MỤC LỤC

LỜI CẢM ƠN

Trước hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới cô giáo hướng dẫn Thạc sỹ Hồ Thị Hương Thơm đã tận tình giúp đỡ em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành báo cáo tốt nghiệp.

Em xin chân thành cảm ơn các thầy cô trong bộ môn tin còn như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành báo cáo.

Xin chân thành cảm ơn các bạn trong và ngoài lớp đã động viên và tạo điều kiện thuận lợi cho em trong quá trình làm báo cáo tốt nghiệp.

Cuối cùng em xin bày tỏ lòng biết ơn sâu sắc tới những người thân trong gia đình đã dành cho em sự quan tâm hết mực và động viên em.

Hải phòng ngày tháng 7 năm 2009

Sinh viên

Mạc Như Hiển

MỞ ĐẦU

Mọi người chắc không ai là không biết về sự kiện 11/9, hai toà cao ốc trung tâm thương mại thế giới của Mĩ đã bị khủng bố, khiến biết bao người thiệt mạng, đó là một ngày kinh hoàng đối với nước Mĩ nói riêng và thế giới nói chung. Vậy làm sao bọn khủng bố lại có thể "qua mặt" cơ quan tình báo CIA của Mĩ để thực hiện được vụ khủng bố một cách dễ dàng như vậy ? Mãi gần đây mới có câu trả lời, đó là vì chúng đã áp dụng công nghệ Data hiding, ở đây tạm dịch là Công Nghệ Giấu Tin, với công nghệ này chúng có thể truyền tin cho đồng bọn trên các phương tiện đại chúng mà không bị phát hiện, nhắm qua mặt cơ quan tình báo.

Cuộc cách mạng thông tin kỹ thuật số đã đem lại những thay đổi sâu sắc trong xã hội và trong cuộc sống của chúng ta. Những thuận lợi mà thông tin kỹ thuật số mang lại cũng sinh ra những thách thức và cơ hội mới cho quá trình đổi mới. Mạng Internet toàn cầu đã biến thành một xã hội ảo nơi diễn ra quá trình trao đổi thông tin trong mọi lĩnh vực chính trị, quân sự, quốc phòng, kinh tế, thương mại…Và chính trong môi trường mở và tiện nghi như thế xuất hiện những vấn nạn, tiêu cực đang rất cần đến các giải pháp hữu hiệu cho vấn đề an toàn thông tin như: nạn xuyên tạc thông tin, truy nhập thông tin trái phép, v.v... Đi tìm giải pháp cho những vấn đề này không chỉ giúp ta hiểu thêm về công nghệ phức tạp đang phát triển rất nhanh này mà còn đưa ra những cơ hội kinh tế mới cần khám phá.

Giải pháp nào cho những vấn đề trên ? Trong một quá trình phát triển lâu dài, nhiều phương pháp bảo vệ thông tin đã được đưa ra trong đó giải pháp dùng mật mã học là giải pháp được ứng dụng rộng rãi nhất . Các hệ mã mật đã được phát triển nhanh chóng và được ứng dụng rất phổ biến cho đến tận ngày nay. Thông tin ban đầu sẽ được mã hoá thành các kĩ hiệu vô nghĩa, sau đó sẽ

được lấy lại thông qua việc giải mã nhờ khoá của hệ mã. Đã có rất nhiều những hệ mã phức tạp được sử dụng như DES, RSA, NAPSACK...và phương pháp này đã được chứng minh thực tế là rất hiệu quả và được ứng dụng phổ biến. Hơn nữa sự phát triển của các phương tiện kỹ thuật số đã làm cho việc lưu trữ, sửa đổi và sao chép dữ liệu ngày càng đơn giản, từ đó việc bảo vệ bản quyền và chống xâm phạm trái phép các dữ liệu đa phương tiện (âm thanh, hình ảnh, tài liệu) cũng gặp nhiều khó khăn. Một công nghệ mới được ra đời đã phần nào giải quyết được các khó khăn trên là giấu thông tin trong các nguồn đa phương tiện như các nguồn âm thanh, hình ảnh, ảnh tĩnh…Xét theo khía cạnh tổng quát thì giấu thông tin cũng là một hệ mã mật nhằm đảm bảo tính an toàn thông tin, những phương pháp này ưu điểm ở chỗ giảm được khả năng phát hiện ra sự tồn tại của thông tin trong các nguồn mạng. Không giống như mã hoá thông tin là để chống sự truy cập và sửa chữa một cách trái phép thông tin, mục tiêu của giấu thông tin là làm cho thông tin trở nên vô hình hay không nghe thấy được đối tượng. Điều này sẽ đánh lừa được sự phát hiện của các tin tặc và do đó sẽ làm giảm khả năng bị giải mã.

