng được dùng làm mỗi trường để giấu tin như text, audio, video, ảnh … Thông tin giấu là mục đích của người sử dụng. Thông tin giấu là một lượng thông tin mang một ý nghĩa nào đó như ảnh, logo, đoạn văn bản… Tùy thuộc vào mục đích của người sử dụng. Thông tin sẽ được giấu vào trong phương tiện chứa thông qua chương trình. Sau khi giấu tin ta thu được phương tiện chưa bản tin đã giấu và phân bố trên mạng. Sau khi nhận được đối tượng phương tiện có giấu tin, quá trình giải mã được thực hiện thông qua chương trình giải mã tương ứng với chương trình mã hóa cùng với khóa của quá trình mã hóa. Kết quả thu được gồm phương tiện chứa gốc và thông tin đã giấu. Bước tiếp theo thông tin giấu sẽ được xử lý kiểm định so sánh với thông tin giấu ban đầu.

Tóm lại, giấu thông tin là nghệ thuật và khoa học của truyền thông, mục đích của giấu thông tin là che giấu những thông báo bên trong những thông báo khác mà không làm ảnh hưởng đáng kể đến thông báo này và bằng một cách thức nào đó sao cho người không có thẩm quyền không thể phát hiện hoặc không thể phá hủy chúng.

1.2 ĐẶC ĐIỂM GIẤU TIN

Hiện nay giấu thông tin trong ảnh là kỹ thuật còn tương đối mới và đang có xu hướng phát triển rất nhanh.

Một kỹ thuật giấu tin trong ảnh được đánh giá dựa trên một số đặc điểm sau:

- Tính vô hình của thông tin được giấu trong ảnh.

- Số lượng thông tin được giấu.
- Tính an toàn và bảo mật của thông tin.
- Chất lượng của ảnh sau khi giấu thông tin bên trong.

1.2.1 Tính vô hình của thông tin

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

1.2.2 Tính bảo mật

Thuật toán nhúng tin được coi là có tính bảo mật nếu thông tin được nhúng không bị tìm ra khi bị tấn công một cách có chủ đích trên cơ sở những hiểu biết đầy đủ về thuật toán nhúng tin và có bộ giải mã (trừ khoá bí mật), hơn nữa còn có được ảnh có mang thông tin (ảnh kết quả). Đối với giấu tin trong ảnh đây là một yêu cầu rất quan trọng. Chẳng hạn đối với thuật toán dò tin trong ảnh đen trắng kích thước m*n, độ phức tạp vẫn còn lên tới $O(2^{m*n})$ khi đã biết ma trận trọng số dùng trong quá trình giấu tin.

1.2.3 Tỷ lệ giấu tin

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

1.2.4 Lựa chọn ảnh

Đối với việc giấu thông tin mật thì hầu hết các chuyên gia về lĩnh vực này khuyên rằng: nên chọn ảnh đa cấp xám là môi trường là hợp lý hơn cả. Sau khi giấu tin mật trong ảnh đa cấp xám thì chất lượng của ảnh là cao, đảm bảo tính vô hình của thông tin mật.

1.2.5 Ảnh gốc đối với quá trình giải mã

Yêu cầu cuối cùng là thuật toán phải cho phép lấy lại được thông tin đã giấu trong ảnh mà không có ảnh gốc. Điều này là một thuận lợi khi ảnh môi trường là duy nhất nhưng lại làm giới hạn khả năng ứng dụng của kỹ thuật giấu tin. Để thực hiện việc giấu tin trong ảnh, trước hết ta phải xử lý được ảnh tức là phải số hoá ảnh. Quá trình số hoá các dạng ảnh khác nhau thì không như nhau. Có nhiều loại ảnh đã được chuẩn hoá như: JPEG, PCX, BMP,... Trong đồ án này chỉ sử dụng ảnh *.BMP.

Ảnh BMP (Bitmap) được phát triển bởi Microsoft Corporation, được lưu trữ dưới dạng độc lập thiết bị cho phép Windows hiển thị dữ liệu không phụ thuộc vào khung chỉ định màu trên bất kì phần cứng nào. Tên file mở rộng mặc định của một file ảnh Bitmap là BMP. Ảnh BMP được sử dụng trên Microsoft Windows và các ứng dụng chạy trên Windows từ version 3.0 trở lên.

Mỗi file ảnh Bitmap gồm 3 phần:

- Bitmap Header
- Palette màu
- BitmapData

Các cấu trúc cụ thể của ảnh Bitmap

Bitmap Header

Bảng 1.1 Cấu trúc ảnh Bitmap

Byte thứ	Ý nghĩa	Giá trị
1-2	Nhận dạng file	‘BM’ hay 1779
3-6	Kích thước file	Kiểu long trong Tubo C
7-10	Dự trữ	Thường mang giá trị 0
11-14	Byte bắt đầu vùng dữ liệu	Offset của byte bắt đầu vùng dữ liệu
15-18	Số byte cho vùng info	4 byte
19-22	Chiều rộng ảnh BMP	Tính bằng pixel
23-26	Chiều cao ảnh BMP	Tính bằng pixel
27-28	Số planes màu	Cố định là 1
29-30	Số bit cho 1 pixel	Có thể là 1, 4, 8, 16, 24 tùy theo loại ảnh

31-34	Kiểu nén dữ liệu	0: không nén 1: nén runlength 8bits/pixel 2: nén runlength 4bits/pixel
35-38	Kích thước ảnh	Tính bằng byte
39-42	Độ phân giải ngang	Tính bằng pixel/ metter
43-46	Độ phân giải dọc	Tính bằng pixel/ metter
47-50	Số màu sử dụng trong ảnh	
51-54	Số màu sử dụng khi hiển thị ảnh	

Palette màu: Bảng màu của ảnh, chỉ những ảnh nhỏ hơn hoặc bằng 8 bit màu mới có palette màu.

BitmapData: Phần này nằm ngay sau phần palette màu của ảnh BMP. Đây là phần chứa giá trị màu của điểm ảnh trong BMP. Các dòng ảnh được lưu từ dưới lên trên từ trái sang phải. Giá trị của mỗi điểm ảnh là một chỉ số trỏ tới phần tử màu tương ứng của Palette màu.

Thành phần Bit Count của cấu trúc Bitmap Header cho biết số bit dành cho mỗi điểm ảnh và số lượng màu lớn nhất của ảnh. Bit Count có thể nhận các giá trị sau:

1: Bimap là ảnh đen trắng, mỗi bit biểu diễn một điểm ảnh. Nếu bit mang giá trị 0 thì điểm ảnh là điểm đen, bit mang giá trị 1 điểm ảnh là điểm ảnh trắng.

4: Bitmap là ảnh 16 màu, mỗi điểm ảnh được biểu diễn bởi 4 bit.

8: Bitmap là ảnh 256 màu, mỗi điểm ảnh dược biểu diễn bởi 1 byte.

16: Bitmap là ảnh high color, mỗi dãy 2 byte liên tiếp trong bitmap biểu diễn cường độ tương đối của màu đỏ, xanh lá cây, xanh lơ của một điểm ảnh.

24: Bitmap là ảnh true color (224 màu), mỗi dãy 3 byte liên tiếp trong bitmap biểu diễn cường độ tương đối của màu đỏ, xanh lá cây, xanh lơ(RGB) của một điểm ảnh.

Thành phần Color Used của cấu trúc Bitmap Header xác định số lượng màu của palette màu thực sự được sử dụng để hiển thị bitmap. Nếu thành phần này được đặt là 0, bitmap sử dụng số màu lớn nhất tương ứng với giá trị của BitCount.

1.3 MÔI TRƯỜNG GIẤU TIN

Kỹ thuật giấu tin đã được nghiên cứu và áp dụng trong nhiều môi trường dữ liệu khác nhau như trong dữ liệu đa phương tiện (text, image, audio, video), trong sản phẩm phần mềm và gần đây là những nghiên cứu trên môi trường cơ sở dữ liệu quan hệ. Trong các môi trường dữ liệu đó thì dữ liệu đa phương tiện là môi trường chiếm tỉ lệ chủ yếu trong các kỹ thuật giấu tin.

1.3.1 Giấu tin trong ảnh

Giấu thông tin trong ảnh hiện nay chiếm tỉ lệ lớn nhất trong các chương trình ứng dụng và phần mềm, hệ thống giấu tin trong đa phương tiện bởi lượng thông tin được trao đổi bằng ảnh là rất lớn, hơn nữa giấu thông tin trong ảnh cũng đóng vai trò hết sức quan trọng đối với 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, điều khiển truy cập, giấu thông tin mật...

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 đem lại rất nhiều ứng dụng quan trọng trên nhiều lĩnh vực đời sống xã hội. Ví dụ, trong các dịch vụ ngân hàng và tài chính ở một số nước phát triển, thuỷ vân số được sử dụng để nhận diện khách hàng trong các thẻ tín dụng. Mỗi khách hàng có một chữ kí viết tay, sau đó chữ kí này được số hoá và lưu trữ trong hồ sơ của khách hàng. Chữ kí này sẽ được sử dụng như là thuỷ vân để nhận thực thông tin khách hàng. Trong các thẻ tín dụng, chữ kí tay được giấu trong ảnh của khách hàng trên thẻ. Khi sử dụng thẻ, người dùng đưa thẻ vào một hệ thống, hệ thống có gắn thiết bị đọc thuỷ vân trên ảnh và lấy được chữ kí số đã nhúng trong ảnh. Thuỷ vân được lấy ra sẽ so sánh với chữ kí số đã lưu trữ xem có trùng hợp không, từ đó xác định nhận thực khách hàng.

1.3.2 Giấu tin trong audio

Giấu tin trong audio mang đặc điểm riêng, không giống với giấu tin trong 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 gốc. Để đảm bảo yêu cầu này, kỹ thuật giấu tin trong ảnh phụ thuộc vào hệ thống thị giác của con người – HVS (Human Vision System), kỹ thuật giấu tin trong audio lại phụ thuộc vào hệ thống thính giác HAS (Human Auditory System).

Một vấn đê 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 thật may là HAS lại kém trong việc phát hiện sự khác biệt các dải tần và công suất, điều này có nghĩa là các âm thanh to, cao tần có thể che giấu được các âm thanh nhỏ thấp một cách dễ dàng. Các mô hình phân tích tâm lí đã chỉ ra điểm yếu trên và thông tin này sẽ giúp ích cho việc chọn các audio thích hợp cho việc giấu tin.

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. Ví dụ để nhúng một đoạn mã java applet vào một đoạn audio (16 bit, 44,100 Hz) có chiều dài bình thường, thì các phương pháp nói chung cũng cần ít nhất là 20 bit/s.

Giấu tin trong audio đòi hỏi yêu cầu rất cao về tính đồng bộ và tính an toàn của thông tin. Các phương pháp giấu tin trong audio đều lợi dụng điểm yếu trong hệ thống thính giác của con người.

1.3.3 Giấu tin trong video

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

Các kỹ thuật giấu tin trong video phát triển mạng mẽ và cũng theo hai khuynh hướng là thủy vân số và giấu dữ liệu. Một phương pháp giấu tin trong video được đưa ra bởi Cox là phương pháp phân bố đều. Ý tưởng cơ bản là phân phối tin giấu dàn trải theo tần số của dữ liệu chứa (gốc). Người ta đã dùng hàm cosin riêng và hệ số truyền sóng riêng để giấu tin.

Trong các thuật toán khởi nguồn, kỹ thuật cho phép giấu tin vào video, nhưng thời gian gần đây các kỹ thuật cho phép giấu tin cả âm thanh và hình ảnh vào video. Phương pháp Swanson đã giấu theo khối, đã giấu được 2 bit vào khối 8*8. Gần đây nhất là phương pháp Mukherjee, giấu audio vào video sử dụng cấu trúc lưới đa chiều.

Kỹ thuật giấu tin sử dụng cả đặc điểm thị giác và thính giác của con người.

1.3.4 Giấu tin trong văn bản dạng text

Giấu tin trong văn bản dạng text khó thực hiện hơn do đó 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 hóa 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.

1.4 PHƯƠNG PHÁP GIẤU TIN

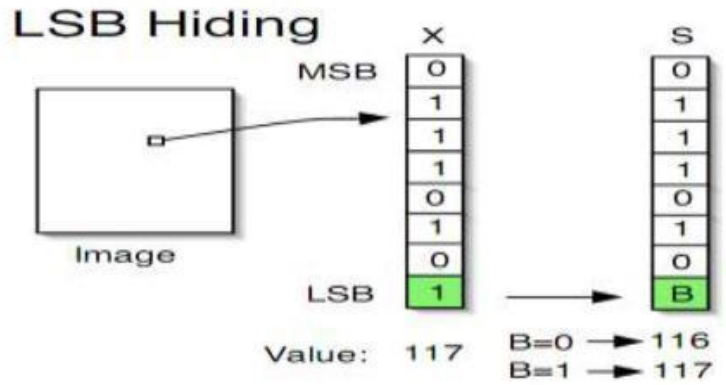
Kỹ thuật giấu tin trong ảnh ra đời dựa trên sự phát triển ưu việt của kỹ thuật thủy vân số (Watermarking), phương pháp thủy vân ảnh số đầu tiên là phương pháp thủy vân trên LSB của ảnh hay còn gọi là phương pháp thay thế LSB (LSB replacement - LSB hiding) và nó cũng trở thành phương pháp giấu tin đầu tiên trong ảnh [1].

Phương pháp giấu tin trên LSB là phương pháp thay thế các bit thông tin vào bit LSB của điểm ảnh. Trong một điểm ảnh của ảnh 8-bit cấp độ xám có thể biểu diễn dưới dạng chuỗi nhị phân 8 bit (giả sử điểm ảnh P có giá trị 236 có thể biểu diễn thành chuỗi nhị phân 8 bit là "11101100") thì 7 bit liên tiếp đầu tiên (là chuỗi bit "1110110") gọi là các bit MSBs (Most Significant Bit) có ý nghĩa quan trọng nhất đối với điểm ảnh, còn bit cuối cùng (bit "0") gọi là bit LSB (least significant bit) vì có ảnh hưởng ít nhất đến sự thể hiện của điểm ảnh. Do vậy, việc thay đổi giá trị của bit LSB (từ "0" sang "1" hay từ "1" sang "0") không làm ảnh hưởng nhiều đến chất lượng trực quan của ảnh.

Kỹ thuật giấu tin trên LSB vẫn còn được ưa chuộng cho đến ngày nay ở chỗ nó rất đơn giản và có khả năng giấu được nhiều thông tin. Mỗi điểm ảnh có thể nhúng được một bit thông tin, do đó tỉ lệ nhúng lớn nhất là một bit thông tin trên một điểm ảnh (hay độ dài bit thông tin có thể nhúng bằng số điểm ảnh của ảnh).

Để đơn giản, giả sử ảnh gốc đầu vào để giấu tin là ảnh xám 8 - bit kích cỡ m×n điểm ảnh, dữ liệu ảnh được biểu diễn dưới dạng vector $X_{m\times n} = \{x_{ij}, i=1, \ldots, m, j=1, \ldots, n, x_{ij} \in \{0, \ldots, 255\}\}$. Sau khi giấu chuỗi bit thông tin $b_l = \{bi, i = 1, \ldots, l,$

$b_i \in \{0,1\}\}$ vào ảnh bằng cách thay thế từng bit $b_i \in B$ vào từng bit LSB của x_{ij} theo thứ tự nào đó ta nhận được ảnh có giấu tin với vector $S_{m\times n} = \{s_{ij}, i=1, \ldots, m, j=1, \ldots, n, s_{ij} \in \{0, \ldots, 255\}\}$ tương ứng. Khi đó LSB của điểm ảnh được giấu tin theo mô tả như hình 1.3 (giấu trên điểm ảnh có giá trị bằng 117).



Hình 1.3. Minh họa giấu thông tin trong LSB của ảnh cấp xám 8 - bit.

Việc áp dụng hàm giấu và tách thông tin có thể thực hiện tương tự trên ảnh 24 - bit màu với 3 kênh màu R, G, B (mỗi kênh 8 - bit), khi đó việc giấu tin thường thực hiện trên kênh màu B (được cho là ít ảnh hưởng đến hệ thống cảm nhận của mắt người) như quá trình giấu tin trên ảnh 8 - bit cấp độ màu. Để đảm bảo ảnh sau khi đã giấu tin bằng kỹ thuật giấu LSB trên miền không gian không bị phá vỡ bằng một số phép tấn công hình học như xoay, nén, co, giãn, … người ta đề xuất một số phương pháp giấu cải tiến LSB khác trên miền tần số: cosine, wavelet. Một số khác còn giấu trên LSB của các hệ số sai phân. Bit LSB của điểm ảnh hay của hệ số biến đổi được chọn để giấu thông tin có thể chọn theo thứ tự tuần tự (quét raster) (như kỹ thuật giấu EzStego, Jstego, DE, …) hoặc theo thứ tự ngẫu nhiên dựa trên một bộ chọn vị trí giả ngẫu nhiên PR (Pseudo Random) (như kỹ thuật giấu Out Guess, F5, Hideand Seek, …). Ngoài ra còn có hai trường đặc biệt giấu trên LSB đó là: phương pháp tăng giảm LSB, phương pháp đồng chẵn lẻ.

Phương pháp tăng giảm LSB (±1 embedding), bit thông tin sẽ được so sánh với bit LSB của điểm ảnh được chọn (việc chọn điểm ảnh có thể là tuần tự hoặc ngẫu nhiên theo bộ chọn PR). Nếu bit thông tin cùng giá trị với bit LSB của điểm ảnh cần giấu thì mặc định sẽ giấu một bit thông tin vào điểm ảnh này, ngược lại điểm ảnh cần giấu sẽ tăng hoặc giảm đi 1 để LSB của nó đồng giá trị với bit thông tin.

Phương pháp đồng chẵn lẻ, chia miền không gian ảnh ra thành nhiều khối bằng nhau kích thước k × t, bit thông tin sẽ được giấu vào từng khối theo quy tắc: số bit LSB có giá trị "1" của khối phải đồng tính chẵn lẻ với bit được giấu, tức là số bit "1" của một khối LSB là lẻ nếu bit thông tin cần giấu là "1", ngược lại là chẵn nếu bit cần giấu là "0". Trong trường hợp không trùng hợp, ta phải thay đổi giá trị LSB của khối đó để đảm bảo đồng tính chẵn lẻ với bit thông tin. Trường hợp đặc biệt, nếu kích thước mỗi khối dùng để giấu tin là 1×1, thu nó trở thành trường hợp giấu thay thế LSB tổng quát.

Có thể có nhiều phương pháp giấu LSB khác nhau không tuân theo bốn phương pháp đã nêu ở trên, đó là các phương pháp kết hợp với một trong bốn phương pháp trên (phương pháp tuần tự, phương pháp ngẫu nhiên, phương pháp tăng giảm, phương pháp đồng chẵn lẻ) cùng với một số thao tác nào đó nhằm nâng cao hiệu quả an toàn cho thông tin được giấu.

Ngoài phương pháp giấu trên LSB còn có một số phương pháp giấu tin khác theo hình thức chèn nhiễu SS hay điều chỉnh hệ số lượng tử QIM như sau:

Kỹ thuật giấu tin theo hình thức chèn nhiễu SS: Dữ liệu đem giấu sẽ được điều biến thành một chuỗi tín hiệu mang thông tin theo một hệ số bền vững α, sau đó được chèn vào dữ liệu ảnh gốc. Với cách thức giấu tin theo kiểu SS đã có nhiều phương pháp được đề xuất. Điển hình như phương pháp của J.Cox, ảnh gốc sẽ được biến đổi Cosine và chọn ra một lượng hệ số DCT x_k ở miền tần số giữa có giá trị lớn nhất bằng độ dài tín hiệu thông tin cần giấu, các tín hiệu thông tin d_k trong chuỗi thông tin sẽ được chèn vào các hệ số x_k này theo một trong ba công thức sau: $s_k = x_k + \alpha d_k$, $s_k = x_k + (\alpha x_k)\ d_k = x_k\ (1+\alpha d_k)$ hoặc $s_k = x_k e\alpha dk$. Theo J.Cox, các biểu thức hiệu chỉnh này cho phép giấu thông tin bền vững trong ảnh trước các tấn công nhiễu và một số phép biến đổi hình học.

Kỹ thuật giấu tin điều chỉnh hệ số lượng tử QIM: là một phương pháp giấu khá phổ biến mặc dù kỹ thuật giấu hơi phức tạp và khả năng giấu thấp hơn kỹ thuật giấu LSB, nhưng cũng giống như kỹ thuật giấu SS, QIM làm cho thông tin có thể bền vững trước các tấn công hình học và nhiễu. Giả sử coi dữ liệu của ảnh gốc và ảnh có giấu tin là các tín hiệu ký hiệu lần lượt là $\{x_n\}^N_{n=1}$ và $\{s_n\}^N_{n=1}$, M là chuỗi thông tin cần giấu, khi đó ta có $S(X, M)=q_M(X)$. Tín hiệu của ảnh có giấu tin bao gồm các giá trị trong tập lượng tử đầu ra, do đó sẽ hạn chế cho trường hợp nén dữ liệu, sẽ làm mất thông tin đã giấu. Để có thể cung cấp một tín hiệu ảnh giấu tin bao

phủ tất cả các giá trị của tín hiệu gốc, việc lượng tử sẽ được dịch chuyển theo một mức thay đổi nhỏ D bằng biểu thức S(X, M) = q(X + D(M)) - D(M) với q_M là hàm lượng tử, D là hàm điều chỉnh lượng tử.

Thời gian gần đây do đặc thù của một số lĩnh vực: y học, quân sự, nghiên cứu năng lượng hoặc hệ thống thông tin vệ tinh, … đòi hỏi yêu cầu sau khi tách thông tin chúng ta có thể khôi phục lại ảnh gốc ban đầu. Vì vậy kỹ thuật giấu tin thuận nghịch ra đời. Năm 1999, Honsinger và các công sự đề xuất kỹ thuật giấu thuận nghịch đầu tiên, mở ra một hướng mới trong lĩnh vực giấu tin. Tiếp đó một loạt các kỹ thuật giấu tin thuận nghịch khác được công bố. Sau đây giới thiệu sơ lược một số kỹ thuật giấu tin biểu.

Kỹ thuật mở rộng sai phân DE (Difference Expansion) do Tian đưa ra (2002), đây là kỹ giấu tin dựa trên mở rộng hệ số sai phân của điểm ảnh, dữ liệu ảnh được tính sai phân, thông tin được giấu trên LSB của các hệ số sai phân sau khi được mở rộng. Sau đó tác giả đề xuất tiếp phương pháp mở rộng trên các hệ số wavelet để giấu tin. Đến năm 2008, Shaowei Weng và các đồng nghiệp đưa ra kỹ thuật DE cải tiến bằng cách thêm vào hàm nén - giãn trong quá trình giấu tin sử dụng DE nhằm giảm nhiễu xẩy ra (theo đánh giá bằng PSNR) của kỹ thuật giấu thuận nghịch DE.

Năm 2003, Ni và cộng sự đề xuất kỹ thuật giấu thuận nghịch dựa trên dịch chuyển biểu đồ tần suất gọi là NSAS. Tiếp đó một loạt các kỹ thuật giấu thuận nghịch dựa phương pháp này ra đời: kỹ thuật DIH (2004) (dịch chuyển biểu đồ tần suất hệ số sai phân), kỹ thuật HKC (cải tiến kỹ thuật giấu NSAS), kỹ thuật IWH (2006) (dựa trên dịch chuyển biểu đồ tấn suất hệ số nguyên wavelet), kỹ thuật RL (2008) là kỹ thuật giấu thuận cho ảnh nhị phân dựa trên dịch chuyển tần suất của các loạt đen trong ảnh.

Một số kỹ thuật giấu thuận nghịch khác không dựa trên biểu đồ tần suất như: kỹ thuật giấu MBNS (Multiple-Base Notational System): dữ liệu cần giấu được chuyển đổi thành các hệ số nhỏ hơn theo phương pháp phân tích nhân tử thành đa thức, các điểm ảnh sẽ được điều chỉnh để lưu trữ các hệ số này, kỹ thuật giấu RCM dựa trên hiệu chỉnh LSB của ảnh theo bản đồ màu tương phản. Kỹ thuật giấu hai pha ngang dọc RVH, chuỗi thông tin giấu M được chia thành hai chuỗi con bằng nhau M1 và M2, sau đó được giấu lần lượt vào hai pha: pha giấu ngang, thực hiện

giấu trên các cột lẻ của ma trận ảnh: pha giấu dọc, thực hiện giấu trên các hàng chẵn của ma trận ảnh.

1.5 PHƯƠNG PHÁP ĐÁNH GIÁ ĐỘ AN TOÀN CỦA MỘT LƯỢC ĐỒ GIẤU TIN

Khi một kỹ thuật giấu tin được đề xuất, từ đòi hỏi "khó có thể cảm nhận bằng mắt thường" hay "không thể phát hiện bằng phương pháp thống kê" Cachin đã đưa ra một khái niệm về giấu tin an toàn.

Đặt C' ký hiệu là tập tất cả các ảnh gốc C, M' là tập các thông tin mật M, K' là tập các khóa K giấu tin, S' là tập tất cả các ảnh stego S. Một lược đồ giấu tin (thuật toán) là một cặp (S_E, S_X), với S_E: C' × M' × K' ->S' là hàm nhúng thông tin và S_X : S' × K'->M' là hàm tách thông tin. Hàm nhúng S_E tạo ra một đối tượng S ∈ S' từ mỗi C ∈ C', M ∈ M' và K ∈ K', tương tự hàm tách S_X tách thông tin M từ S bằng khóa K.

Giả sử hàm phân bố xác xuất của C ∈ C'. Nếu khóa K ∈ K' và M ∈ M' được chọn ngẫu nhiên thì lược đồ giấu tin (S_E, S_X) cùng với hàm phân bố xác suất P_C sẽ được hàm phân bố xác suất P_S tương ứng của S ∈ S'. Khi đó theo khái niệm về giấu tin an toàn của Cachin ta có định nghĩa sau:

Định nghĩa 1.1 - Một lược đồ (thuật toán) giấu tin được gọi là an toàn nếu sai phân Kullback - Leibler giữa hàm mật độ xác suất của P_C và P_S theo (1.1) là bằng 0

$$D_{KL}(PC \,||\, PS) = \sum_{C \in C'} P_C(C) \log \frac{PC(C)}{PS(S)} \quad (1.1)$$

Khi $DKL(P_C \,||\, P_S) < \varepsilon$ thì lược đồ giấu tin có độ an toàn ε (ε - secure), trong đó ε là một số thực dương đủ nhỏ tùy ý cho trước.

Đây là khái niệm đứng từ quan điểm lý thuyết, nó rất khó thực hiện trong thực tế vì một lược đồ giấu tin để đảm bảo $D_{KL}(P_C \,||\, P_S) = 0$ là không thể vì điều này nghĩa là không thay đổi gì trên ảnh gốc, tức là $P_C = P_S$ (theo bổ đề cơ bản trong lý thuyết thông tin). Vì vậy, người ta thường giấu sao cho đạt độ an toàn ε – secure đảm bảo thay đổi trên ảnh nhỏ nhất mà mắt người không thể cảm nhận.

Tuy nhiên, rất nhiều lược đồ giấu tin chủ yếu sử dụng đánh giá khả năng cảm nhận của con người dựa vào độ đo PSNR (Peaksignal to noise ratio) giữa ảnh gốc ban đầu và ảnh sau khi giấu tin. PSNR là phương pháp đánh giá độ an toàn dựa theo hướng tiếp cận chủ quan. Theo hướng tiếp cận này thì cảm nhận của con người

được phân làm năm mức khác nhau. Trên mỗi mức, chất lượng ảnh sẽ được tính theo PSNR, sau đó tùy vào giá trị tính được mà ảnh sẽ được đánh giá là thuộc vào ngưỡng nào. Chất lượng PSNR được ánh xạ vào thang đo đánh giá bình quân MOS (Mean Opinion Score) theo bảng 1.1.

Bảng 1.2. Mối quan hệ giữa các giá trị PSNR và MOS

PSNR [dB]	MOS
>37	5 (rất tốt)
31 – 37	4 (tốt)
25 – 31	3 (Trung bình)
20 – 25	2 (Tồi)
< 20	1 (Rất tồi)

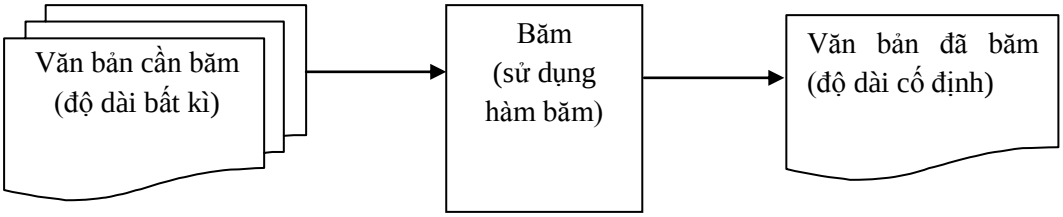
Nhiều kỹ thuật giấu tin như thường cố gắng tác động lên ảnh sau khi giấu tin làm cho chất lượng ảnh theo đánh giá PSNR nằm ở mức 5, với giá trị của PSNR từ 39 dB - 46 dB.

1.6 HÀM BĂM

1.6.1 Định nghĩa tổng quát của hàm băm

Hàm h(x) được gọi là một hàm băm nếu thỏa mãn hai tính chất sau:

- Nén (conpression): hàm h(x) tương ứng chuỗi bit đầu vào có chiều dài hữu hạn tùy ý được chuỗi bit ra y= h(x) có chiều dài cố định n>0 cho trước.
- Dễ tính toán (ease of computation): với mọi bit đầu vào x có chiều dài hữu hạn (tùy ý), h(x) được tính toán "dễ dàng".