Giấu thông tin là một kỹ thuật còn tương đối mới và đang phát triển rất nhanh thu hút được sự quan tâm của cả giới khoa học và giới công nghiệp nhưng cũng còn rất nhiều thách thức và trở ngại.

Bản báo cáo này em xin trình bày về giấu thông tin trong các nguồn đa phương tiện nói chung và ở đây cụ thể là giấu thông tin trong ảnh GIF. Đồng thời trình bày một số kỹ thuật giấu thông tin trong ảnh GIF.

CHƯƠNG 1: TỔNG QUAN VỀ KỸ THUẬT GIẤU TIN TRONG ẢNH

1.1 Định nghĩa giấu tin và mục đích của việc giấu tin

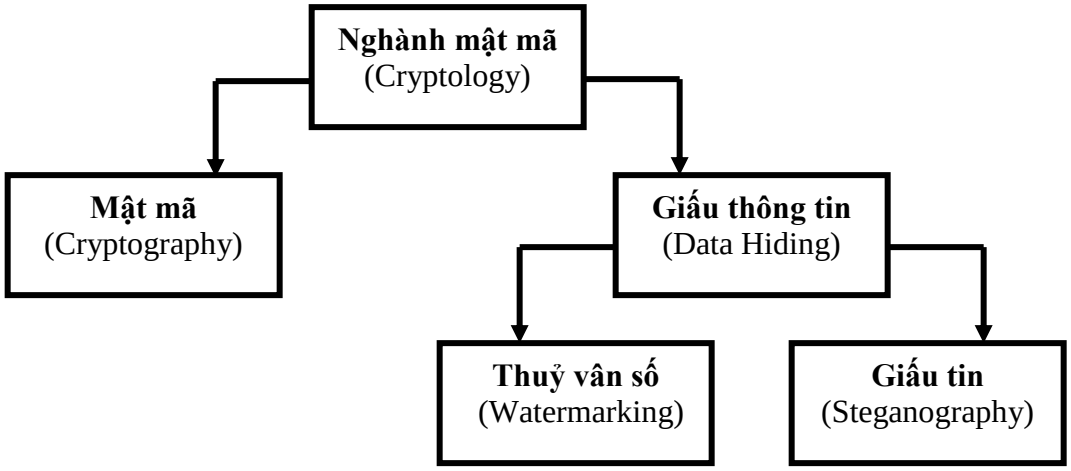
- Định nghĩa giấu tin: Đây là kỹ thuật nhúng một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác.

- Mục đích của việc giấu tin là đảm bảo an toàn và bảo mật thông tin. Có 2 khía cạnh cần được quan tâm đó là:

+ Bảo mật cho dữ liệu được đem giấu. Khía cạnh này tập trung vào các kỹ thuật giấu tin mật tức là giấu tin sao cho thông tin giấu được nhiều và người khác khó phát hiện ra thông tin có được giấu trong đó hay không. VD: Trao đổi thông tin mật.

+ Bảo mật cho chính đối tượng được đem giấu thông tin còn gọi là thủy vân số. Thủy vân số đánh dấu vào chính đối tượng nhằm khẳng định bản quyền sở hữu hay phát hiện xuyên tạc thông tin.

1.2 Phân loại các kỹ thuật giấu tin



Hình 1:Mô hình ngành mật mã

- Có thể chia kỹ thuật giấu dữ liệu ra làm 2 hướng lớn, đó là: watermarking và steganography.

+ Watermaking quan tâm tới việc giấu các mẩu tin ngắn nhưng đòi hỏi độ bền vững cao của các thông tin cần giấu đối với các biến đổi thông thường của các tệp dữ liệu môi trường.

+ Steganography quan tâm tới ứng dụng che dấu các bản tin đòi hỏi bảo mật và dung lượng càng lớn càng tốt.

- Việc phân loại cú thể tiếp tục theo từng chỉ tiêu khác nhau.

Ví dụ:

- Theo ảnh hưởng từ bên ngoài chia Watermark.
  + Bền vững với các tác động sao chép trái phép.
  + Dễ phá hủy với các tác động trên.
- Chia Watermark theo đặc tính:
  + Cần che giấu đối với mặt người.
  + Phải được mọi người nhìn thấy.