Hình 1.4:Minh họa hàm băm

Các định nghĩa hàm băm:

- Định nghĩa 1: hàm băm h là hàm không va chạm yếu nếu khi cho trước một bức điện x không thể tiến hành về mặt tính toán để tìm được một bức điện x'≠ x sao cho h(x') = h(x)
- Định nghĩa 2: hàm băm h là hàm không va chạm yếu nếu không có khả năng tính toán để tìm ta bức điện x và x' sao cho x ≠ x' và h(x) = h(x').
- Định nghĩa 3: hàm băm là hàm một chiều nếu khi cho trước một bản tóm lược thông bao không thể thực hiện về mặt tính toán để tìm bức điện x sao cho h(x) = z.[2]

1.6.2 Một số tính chất cơ bản của hàm băm.

- **Tính kháng tiền ảnh (preimage resistance)**: với mọi đầu ra y cho trước, không thể tính toán để tìm được bất kì dữ liệu đầu vào x' nào sao cho giá trị hàm băm h(x') của nó bằng giá trị đầu ra y đã cho.
- **Tính kháng tiềm ảnh Thứ hai (2^{nd}- preimage resistance):** với mọi dữ liệu x_2 nào ($x_2 \neq x_1$) mà giá trị hàm băm $h(x_2)$ của nó bằng giá trị băm $h(x_1)$ của x_1.
- **Tính kháng xung đột (collision resistase):** không thể tính toán để hai dữ liệu đầu vào x_1 và x_2 phân biệt sao cho chúng nó bằng giá trị băm (tức là $h(x_1) = h(x_2)$).

Bảng1.3: Danh sách các hàm băm mật mã học.

Thuật toán	Kích thước đầu ra (*output size*)	Kích thước trạng thái trong (*Internal state size*)	Kích thước khối (*Block size*)	Độ dài (*Length size*)	Kích thước word (*Word size*)	Xung đột (*Collision*)
HAVAL	256/224/192/160/128	256	1024	64	32	Có
MD2	128	384	128	Không	8	khả năng lớn
MD4	128	128	512	64	32	Có
MD5	128	128	512	64	32	Có
PANAMA	256	8736	256	No	32	Có lỗi
RIPEMD	128	128	512	64	32	Có
RIPEMD-128/256	128/256	128/256	512	64	32	Không
RIPEMD-160/320	160/320	160/320	512	64	32	Không
SHA-0	160	160	512	64	32	Không
SHA-1	160	160	512	64	32	Có lỗi
SHA-256/224	256/224	256	512	64	32	Không
SHA-512/384	512/384	512	1024	128	64	Không
Tiger(2)-192/160/128	192/160/128	192	512	64	64	Không
VEST-4/8 (hash mode)	160/256	256/384	8	80/128	1	Không
VEST-16/32 (hash mode)	320/512	512/768	8	160/256	1	Không
WHIRLPOOL	512	512	512	256	8	Không

1.6.3 Hàm băm MD5

Thuật toán MD5 do Ronald Rivest thiết kế năm 1991 đại học MIT

- Input: thông điệp có độ dài bất kì.
- Output: giá trị băm 128 bit.

Giải thuật gồm 5 bước thao tác trên khối 512 bit.

Bước 1: Nhồi dữ liệu.

- Nhồi thêm các bit sao cho dữ liệu có độ dài l =448 mod 512 hay l=n* 512+448 (l, n số nguyên).
- Luôn thực hiện nhồi dữ liệu ngay cả khi dữ liệu ban đầu có độ dài mong muốn. Ví dụ dữ liệu có độ dài 448 được nhồi thêm 512 bit để được độ dài 960 bit.
- Số lượng bit nhồi thêm nằm trong khoảng 1 đến 512.
- Các bit được nhồi thêm gồm 1 bit "1" và các bit 0 theo sau.

Bước 2: thêm vào độ dài

- Độ dài của khối dữ liệu ban đầu được biểu diễn dưới dạng nhị phân 64 bit và được thêm vào cuối chuỗi nhị phân kết quả của bước 1.

- Nếu độ dài của khối dữ liệu ban đầu > 2^{64}, chỉ 64 bit thấp nhất được sử dụng, nghĩa là giá trị được thêm vào bằng K mod 264.
- Kết quả có được từ 2 bước đầu là khối dữ liệu có độ dài là bội số của 512. Khối dữ liệu được biểu diễn
  - Bằng 1 dãy L khối 512 bit Y_0, Y_1,….,Y_{L-1}
  - Bằng 1 dãy N từ 32 bit M_0, M_1,…,M_{N-1}. Vậy N = L*16(32*16=512)

Bước 3: khởi tạo bộ đệm MD

- Một bộ đệm 128 bit
- Một bộ đệm được biểu diễn bằng 4 thanh dùng lưu trữ các giá trị băm trung gian và kết quả ghi 32 bit với các giá trị khởi tạo dưới dạng little-endien (byte có trọng số nhỏ nhất trong từ nằm địa chỉ thấp nhất) như sau: A= 067452301;
  B= 0xefcdab89;
  C= 0x98badcfe;
  D= 0x10324576;

Bước 4: xử lý các khối dữ liệu 512 bit

Trọng tâm của giải thuật là hàm nén gồm 4 vòng xử lý. Các vòng này có cấu trúc giống nhau nhưng sử dụng các hàm luận lý khác nhau gồm F, G, H và I.

$F(X,Y,Z)= X \wedge Y \vee \bar{X} \wedge Z$

$G(X,Y,Z) = X \wedge Z \vee Y \wedge \bar{Z}$

$H(X,Y,Z) = X \text{ xor } Y \text{ xor } Z$

$I(X,Y.Z)= Y \text{ xor } (X \vee \bar{Z})$

Mảng 64 phần tử được tính theo công thức : $T[i]= 2^{32} x \text{ abs } (\sin(i))$, I tính theo radian. Kết quả của 4 vòng được cộng theo modulo 2^{32} với đầu vào CV_p để tạo CV_{q+1}

Bảng 1.4: Các giá trị trong bảng T

T[1] = d76aa478	T[17] = f61e2562	T[33] = fffa3942	T[49] = f4292244
T[2] = e8c7b756	T[18] = c040b340	T[34] = 8771f681	T[50] = 432aff97
T[3] = 242070db	T[19] = 265e5a51	T[35] = 6d9d6122	T[51] = ab9423a7
T[4] = c1bdceee	T[20] = e9b6c7aa	T[36] = fde5380c	T[52] = fc93a039
T[5] = f57c0faf	T[21] = d62f105d	T[37] = a4beea44	T[53] = 655b59c3
T[6] = 4787c62a	T[22] = 2441453	T[38] = 4bdecfa9	T[54] = 8f0ccc92
T[7] = a8304613	T[23] = d8a1e681	T[39] = f6bb4b60	T[55] = ffeff47d
T[8] = fd469501	T[24] = e7d3fbc8	T[40] = bebfbc70	T[56] = 85845dd1
T[9] = 698098d8	T[25] = 21e1cde6	T[41] = 289b7ec6	T[57] = 6fa87e4f
T[10] = 8b44f7af	T[26] = c33707d6	T[42] = eaa127fa	T[58] = fe2ce6e0
T[11] = ffff5bb1	T[27] = f4d50d87	T[43] = d4ef3085	T[59] = a3014314
T[12] = 895cd7be	T[28] = 455a14ed	T[44] = 4881d05	T[60] = 4e0811a1
T[13] = 6b901122	T[29] = a9e3e905	T[45] = d9d4d039	T[61] = f7537e82
T[14] = fd987193	T[30] = fcefa3f8	T[46] = e6db99e5	T[62] = bd3af235
T[15] = a679438e	T[31] = 676f02d9	T[47] = 1fa27cf8	T[63] = 2ad7d2bb
T[16] = 49b40821	T[32] = 8d2a4c8a	T[48] = c4ac5665	T[64] = eb86d391

Bảng 1.5: Bảng T chính là giá trị của 4 chu kì của FF, GG, HH, II.

Chu kỳ 1	Chu kỳ 2
FF(a,b,c,d,M0 , 7,0xd76aa478)	GG(a,b,c,d,M1 , 5,0xf61e2562)
FF(d,a,b,c,M1 ,12,0xe8c7b756)	GG(d,a,b,c,M6 , 9,0xc040b340)
FF(c,d,a,b,M2 ,17,0x242070db)	GG(c,d,a,b,M11,14,0x265e5a51)
FF(b,c,d,a,M3 ,22,0xc1bdceee)	GG(b,c,d,a,M0 ,20,0xe9b6c7aa)
FF(a,b,c,d,M4 , 7,0xf57c0faf)	GG(a,b,c,d,M5 , 5,0xd62f105d)
FF(d,a,b,c,M5 ,12,0x4787c62a)	GG(d,a,b,c,M10, 9,0x02441453)
FF(c,d,a,b,M6 ,17,0xa8304613)	GG(c,d,a,b,M15,14,0xd8a1e681)
FF(b,c,d,a,M7 ,22,0xfd469501)	GG(b,c,d,a,M4 ,20,0xeid3fbc8)
FF(a,b,c,d,M8 , 7,0x698098d8)	GG(a,b,c,d,M9 , 5,0x21e1cde6)
FF(d,a,b,c,M9 ,12,0x8b44f7af)	GG(d,a,b,c,M14, 9,0xc33707d6)
FF(c,d,a,b,M10,17,0xffff5bbl)	GG(c,d,a,b,M3 ,14,0xf4d50d87)
FF(b,c,d,a,M11,22,0x895cd7be)	GG(b,c,d,a,M8 ,20,0x455a14ed)
FF(a,b,c,d,M12, 7,0x6b901122)	GG(a,b,c,d,M13, 5,0xa9e3e905)
FF(d,a,b,c,M13,12,0xfd987193)	GG(d,a,b,c,M2 , 9,0xfcefa3f8)
FF(c,d,a,b,M14,17,0xa679438e)	GG(c,d,a,b,M7 ,14,0x676f02d9)
FF(b,c,d,a,M15,22,0x49b40821)	GG(b,c,d,a,M12,20,0x8d2a4c8a)

Chu kỳ 3	Chu kỳ 4
HH(a,b,c,d,M5 , 4,0xfffa3942)	II(a,b,c,d,M0 , 6,0xf4292244)
HH(d,a,b,c,M8 ,11,0x8771f6811	II(d,a,b,c,M7 ,10,0x432aff97)
HH(c,d,a,b,M11,16,0x6d9d6122)	II(c,d,a,b,M14,15,0xab9423a7)
HH(b,c,d,a,M14,23,0xfde5380c)	II(b,c,d,a,M5 ,21,0xfc93a039)
HH(a,b,c,d,M1 , 4,0xa4beea44)	II(a,b,c,d,M12, 6,0x655b59c3)
HH(d,a,b,c,M4 ,11,0x4bdecfa9)	II(d,a,b,c,M3 ,10,0x8f0ccc92)
HH(c,d,a,b,M7 ,16,0xf6bb4b60)	II(c,d,a,b,M10,15,0xffeff47d)
HH(b,c,d,a,M10,23,0xbebfbc70)	II(b,c,d,a,M1 ,21,0x85845dd1)
HH(a,b,c,d,M13, 4,0x289biec6)	II(a,b,c,d,M8 , 6,0x6fa87e4f)
HH(d,a,b,c,M0 ,11,0xeaal27fa)	II(d,a,b,c,M15,10,0xfe2ce6e0)
HH(c,d,a,b,M3 ,16,0xd4ef3085)	II(c,d,a,b,M6 ,15,0xa3014314)
HH(b,c,d,a,M6 ,23,0x04881d05)	II(b,c,d,a,M13,21,0x4e0811a1)
HH(a,b,c,d,M9 , 4,0xd9d4d039)	II(a,b,c,d,M4 , 6,0xf7537e82)
HH(d,a,b,c,M12,11,0xe6db99e5)	II(d,a,b,c,M11,10,0xbd3af235)
HH(c,d,a,b,M15,16,0x1fa27cf8)	II(c,d,a,b,M2 ,15,0x2ad7d2bb)
HH(b,c,d,a,M2 ,23,0xc4ac5665)	II(b,c,d,a,M9 ,21,0xeb86d391)

Bước 5: xuất kết quả

- Sau khi sử lý hết L khối 512 bit, đầu ra của lần xử lý thứ L là giá trị băm 128 bit.

1.6.4 Ứng dụng hàm băm.

- Tạo khóa bí mật từ mật khẩu.
- Kiểm tra tính toàn vẹn dữ liệu.
- Mã chứng thực thông điệp sử dụng hàm băm.
- Chữ ký điện tử.

CHƯƠNG 2: PHƯƠNG PHÁP GIẤU TIN DỰA VÀO AUTOMATA 2D–CA

2.1 GIỚI THIỆU

Phương pháp giấu tin 2D-CA cho thông điệp mật đề xuất bởi Biswapati Jana, Debasis Giri, Shymal Kumar Mondal, Pabitra Pal năm 2013 [7]. Kỹ thuật giấu tin dựa vào automata 2D-CA là một phương pháp giấu tin mới và hiệu quả bằng cách nhúng các thông điệp mật vào một ảnh màu cấp xám. Từ thông điệp ban đầu, sau khi đệm thêm bit thông tin có độ dài là bội số của 1024 bit, sau đó được chia nhỏ thông điệp giấu thành các khối con có độ dài 1024 bit. Thông điệp được giấu trên miền LSB của ảnh gốc bằng cách lấy thông điệp giấu XOR với khóa mật có độ dài 1024 bit. Từ các bit giấu, ta áp dụng 2D-CA để cập nhật giá trị điểm ảnh trung tâm của từng khối ma trận ma trận con kích cỡ 3x3 của ảnh gốc bằng quy tắc 341 kiểm tra tính chẵn lẻ bit 1 của khối bit. Trong giải mã, sử dụng 2D-CA quy tắc 341 kiểm tra tính chẵn lẻ bit 1 để lấy thông điệp mật. Để khôi phục thông điệp ban đầu, cần phải kết hợp các khối thông điệp sau đó XOR với khóa mật được chia sẻ giữa bên gửi và bên nhận.

2.2. AUTOMATA HAI CHIỀU

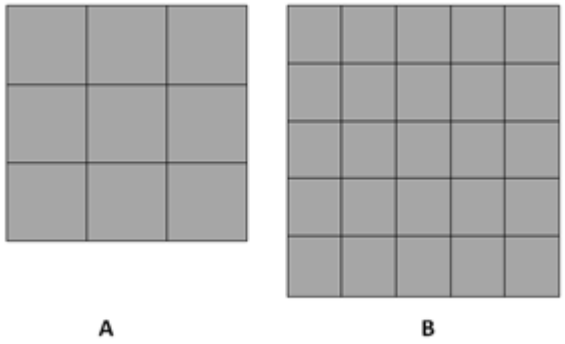
Automata hữu hạn hai chiều (2D-CA) là hệ thống rời rạc tạo bởi một hữu hạn trạng thái mỗi trạng thái là một mảng hai chiều hữu hạn r × s đối tượng (được gọi là ô). Trạng thái của mỗi ô là một phần tử của tập hợp hữu hạn S. Ở đây ta chỉ, xét $S = Z_c$, trong đó $c = 2^b$ là số màu của ảnh. Nghĩa là ảnh đem trắng (black & white image) có giá trị b = 1, ảnh cấp xám có giá trị b = 8 và ảnh màu có giá trị b = 24 [6].

Trạng thái của mỗi ô phụ thuộc vào n biến của hàm dịch chuyển, đó là trạng thái thực của một tập hợp các ô bao gồm cả ô đang xét và các láng giềng. 2D-CA có một số láng giềng, nhưng trong nghiên cứu này chỉ xét láng giềng Moore.

Láng giềng Moore là tập hợp của tất cả các đối tượng (ô) trực giao hoặc đường chéo liền kề với khu vực quan tâm. Với láng giềng Moore phạm vi r được xác định như sau [7].

$$N^M_{(x0,y0)} = \{(x,y) : |x - x0| \leq r, |y - y0| \leq r\}$$

Láng giềng Moore cho các phạm vi r = 1 và r = 2 được minh họa trong hình 2.1. Số ô láng giềng Moore trong phạm vi r là ô vuông được tính theo công thức $(2r + 1)^2$. Nếu r ≥ 2, được xem như là láng giềng Moore mở rộng.



Hình 2.1: A) Láng giềng Moore B) Láng giềng Moore mở rộng

Láng giềng của đối tượng <i, j> được hình thành bởi chín ô gần nhất:

$V_{i,j}$= {<i-1, j-1>, <i-1, j>, <i-1, j + 1>, <i, j-1>,<i, j>, <i, j + 1>, <i + 1, j-1>, <i + 1, j>, <i + 1, j + 1>}.

Có thể minh họa như hình sau:

<i-1, j-1>	<i-1, j>	<i-1, j + 1>
<i, j-1>	<i, j>	<i, j + 1>
<i + 1, j-1>	<i + 1, j>	<i + 1, j + 1>

Trong đó quá trình hàm dịch chuyển f: $(Zc^9) \rightarrow Zc$ là

$$a_{ij}^{(t+1)} = f\,(a^{(t)}_{i-1,j-1}, a^{(t)}_{i-1,j}, a^{(t)}_{i-1,j+1}, a^{(t)}_{i,j-1}, a^{(t)}_{i,j}, a^{(t)}_{i,j+1}, a^{(t)}_{i+1,j-1}, a^{(t)}_{i+1,j}, a^{(t)}_{i+1,j+1},)$$

hoặc tương đương

$$a_{ij}^{(t+1)} = f(V_{ij}^{(t)}),\ 0 \le i \le r - 1,\ 0 \le j \le s - 1$$

trong đó $V^{(t)}_{ij} \subset (Zc)^9$ là trạng thái của các ô<i,j> ở thời điểm t.

$$C^{(t)} = \begin{pmatrix} a_{00}^{(t)} & \cdots & a_{0,s-1}^{(t)} \\ \vdots & \ddots & \vdots \\ a_{r-1,0}^{(t)} & \cdots & a_{r-1,s-1}^{(t)} \end{pmatrix}$$

Ma trận $C^{(t)}$ được gọi là trạng thái tại thời điểm t của 2D-CA và $C^{(0)}$ là trạng thái ban đầu của CA. Ngoài ra, $\{C^{(t)}\}$ $0 \le t \le k$ được gọi là sự phát triển k của 2D-CA và C là tập hợp của tất cả các trạng thái có thể của 2D-CA do đó $| C | = c^{r \cdot s}$.

Khi số ô của 2D-CA là hữu hạn, xét điều kiện để đảm bảo các trạng thái được xác định của CA. Ở đây, điều kiện được thực hiện:

$$a^{(t)}_{ij} = a^{(t)}_{uv} <=> i \equiv u \pmod r, j \equiv v \pmod s$$

Các mô hình chuẩn CA cho rằng trạng thái của các ô ở thời điểm t + 1 phụ thuộc vào trạng thái của một số ô (các vùng lân cận) tại thời điểm t. Tuy nhiên, có thể xét CA mà trạng thái của tất cả các ô lúc t + 1 không chỉ phụ thuộc vào trạng thái của một số ô tại thời điểm t, mà còn phụ thuộc vào các trạng thái (có thể) các nhóm khác nhau của các ô khác ở t - 1, t - 2, vv... đó là MCA (memory cellular automata). Xét một loại hình gọi là LMCA (linear memory cellular automata) tuyến tính thứ k của MCA mà hàm dịch chuyển có dạng sau:

$$a_{ij}^{(t+1)} = \sum_{m=0}^{k-1} fm + 1 \, (V_{ij}^{(t-m)}) \pmod c \quad (1)$$

với $0 \le i \le r - 1$, $0 \le j \le s - 1$, khi đó f_l, $1 \le l \le k$ là hàm dịch chuyển 2D-LCA (linear cellular automata) của k.

Nếu hàm toàn cục định nghĩa 2D-CA với hàm dịch chuyển cục bộ f_k được xác định : $f_k(V_{ij}^{(t-k+1)}) = a_{ij}^{(t-k+1)}$thì các LMCA cho bởi (1) là một MCA 2D đảo ngược, mà CA nghịch đảo là hàm dịch chuyển khác của LMCA:

$$a_{ij}^{(t+1)} = -\sum_{m=0}^{k-2} fk - m - 1 \, (V_{ij}^{(t-m)}) + a_{ij}^{(t-k+1)} \pmod c$$

$0 \le i \le r - 1$, $0 \le j \le s - 1$.

2.3 QUÁ TRÌNH GIẤU TIN TRONG ẢNH DỰA VÀO AUTOMATA 2D-CA

2.3.1 Thuật toán giấu tin.

Quá trình giấu tin automata 2D-CA gồm đầu vào, đầu ra và các bước thực hiện sau:

- Đầu vào:
  - Ảnh sử dụng để giấu tin.
  - Thông điệp giấu.
- Đầu ra:
  - Ảnh đã giấu tin.

Các bước thực hiện:

(Giả sử ảnh đầu vào có kích cỡ ảnh 512 x 512)

- Bước 1: Sau khi sử dụng kỹ thuật nhồi thêm bit thông tin, thông điệp giấu ban đầu M sẽ được chia thành n khối con M_1, M_2,…, M_n, sao cho M = M1 || M2 ||... || M_n, mỗi khối có chiều dài 1024 bit.
- Bước 2: Xét trường hợp, nếu n x 1024 > (k x q)/2 (k, q kích cỡ ảnh cần giấu tin). Giả sử ở đây ta xét ảnh 512 x 512 thì ta sẽ có 512 x 512/2 = 131072 = 128 x 1024. Coi N là số ảnh con trong đó N = [n/128] (với [x] biểu thị hàm tính số nguyên gần x nhất). Xét các ảnh C_k, k từ 1 đến N.
- Bước 3: Chuyển đổi thông tin mật M_i thành H_i với $H_i = M_i$ XOR h (K || i), i từ 1 đến n, còn h là hàm băm 1 chiều SHA đầu ra là 1024 bit. K là khóa bí mật được chia sẻ giữa người gửi và người nhận.
- Bước 4: lặp k từ 1 đến N

  Xét automata 2D-CA khối kích cỡ 3x3, r = 2 và láng giềng Moore (hình 2.1 A). Tổng số ô láng giềng và ô trung tâm bằng 9. Giá trị trung tâm R thay đổi.
- Bước 5: lặp k từ 1 đến N
- Bước 6: lặp row từ 1 đến 512
- Bước 7: lặp col từ 1 đến 512

  B [row] [col] = LSB của ảnh C_k [row] [col] ;

  Kết thúc vòng lặp (Bước 7)

  Kết thúc vòng lặp (Bước 6)
- Bước 8: lặp row từ 1 đến 512
- Bước 9: lặp col từ 1 đến 512
- Bước 10: lặp p từ 1 đến 3
- Bước 11 : lặp q từ 1 đến 3
- Bước 12 : Sử dụng 2D-CA quy tắc 341 kiểm tra tính chẵn lẻ bit 1 của khối B[p] [q] kích cỡ 3x3.

  Kết thúc vòng lặp (Bước 11)

Kết thúc vòng lặp (Bước 10)

- Bước 13 : Cập nhật R sử dụng bảng 3.1 và thay thế đơn vị trung tâm của R.
- Bước 14 : Cập nhật giá trị pixel ảnh gốc sử dụng R đã cập nhật.
- Bước 15 : Di chuyển trạng thái 2D-CA với col = col + 1

Kết thúc vòng lặp (Bước 9)

- Bước 16 : Di chuyển trạng thái 2D-CA với row = row + 2

Kết thúc vòng lặp (Bước 8)

- Bước 17 : Kết thúc vòng lặp (Bước 5)
- Bước 18 : Kết thúc

Bảng 3.1: Quy tắc cập nhật điểm ảnh trung tâm sử dụng 2D-CA

Bit mã hóa ⇒ Parity ⇓	H= 0	H= 1
Lẻ	Thay đổi	Không thay đổi
Chẵn	Không thay đổi	Thay đổi

2.3.2 Ví dụ minh họa quá trình giấu tin

Chuỗi thông điệp cần giấu "xin chao cac ban"

Với chuỗi thông điệp cần giấu trên, chuyển chuỗi thông điệp giấu sang dạng nhị phân ta như sau:

M = 01111000011010010110111000100000011000110110100000110000101101111001000000110001101100001011000110010000001100010011000010110111100

Sau khi đệm bit 0 vào chuỗi nhị phân M và có độ dài 1024 bit

M=01111000011010010110111000100000011000110110100000110000101101111001000000110001101100001011000110010000001100010011000010110111100000000000000000000000000000000000000000000000000000000000000000000000000000…

Cho khóa bí mật K= khoa

Khóa K chuyển sang dạng nhị phân ta được:

MM = 01101011011010000110111101100001

Sử dụng hàm băm SHA512 cho khóa K.

SHA512=D8812E902C9DEF20607592377D4AE94858DF31651077F60E
BDD9E2B7FAF99064F321A2E7923C782542FACCE418715970C7AB55F684EE
E18BCB770DF85A98B5EC

Tiếp tục chuyển khóa K đã được mã hóa kỹ thuật hàm băm SHA512 ta chuyển sang chuỗi nhị phân:

MMi=01000100001110000011100000110001001100100100001010011100100
11000000110010010000110011100101000100010001010100011000110010001
10000001101100011000000110111001101010011100100110010001100110011011
11001101110100010000110100010000010100010100111001001101000011100000
01101010011100001000100010001100011001100110001001101100011010100110
00010011000000110111001101110100011000110110001100000100010101000010
00100010001000100001110010100010100110010010000100011011101000110010
00000101000110001110010011100100110000001101100011010001000110001100
01100110010001100010100000100110010010001010011011100111001001100100
00110011010000110011011100111000001100100011010100110100000110010010
00110010000010100001101000011010001010011010000110001001110000011010
11001100010011010100111001001101110011000001000011001101110100000010
10000100011010100110101010001100011011000111000001101000100010101000
01010100010100110001001110000100001001000011010000100011011100110110
10011000001000100010001100011100000110101010000010011100100111000010
00001000110101010001010100011

HH = M XOR MMi ta được chuỗi bit mã hóa:

HH=00111100010100010101011000010001010100010010110101011000010
11111000100100010000001011000001001110110010100100100010100110101111
10001101100011000000110111001101010011100100110010001100110011011
00110111010001000011010001000001010001010011100100110100001110000010
10100111000010001000100011000110011001100010011011000110101001100
01001100000011011100110111010001100011011000110000010001010100001000
10001000100010000111001010001010011001001000010001101111010001100100
00010100011000111001001110010011000000110110001101000100011000110010
10011001000110001010000010011001001000101001101110011100100110010000
11001101000011001101110011100000110010001101010011010000110010010000
11001000001010000110100001101000101001101000011000100111000001101111

00110001001101010011100100110111001100000100001100110111010000010100001000110101001101010100011000110111000011100000110100010001010100010101000101001100010011100001000010010000110100001000110111001101110011000001000100010001100011100000110101010000010011100100111000010000100010000110101010000011

Ảnh đầu vào được chia thành các khối con có kích cỡ 3x3. Xét khối ma trận con đầu tiên:

$$khoi = \begin{matrix} 108 & 181 & 177 \\ 106 & 180 & 179 \\ 108 & 176 & 173 \end{matrix}$$

Tách lsb của khối đó

$$lsb = \begin{matrix} \underline{0} & 1 & \underline{1} \\ 0 & \underline{0} & 1 \\ \underline{0} & 0 & \underline{1} \end{matrix}$$

Sử dụng 2D-CA quy tắc 341 kiểm tra tính chẵn lẻ bit 1 của khối.

parity = 2

Xét bit thông điệp giấu đầu tiên và khối ma trận con đầu tiên. Sử dụng bảng 3.1 để xác định thay đổi bit trung tâm của từng khối ma trận con 3x3. Có giá trị bit mật là 0, tính chẵn lẻ của bit 1 của khối 2D-CA quy tắc 341 là chẵn. Vì vậy, giữ nguyên giá trị bit trung tâm khối.

$$lsb = \begin{matrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{matrix}$$

Ta được ma trận ảnh sau khi giấu tin:

$$khoi = \begin{matrix} 108 & 171 & 177 \\ 106 & 180 & 179 \\ 108 & 176 & 173 \end{matrix}$$

2.3.3 Thuật toán tách tin

Quá trình tách tin automata 2D-CA gồm đầu vào, đầu ra và các bước thực hiện sau:

- Đầu vào:
  - Ảnh giấu tin.
  - Khóa K.
- Đầu ra:
  - Thông điệp giấu.

Các bước thực hiện:

- Bước 1: lặp k từ 1 đến N

  Xét từng ảnh con đã giấu tin S_k và khóa bí mật được chia sẻ K.

- Bước 2: Xét automata 2D-CA khối kích cỡ 3x3, r = 2 và láng giềng Moore (hình 2.1 A). Tổng số ô láng giềng và ô trung tâm bằng 9.

- Bước 3: lặp row từ 1 đến 512

- Bước 4: lặp col từ 1 đến 512

  B [row] [col] = LSB của ảnh S_k [row] [col];

  Kết thúc vòng lặp (Bước 4)

  Kết thúc vòng lặp (Bước 3)

- Bước 5: lặp row từ 1 đến 512

- Bước 6: lặp col từ 1 đến 512

- Bước 7: lặp p từ 1 đến 3

- Bước 8: lặp q từ 1 đến 3

- Bước 9: Sử dụng 2D-CA quy tắc 341 kiểm tra tính chẵn lẻ bit 1 của khối B[p] [q] kích cỡ 3x3.

  Kết thúc vòng lặp (Bước 8)

  Kết thúc vòng lặp (Bước 7)

- Bước 10: Nếu tính chẵn lẻ của số bit 1 trong H_i bằng 0 thì tách được bit 0 ngược lại tách được bit 1, i từ 1 đến n.

- Bước 11: Di chuyển trạng thái 2D-CA với col = col + 1 trong ảnh giấu tin.

  Kết thúc vòng lặp (Bước 6)

- Bước 12: Di chuyển trạng thái 2D-CA với row = row + 2

  Kết thúc vòng lặp (Bước 5)

- Bước 13: Để nhận được thông điệp $M_i = H_i$ XOR h (K || i) với i từ 1 đến n.

- Bước 14: Để khôi phục lại thông điệp M, người nhận loại bỏ các bit đệm thấp hơn từ các khối cuối cùng từ M_n, số bit đệm là số được lưu trữ trong 11 bit quan trọng nhất của M_n.