1.3 Giấu tin trong dữ liệu đa phương tiện

1.3.1 Giấu 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 còn 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 thông tin được giấu một cách vô hình, nó như là cách truyền thông tin mật cho nhau mà người khác không thể biết được bởi sau khi giấu thông tin chất lượng ảnh gần như không thay đổi đặc biệt đối với ảnh màu hay ảnh xám hoặc có thay đổi nhỏ nhưng mắt người không thể phát hiện ra.

1.3.2 Giấu tin trong Audio

- Kỹ thuật này phụ thuộc vào hệ thống thính giác của con người, sử dụng các âm thanh to cao tần để che giấu các âm thanh nhỏ, thấp.

- Giấu thông tin trong audio mang những đặc điểm riêng khác với giấu thông tin trong các đối tượng đa phương tiện khác như ảnh, video, văn bản... 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).

- Khó khăn của việc giấu thông tin trong audio:

+ Thứ nhất: Hệ thống thính giác của con người nghe được các tín hiệu ở các giải tần rộng và công suất lớn nên đó 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 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.

+ Thứ hai: Đó 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.

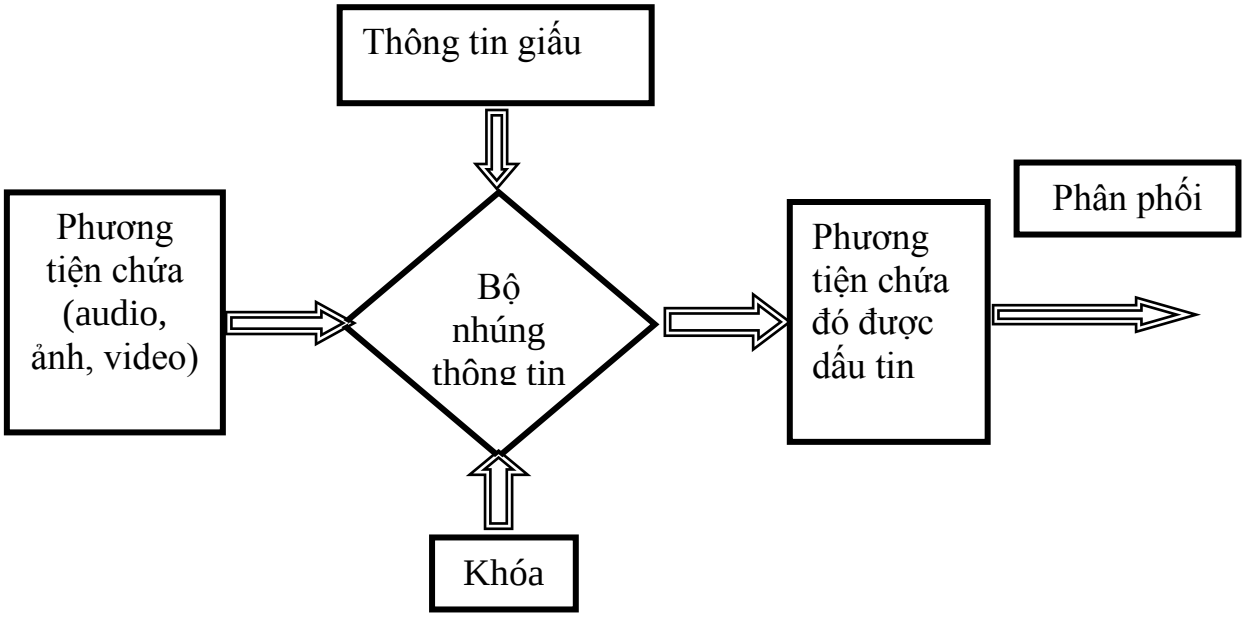
1.3.3 Giấu thông tin trong video

- Ý tưởng cơ bản của phương pháp này là phân phối thông tin giấu giàn trải theo tần số của dữ liệu gốc. Cụ thể giấu cả âm thanh và hình ảnh vào video. Phương pháp này được đưa ra bởi Cox và được nhiều nhà nghiên cứu

thử nghiệm dùng các hàm cosin riêng và các hệ số truyền sóng riêng để giấu tin và đem lại hiểu quả cao.