- Bước 15: Kết thúc vòng lặp (Bước 1)

- Bước 16: Kết thúc

2.3.4 Ví dụ minh họa quá trình tách tin

Ảnh giấu tin được chia thành các khối con có kích cỡ 3x3.

khoi =
108 171 177
106 180 179
108 176 173

Tách lsb của khối đó

lsb =
0 1 1
0 0 1
0 0 1

Sử dụng 2D -CA quy tắc 341 kiểm tra tính chẵn lẻ bit 1 của khối.

parity = 2

Tính mod(parity,2) khối bằng 0 thì ta lưu trữ bit đó tạo thành chuỗi bit mã hóa tách ở dạng nhị phân.

HH=0011110001010001010101100001000101010001001011010101100001
011111000100100010000001011000001001110110010100100100010100110101
110001101100011000000110111001101010011100100110010001100110011011
100110111010001000011010001000001010001010011100100110101000011100000
101010011100001000100010001100011001100110001001101100011010100110
010011000000110111001101110100011000110110001100000100010101000010
0

1000100010001000011100101000101001100100100001000110111010001100100
0001010001100011100100111001001100000011011000110100010001100011001
1001100100011000101000001001100100100010100110111001110010011001000
1100110100001100110111001110000011001000110101001101000011001001000
1100100000010100001101000011010001010011010000110001001110000001101111
0011000100110101001110010011011100110000010000110011011101000001010
0001000110101001101010100011000110110001110000011010001000101010001
0101000101001100010011100001000010010000110100001000110111001101110
0110000001000100010001100011100000110101010000010011100100111000001 00
0010001101010100010101000011

Chia sẻ khóa mật K = khoa

Khóa chuyển sang dạng nhị phân ta như sau:

MM = 01101011011010000110111101100001

Sử dụng hàm băm SHA512 cho khóa K.

SHA512=D8812E902C9DEF20607592377D4AE94858DF31651077F60E BDD9E2B7FAF99064F321A2E7923C782542FACCE418715970C7AB55F684EE E18BCB770DF85A98B5EC

Tiếp tục chuyển khóa K đã được mã hóa kỹ thuật hàm băm SHA512 ta chuyển sang chuỗi nhị phân:

MMi=0100010000111000000111000000110001001100100100010100111001
0011000000110010010000110011100101000100010001010100011000110010001
1000000110110001100000011011100110101001110010011001000110011001101
1100110111010001000011010001000000101000101001110010011010000111000 0
0110101001110000100010001000110001100110011000100110110001101010011
0001001100000011011100110111010001100011011000110000010001010100001
0010001000100010000111001010001010011001001000010001101110100011001
0000010100011000111001001110010011000000110110001101000100011000110
0110011001000110001010000010011001001000101001101110011100100110010
0011001101000011001101110011100000110010001101010011010000110010010
0011001000000101000011010000110100010100110100001100010011100000 1101
1100110001001101010011100100110111001100000100001100110111010000010
1000010001101010011010101000110001101100011100000110100010001010100
0101010001010011000100111000010000100100001101000010001101110011011

10011000001000100010001100011100000110101010000010011100100111000010000100011010101000101010000011

M = HH XOR MMi ta được chuỗi thông điệp giấu ở dạng nhị phân:

M= 01111000011010010110111000100000011000110110100001100001011011100100000011000110110000101100011001000000110001001100001011011100000000000000000000000000000000000000000000000000000000000000000000000000000…

Chuyển chuỗi nhị phân sang văn bản ta nhận được thông điệp giấu ban đầu: “xin chao cac ban”.

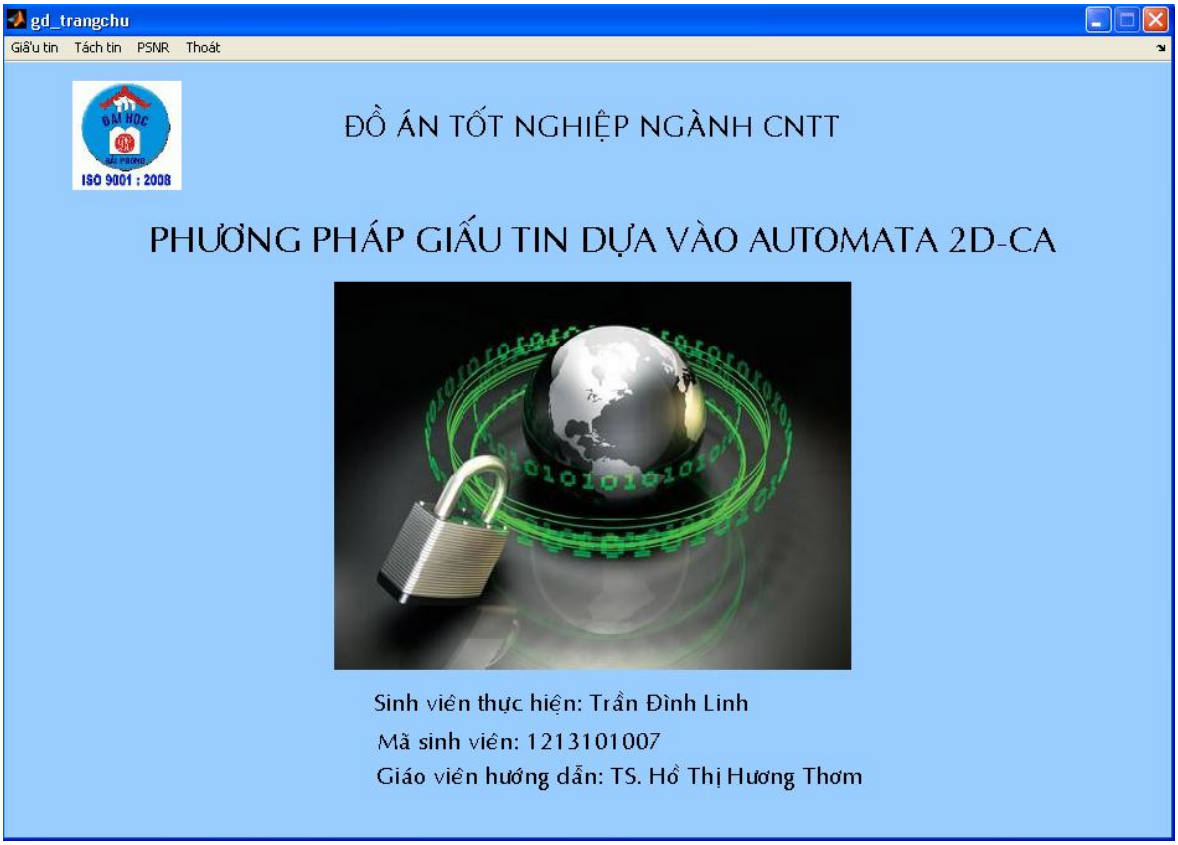
CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM

3.1. MÔI TRƯỜNG CÀI ĐẶT

Ngôn ngữ cài đặt, môi trường soạn thảo và chạy chương trình được thực hiện trên ngôn ngữ lập trình Matlab 7.8.

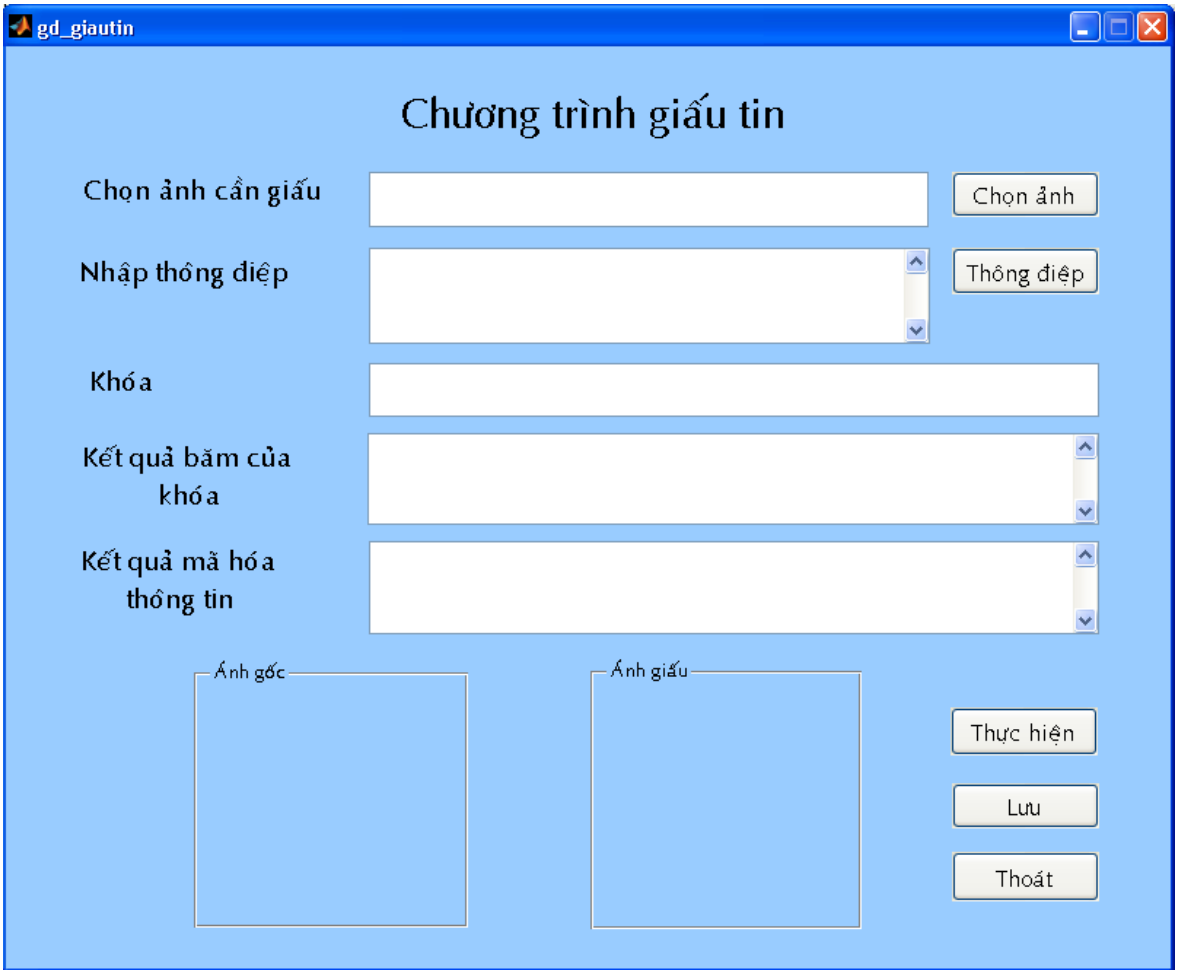
Hệ điều hành Window XP và môi trường NetFarme Work 2.0.

3.2. GIAO DIỆN CHƯƠNG TRÌNH



Hình 3.1. Giao diện chính của chương trình

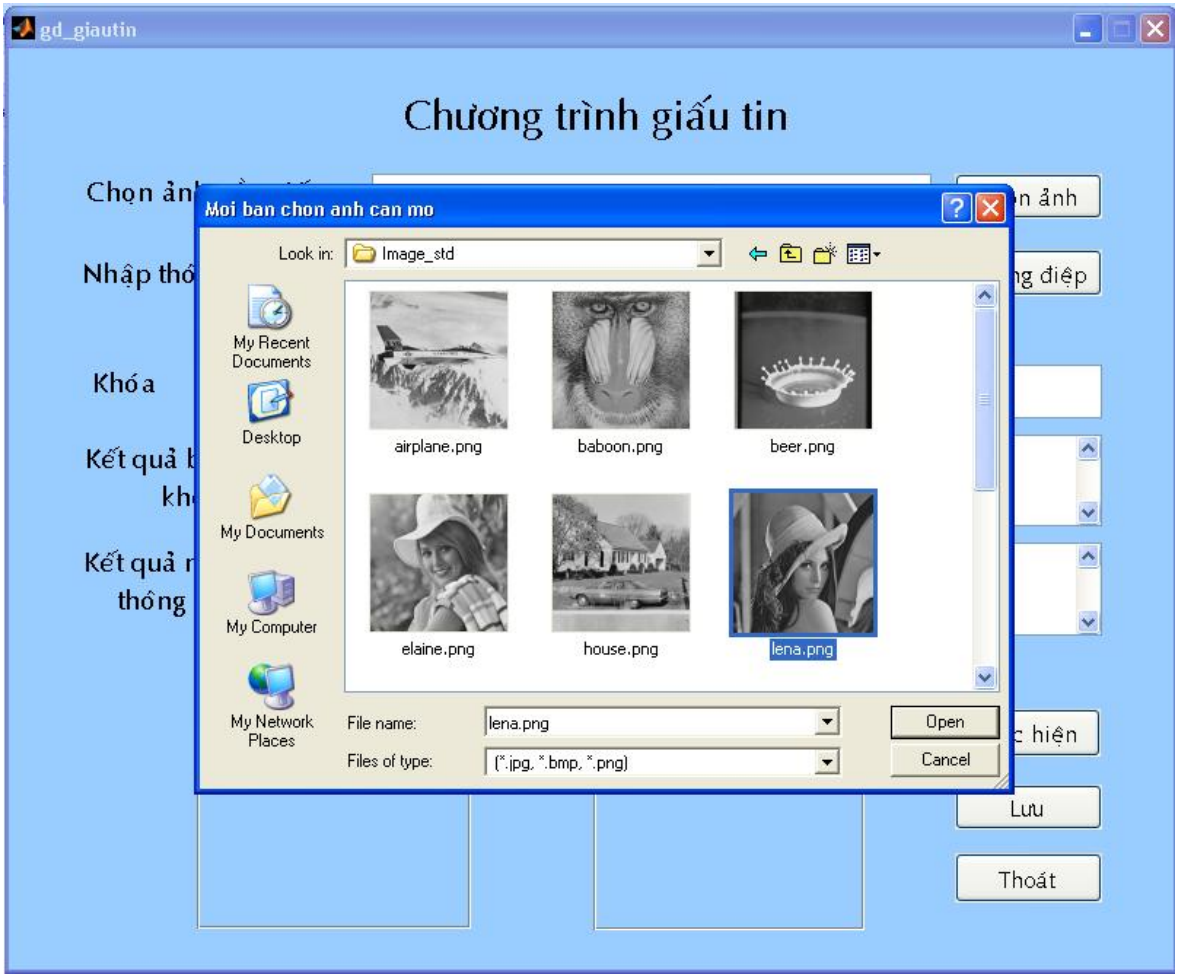
Từ menu chọn "Giấu tin" sẽ gọi đến giao diện giấu tin hình 3.2.



Hình 3.2. Giao diện giấu tin

Đây là giao diện để giấu thông tin vào ảnh sử dụng kỹ thuật giấu tin automata 2D-CA.

Để nhập ảnh vào ta chọn nút "Chọn ảnh" trong giao diện giấu tin, một hộp thoại sẽ được mở ra để ta chọn ảnh đưa vào giấu tin (chương trình chỉ sử dụng ảnh cấp xám).

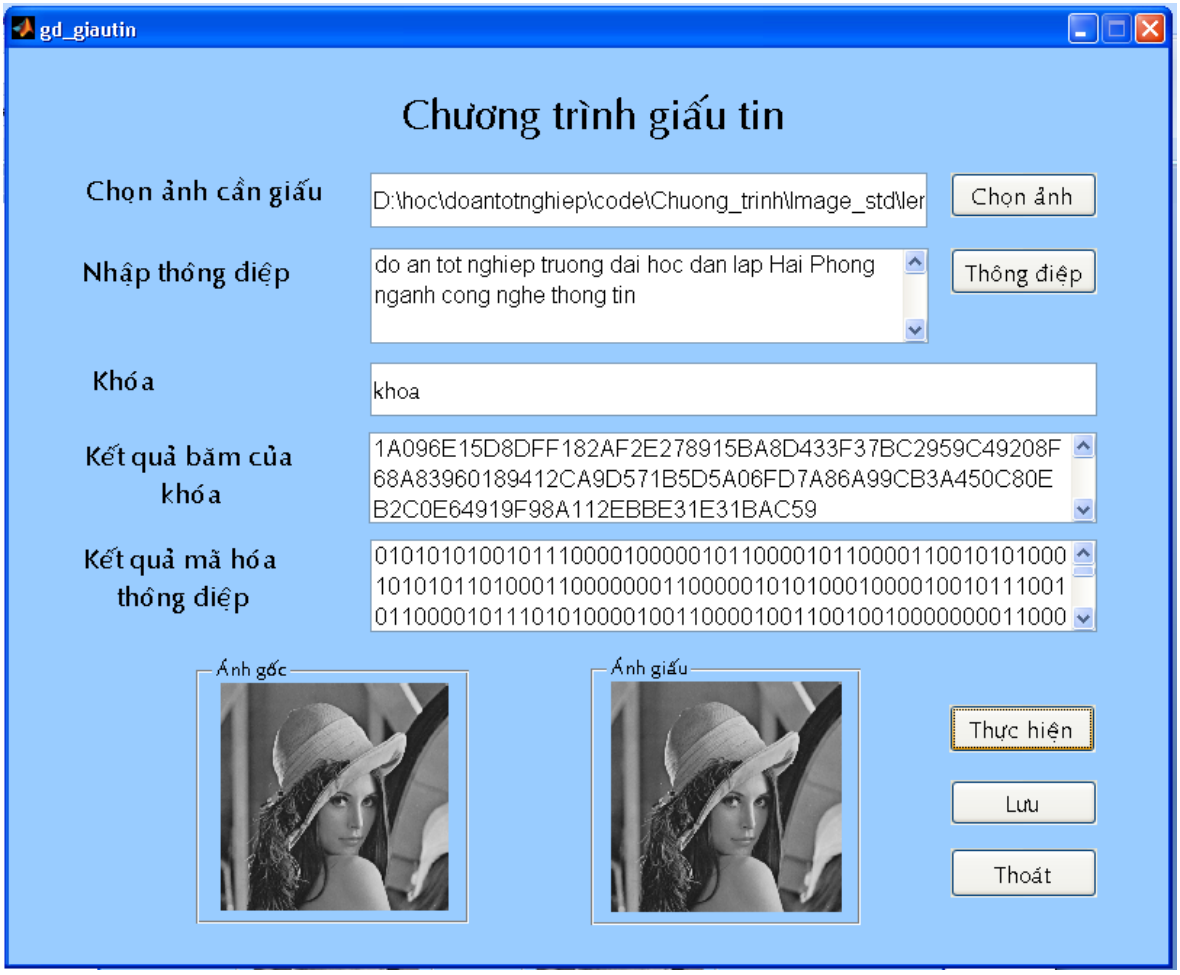


Hình 3.3. Chọn ảnh để giấu tin.

Tiếp theo, ta sẽ nhập thông điệp cần giấu vào ảnh bằng cách nhập trực tiếp thông tin giấu vào ô text "Nhập thông điệp" hoặc chọn nút "Thông điệp" chọn tệp tin thông điệp, nhập khóa cho chương trình giấu tin như hình 3.4.

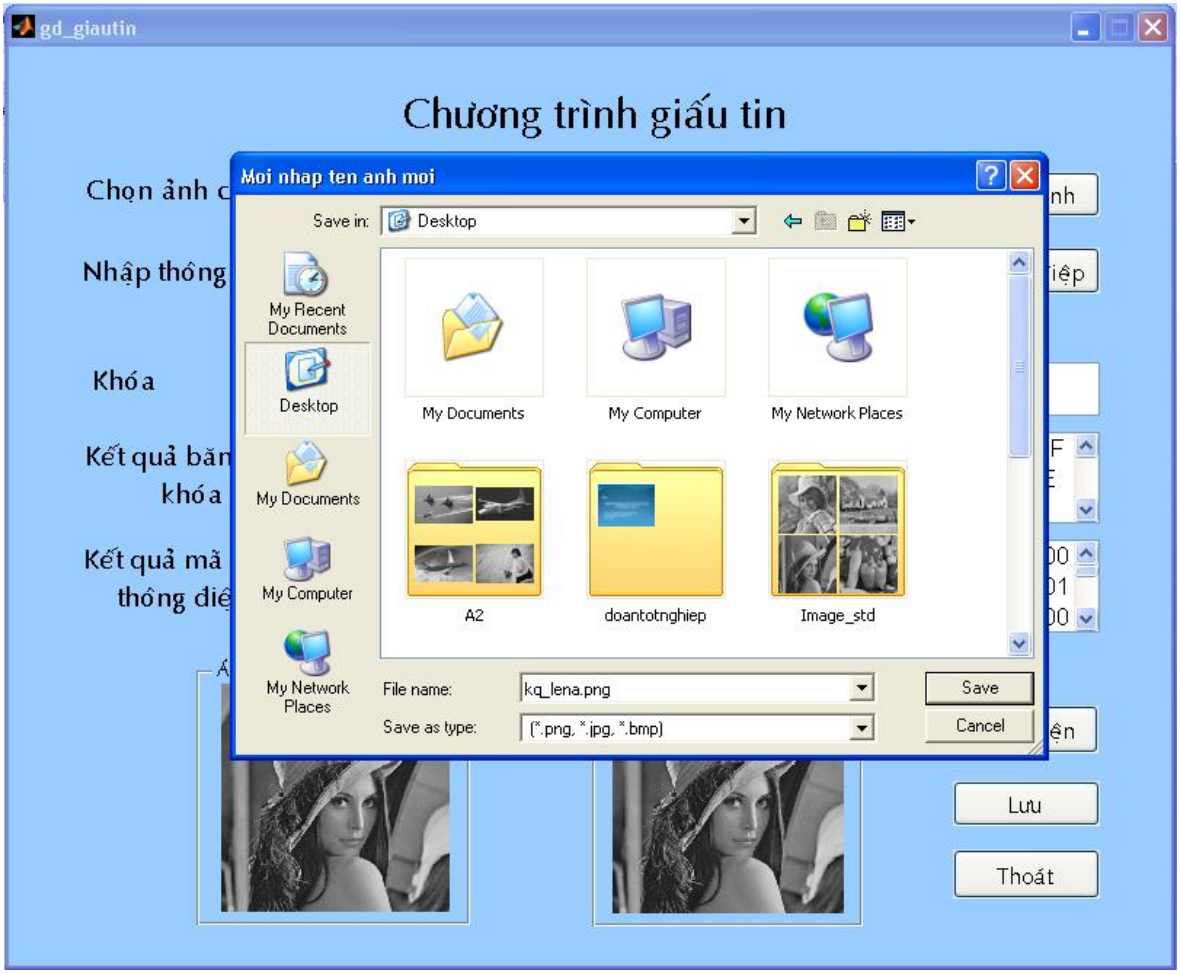
Hình 3.4. Nhập thông điệp giấu và nhập khóa.

Tiếp theo ta thực hiện quá trình giấu tin bằng phương pháp Automata 2D-CA bằng cách chọn nút "Thực hiện", kết quả hiện ra giá trị băm khóa và giá trị mã hóa thông điệp giấu vào ảnh như hình 3.5



Hình 3.5. Thực hiện giấu tin trong ảnh.

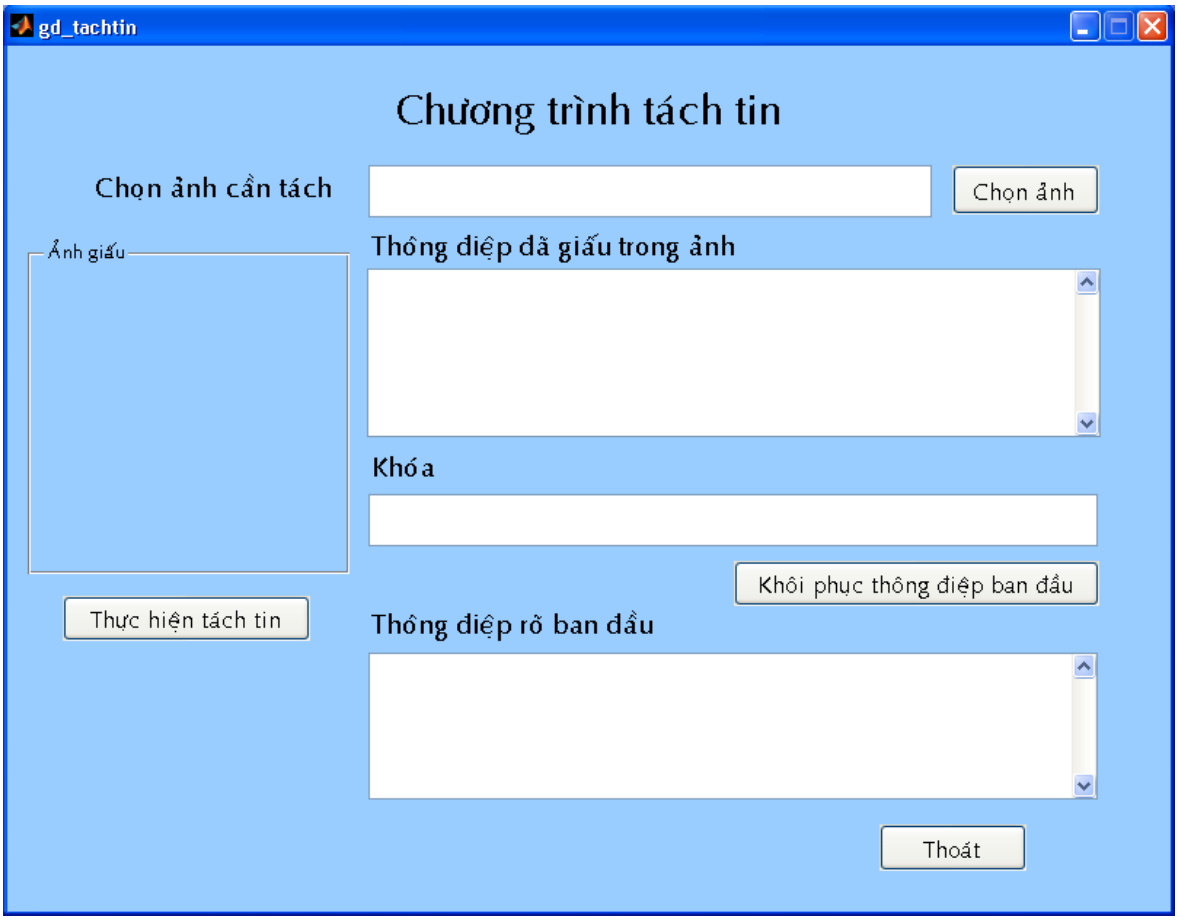
Tiếp theo ta sẽ nhập tên và chọn nơi lưu ảnh kết quả bằng cách kích vào nút "Lưu" như hình 3.6.



Hình 3.6. Chọn lưu ảnh kết quả.

Khi lưu ảnh giấu xong ta chọn nút "Thoát" để kết thúc chương trình giấu tin.

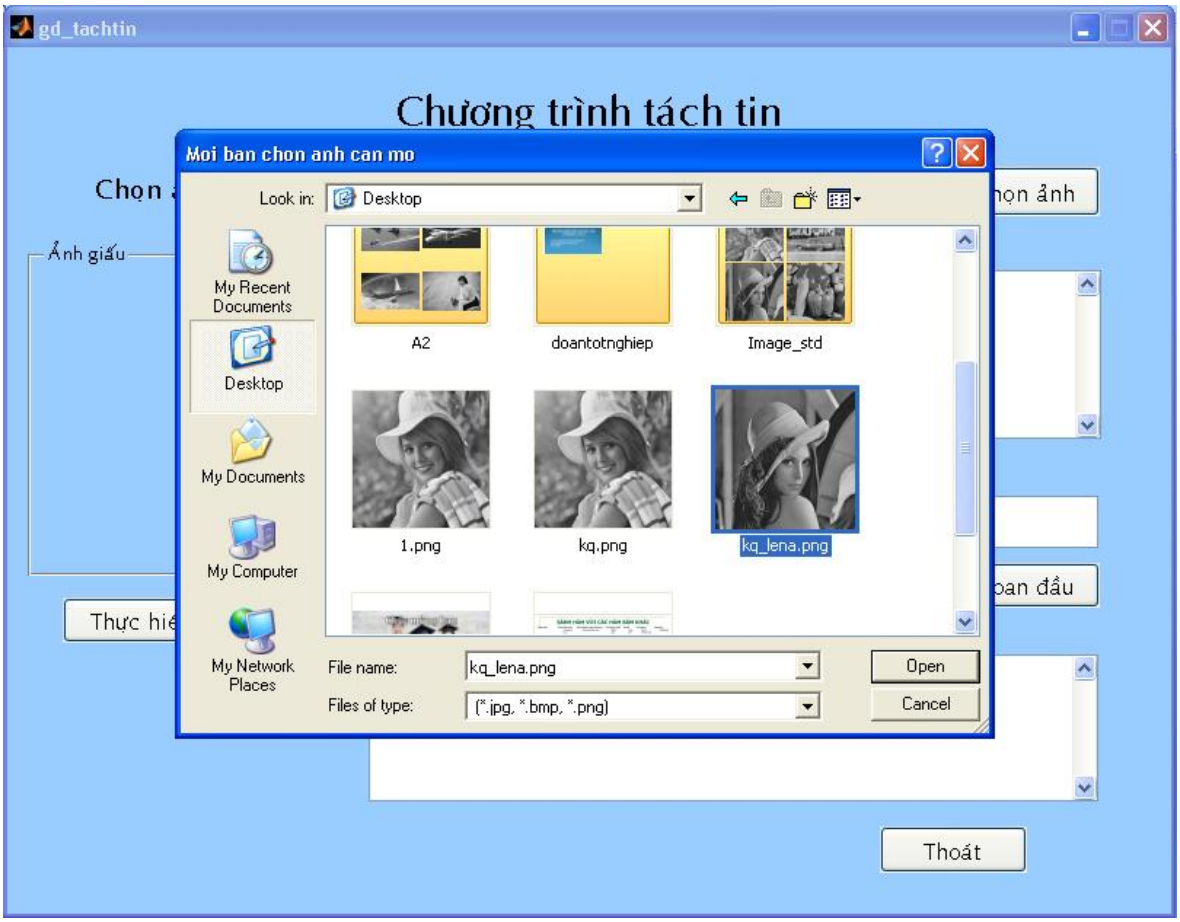
Từ menu chọn "Tách tin" trên giao diện chính gọi ra giao diện tách tin hình 3.7.



Hình 3.7. Giao diện tách tin

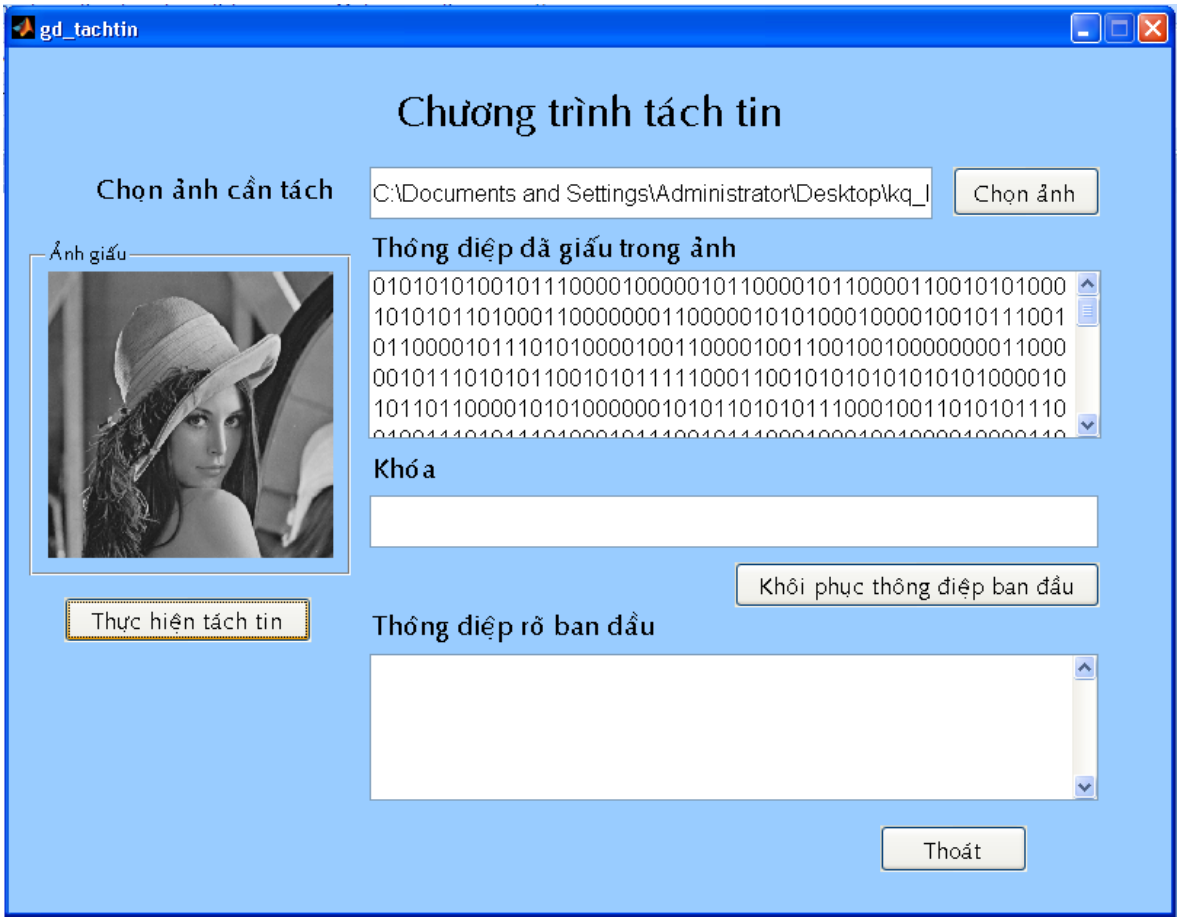
Đây là giao diện sẽ lấy ảnh đã giấu thông tin để xử lý tách tin lấy ra dữ liệu đã giấu trong ảnh.

Thực hiện mở ảnh giấu tin để tách thông tin đã giấu bằng cách chọn nút "Chọn ảnh" như hình 3.8.



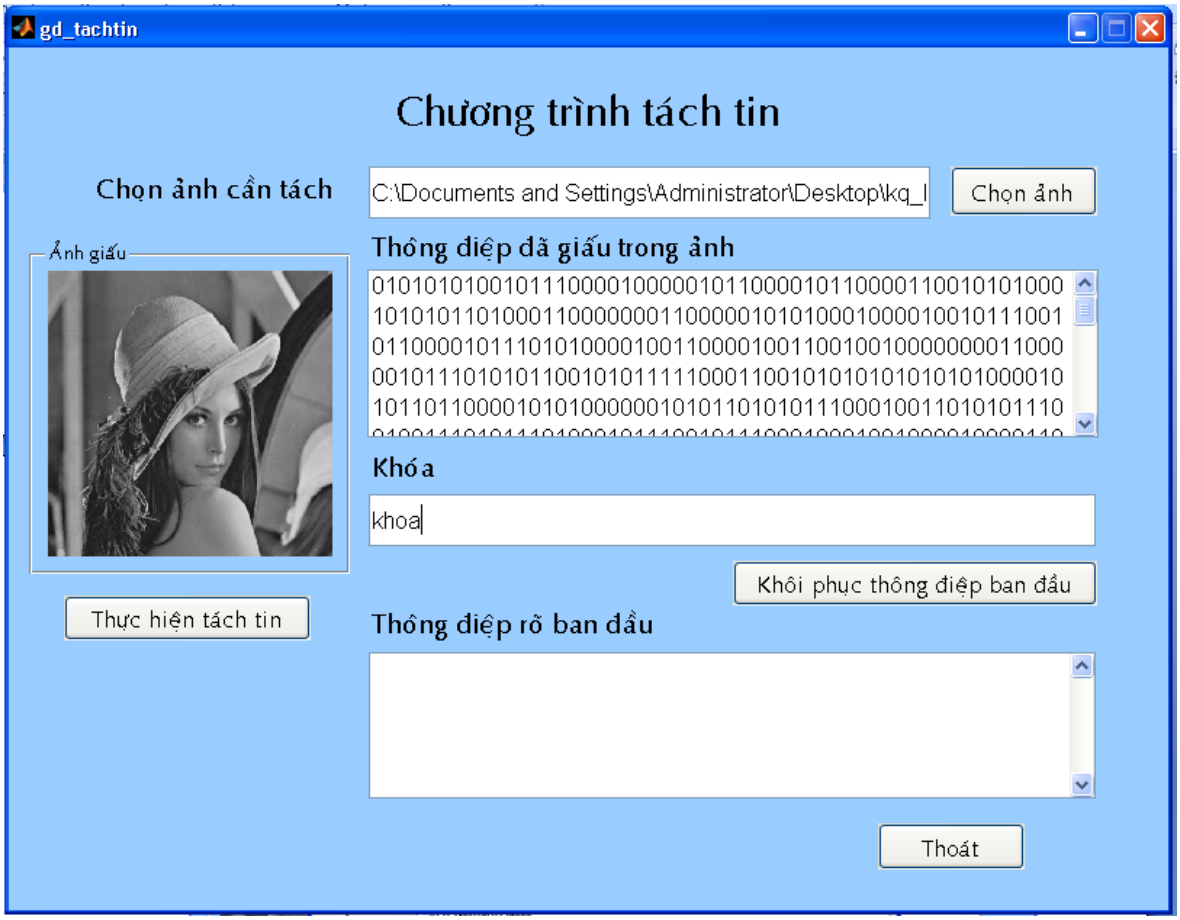
Hình 3.8. Chọn ảnh để tách tin.

Sau khi nhập ảnh đã giấu tin, ta thực hiện tách chuỗi thông điệp mã hóa bằng cách chọn nút "Thực hiện tách tin" để tách. Thông tin lấy được là thông điệp mã hóa đã được giấu trong ảnh dưới dạng nhị phân như hình 3.9.



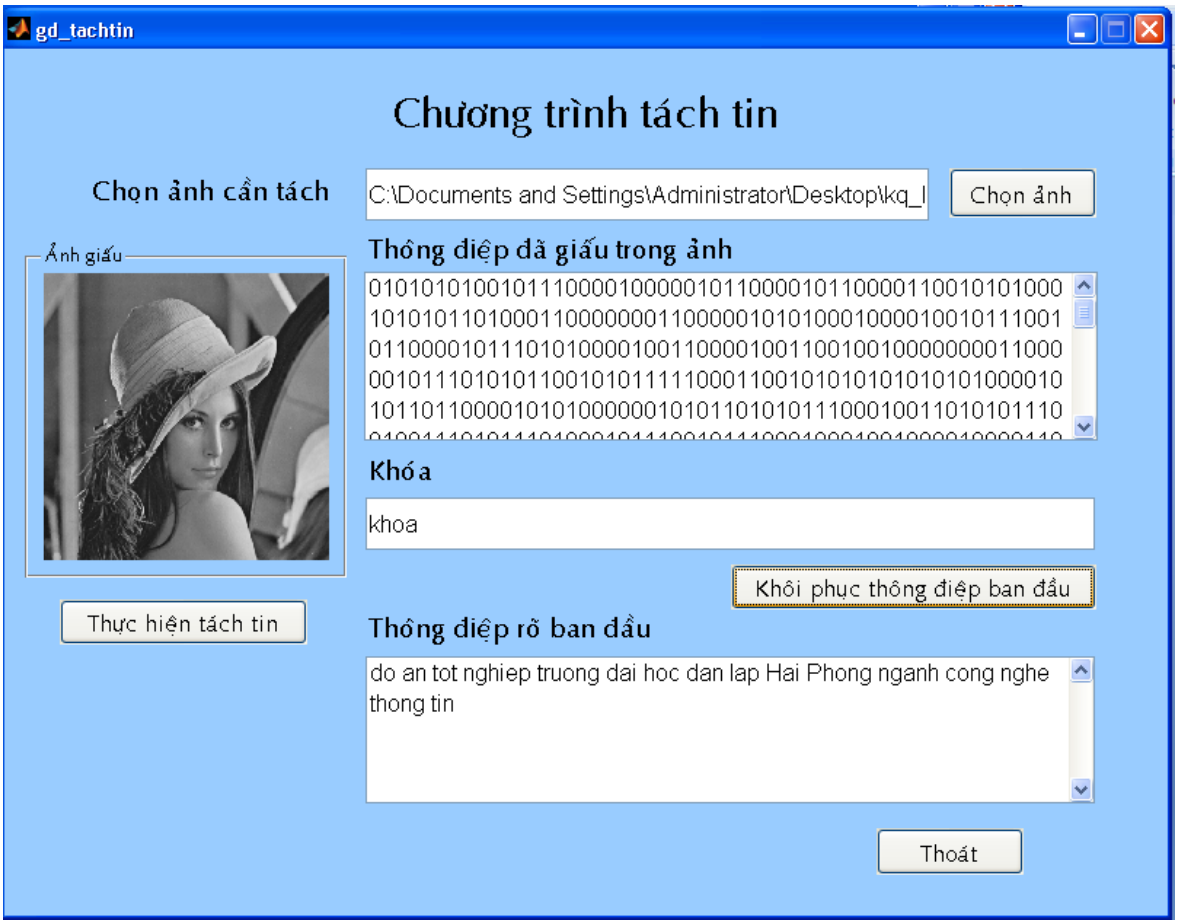
Hình 3.9. Tách thông điệp giấu.

Nhập Khóa hình 3.10.



Hình 3.10. Nhập khóa.

Ta chọn nút "Khôi phục thông điệp ban đầu" để lấy thông điệp giấu và đưa ra kết quả như hình 3.11



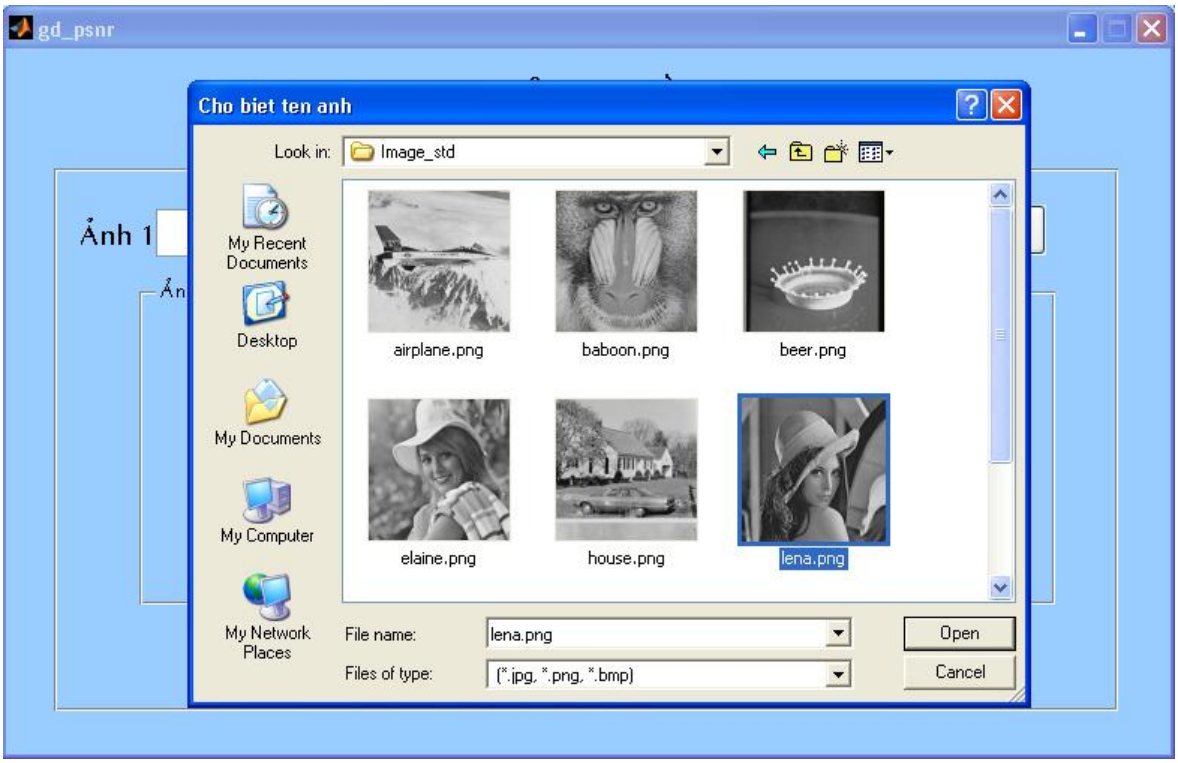
Hình 3.11. Khôi phục lại thông điệp ban đầu.

Chọn nút "Thoát" để kết thúc chương trình tách tin.

Từ menu chọn "PSNR" trên giao diện chính gọi ra giao diện đánh giá ảnh PSNR hình 3.12.

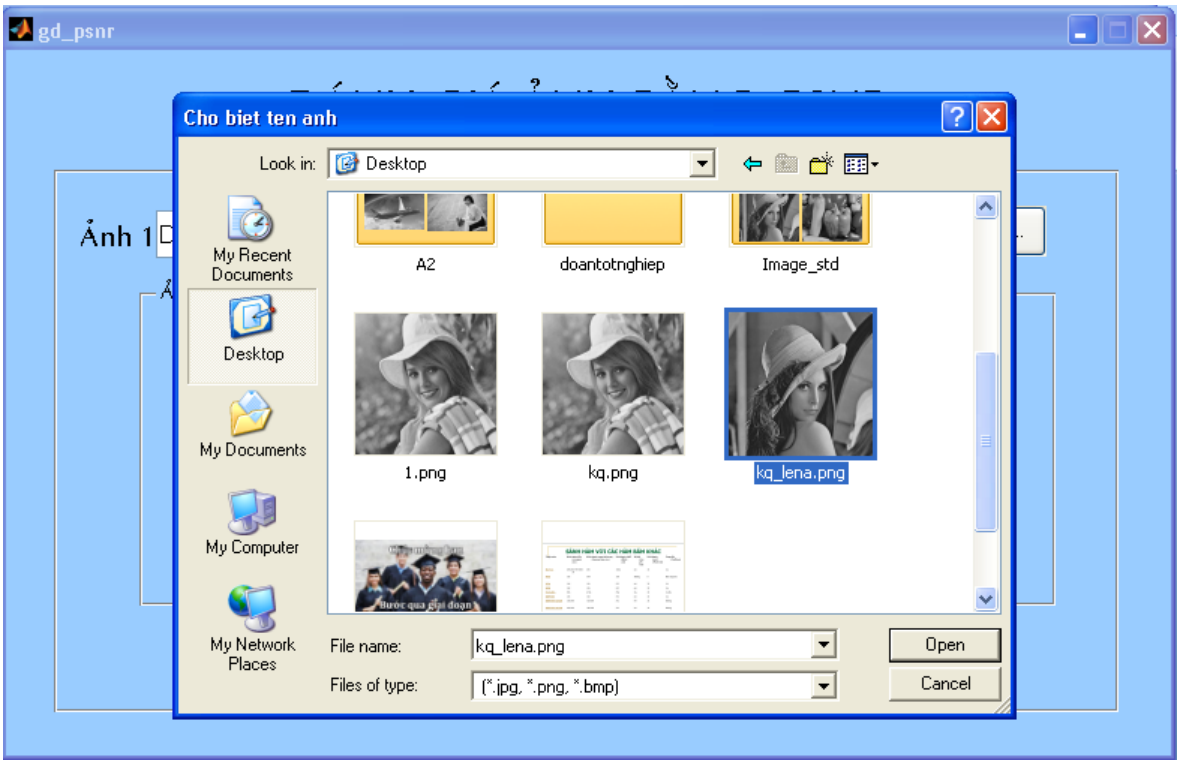
Hình 3.12. Giao diện đánh giá ảnh PSNR.

Chọn nút "…" trong mục "Ảnh 1" để chọn ảnh trước khi giấu tin như hình 3.13.



Hình 3.13. Mở ảnh gốc.

Chọn nút "…" trong mục "Ảnh 2" để chọn ảnh ảnh đã giấu tin như hình 3.14.

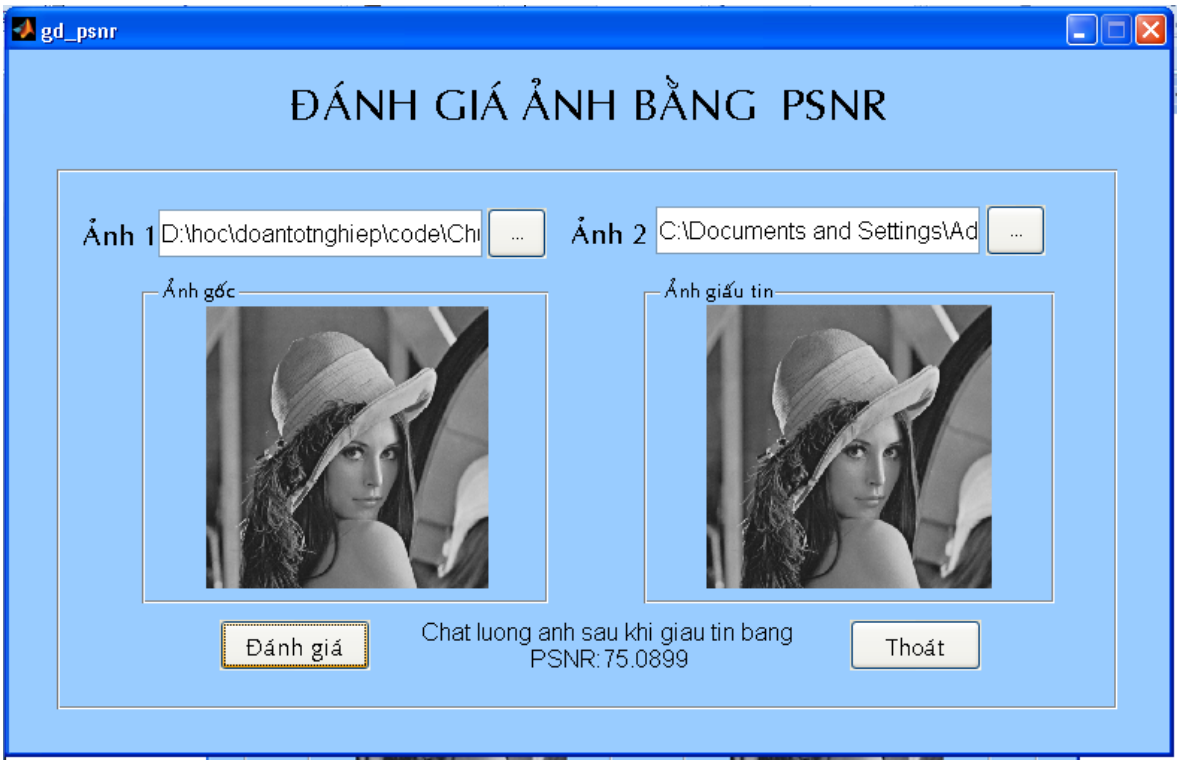


Hình 3.14. Mở ảnh giấu thông tin.

Sau khi đã chọn 2 ảnh để đánh giá với nhau, chọn nút "Đánh giá" để thực hiện quá trình đánh giá như hình 3.15.

Kết quả sẽ cho ta giá trị đánh giá ảnh nếu :

- Giá trị là 100 db thì 2 ảnh là 1 không có thay đổi.
- Giá trị lớn hơn 35 db nhỏ hơn 100 db thì ảnh có thay đổi nhưng ở mức chấp nhận được.
- Giá trị nhỏ hơn 35 db thì ảnh biến dạng mắt thường có thể nhận thấy.
- Giá trị nhỏ hơn 20 db thì ảnh biến dạng không thể chấp nhận được.



Hình 3.15. Giao diện sau khi đánh giá chất lượng ảnh.

Chọn nút "Thoát" để kết thúc chương trình đánh giá ảnh bằng PSNR.

3.3. KẾT QUẢ THỬ NGHIỆM CHƯƠNG TRÌNH VÀ NHẬN XÉT

3.3.1. Kết quả thử nghiệm chương trình

Thực nghiệm này sẽ đưa ra độ đánh giá PSNR với ảnh trước và sau khi giấu tin sử dụng kỹ thuật giấu tin Automata 2D-CA. Tập ảnh thử nghiệm là ảnh định dạng *.png gồm tập A1 là 9 ảnh cấp xám chuẩn định dạng png có kích thước 512x512. Và tập ảnh A2 là 20 ảnh tải về trên mạng có kích thước khác nhau được đặt tên từ anh1 tới anh20 được chuyển thành cảnh cấp xám thông qua phần mềm matlab 7.8

Tập ảnh xám chuẩn A1 trước khi giấu tin hình 4.16.

Hình 3.16. Tập ảnh xám chuẩn A1 trước khi giấu tin.

Tập ảnh xám ngẫu nhiên A2 trước khi giấu tin hình 3.17.

Hình 3.17. Tập ảnh xám ngẫu nhiên A2 trước khi giấu tin.

Chuỗi thông điệp giấu có nội dung: "do an tot nghiep truong dai hoc dan lap Hai Phong nganh cong nghe thong tin".

Tập ảnh xám chuẩn A1 sau khi giấu tin hình 3.18.

Hình 3.18. Tập ảnh xám chuẩn A1 sau khi giấu tin.

Tập ảnh xám ngẫu nhiên A2 sau khi giấu tin hình 4.19.

Hình 3.19. Tập ảnh xám ngẫu nhiên A2 sau khi giấu tin.

Bảng 3.1. Kết quả đánh giá PSNR sau khi giấu tin và khả năng giấu tin với tập ảnh A1 và A2.

Ảnh gốc	Đánh giá PSNR (dB)	Khả năng giấu bit
airplane.png	75.4213	29127
baboon.png	75.4984	29127
beer.png	75.3526	29127
elaine.png	75.0734	29127
house.png	75.1792	29127
lena.png	75.0899	29127
peppers.png	75.3966	29127
sailboat.png	75.4498	29127
tiffany.png	75.2235	29127
anh1.png	72.3995	14222
anh2.png	72.7355	16200

anh3png	87.2034	455111
anh4.png	67.967	5585
anh5.png	68.0661	5583
anh6.png	76.7362	41666
anh7.png	79.8447	87381
anh8.png	79.9947	87381
anh9.png	73.1497	18350
anh10.png	73.8022	20833
anh11.png	68.0506	5607
anh12.png	68.0595	5583
anh13.png	67.9369	5589
anh14.png	67.8449	5597
anh15.png	67.0928	5598
anh16.png	80.9583	87381
anh17.png	75.031	26666
anh18.png	74.9727	28942
anh19.png	74.7936	46433
anh20.png	73.1624	17944

3.3.2 Nhận xét

Đây là kỹ thuật giấu tin dù lượng thông tin cần giấu ít nhưng khi giấu ta giấu trên toàn ảnh gốc (vì sử dụng lặp lại trên toàn ảnh). Đây là kỹ thuật giấu có thể coi là thủy vân ảnh nhằm mục đích toàn vẹn dữ liệu ảnh hoặc bảo vệ bản quyền.

Với kết quả thử nghiệm thu được, nếu quan sát bằng mắt thường thì khó có thể phân biệt được đâu là ảnh đã giấu tin và chưa giấu tin. Giá trị PSNR trung bình đạt được là khá cao so với thang đo giá trị trung bình PSNR (của MOS) là 37dB, giá trị PSNR trung bình của tập ảnh đạt 74.05126dB. Giá trị thấp nhất của tập ảnh là 67.0928dB, giá trị PSNR cao nhất là 87.2034dB. Khả năng giấu bit của ảnh phụ thuộc vào kích cỡ ảnh giấu tin.

Qua thử nghiệm em nhận thấy kỹ thuật giấu tin dựa vào automata 2D-CA đạt một số kết quả sau.

Ưu điểm:

- Độ an toàn bảo mật thông tin cao.

- Không thay đổi ảnh hưởng đến thông tin giấu.
- Chất lượng ảnh giấu tin cao.

Nhược điểm:

- Thời gian xử lý giấu tin phụ thuộc lớn vào dữ liệu đầu vào như kích thước ảnh gốc, thông điệp giấu lớn hay nhỏ.

Thuật toán giấu tin 2D-CA có độ bảo mật rất cao, 2 lần bảo mật: giấu trong ảnh (bảo mật vô hình), bảo mật thông điệp giấu nếu tin bị tách ra thì người ta áp dụng mã hóa thông điệp bằng khóa K và áp dụng hàm băm SHA kết hợp với hàm XOR.

KẾT LUẬN

Đồ án của em đã thực hiện những nhiệm vụ sau:

- Tổng quan về giấu tin, môi trường giấu tin, ứng dụng của giấu tin, đánh giá chất lượng ảnh bằng PSNR, hàm băm SHA.
- Tìm hiểu automata 2 chiều 2D-CA.
- Tìm hiểu phương pháp giấu tin áp dụng automata 2 chiều 2D-CA.
- Xây dựng được chương trình giấu tin trong ảnh.

Với kỹ thuật giấu tin dựa vào automata 2D-CA thì tính vô hình và bền vững của thông tin sau khi giấu được đảm bảo. Về mặt lý thuyết thì sau khi đã có lượng thông tin được giấu vào trong ảnh gốc, chất lượng ảnh sẽ khác với ảnh gốc ban đầu. Tuy nhiên sau khi thực hiện kỹ thuật giấu tin, quan sát bằng mắt thường thì khó có thể phân biệt đâu là ảnh gốc, đâu là ảnh giấu tin. Dùng phương pháp đánh giá PSNR để đánh giá chất lượng ảnh trước và sau khi giấu tin kết quả PSNR đạt được là khá cao, điều đó cho thấy sự biến dạng của ảnh hầu như không có. Như vậy kỹ thuật giấu tin đã cho những kết quả tốt.

Sau một thời gian tìm hiểu và nghiên cứu dưới sự hướng dẫn tận tình của cô giáo hướng dẫn TS Hồ Thị Hương Thơm em đã hoàn thành báo cáo và chương trình thử nghiệm.Tuy nhiên kỹ thuật giấu tin dựa vào automata 2D-CA còn mới mẻ, với thời gian thực hiện đề tài có hạn cộng với khả năng, kinh nghiệm còn hạn chế nên báo cáo của em còn gặp nhiều thiếu sót. Vì vậy em rất mong nhận được sự đóng góp ý kiến của các thầy giáo cô giáo để bài báo cáo tốt nghiệp của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1]. Luận án tiến sĩ Hồ thị Hương Thơm năm 2012.

[2]. TS Hồ Thị Hương Thơm, bài giảng "AN TOÀN BẢO MẬT THÔNG TIN" chương 1+2.

[3]. Vũ Thị Ngọc- 111247, "Nghiên cứu một giải pháp giấu tin trong ảnh", đồ án tốt nghiệp ngành công nghệ thông tin 2011.

[4]. Đào Đình Hùng- Ct1201, "Kỹ thuật giấu tin trong ảnh sử dụng kết hợp mãhóa aes và giấu tin", đồ án tốt nghiệp ngành công nghệ thông tin 2012.

[5]. Vũ Thùy Dung- CT1101, "Kỹ thuật giấu tin trong ảnh SES", đồ án tốt nghiệp ngành công nghệ thông tin 2011.

Tài liệu tiếng Anh

[6]. G. lvarez Maranón, L.H. Encinas, A.M. del Rey, Sharing secret color imagesusing cellular automata with memory, CoRR 0312034 (2003).

[7]. Biswapati Jana, Debasis Giri, Shymal Kumar Mondal, Pabitra Pal, IMAGE STEGANOGRAPHY BASEDON CELLULAR AUTOMATA (2013).

Tài liệu web

[8]. http://vtct.wordpress.com/2013/06/19/mot-vai-khai-niem-co-ban-ve-anh-so-digital-image/

[9]. http://luanvan.co/luan-van/luan-van-giau-tin-trong-anh-va-ung-dung-trong-an-toan-bao-mat-thong-tin-39371/