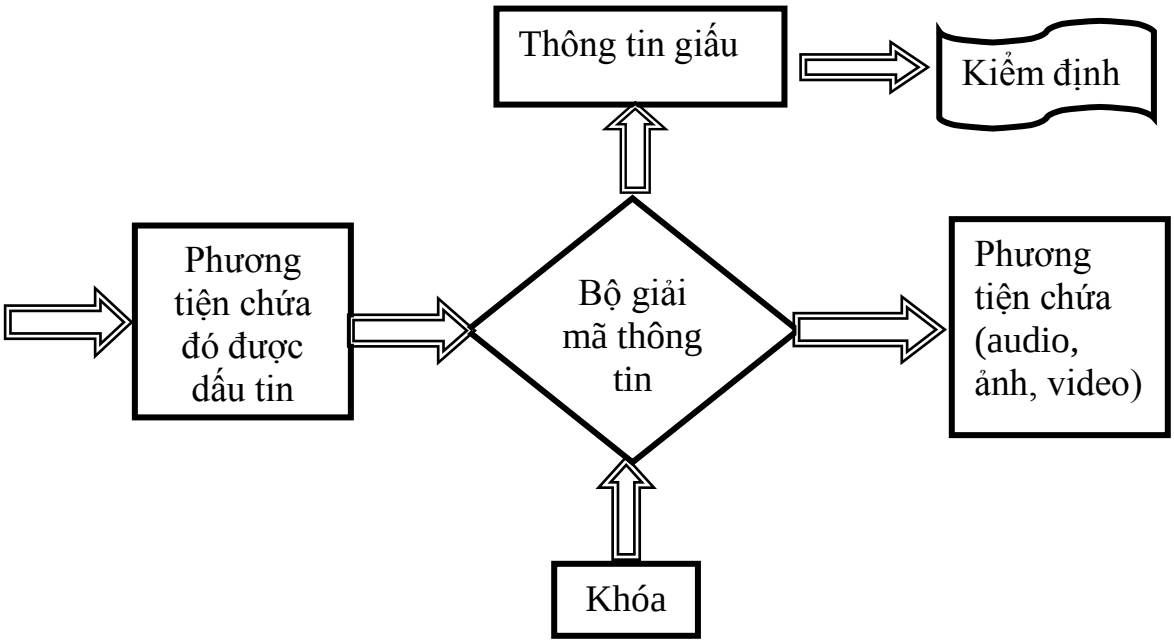
1.4 Mô hình kỹ giấu và phát hiện thông tin cơ bản

Giấu thông tin vào trong phương tiện chứa và tách lấy thông tin là hai quá trình trái ngược nhau và được mô tả như sau:



Hình 2: Lược đồ chung cho quá trình giấu tin.

- Thông tin cần giấu tuỳ theo mục đích của người sử dụng, nó có thể là thông điệp (với các tin bí mật) hay các logo, hình ảnh bản quyền.
- Phương tiện chứa: các file ảnh, text, audio… là môi trường để nhúng tin.
- Bộ nhúng thông tin: là những chương trình thực hiện việc giấu tin.
- Phương tiện chứa được dấu tin: là các phương tiện chứa mà đã được giấu thông tin trong đó.
- Khóa: là khóa bí mật dùng để giấu tin.
- Phân phối: sau khi giấu tin xong phương tiện chứa thông tin sẽ được phân phối đi với nhiều hình thức khác nhau.



Hình 3:Lược đồ quá trình giải mã thông tin

Tách thông tin từ các phương tiện chứa diễn ra theo quy trình ngược lại với đầu ra là các thông tin đã được giấu vào phương tiện chứa. Phương tiện chứa sau khi tách lấy thông tin có thể được sử dụng, quản lý theo những yêu cầu khác nhau.

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

1.5 Một số ứng dụng

- Bảo vệ bản quyền tác giả.
- Nhận thực thông tin hay xác định xuyên tạc.
- Dấu vân tay hay dán nhãn.
- **....**

CHƯƠNG 2: CẤU TRÚC ẢNH GIF VÀ KỸ THUẬT NÉN LZW

2.1 Cấu trúc của ảnh GIF

Ảnh GIF (Graphics Interchange Format) là một định dạng tập tin hình ảnh bitmap cho các hình ảnh dùng ít hơn 256 màu sắc khác nhau và các hoạt hình dùng ít hơn 256 màu cho mỗi khung hình. Gif thường dùng cho sơ đồ, hình vẽ, nút bấm và các hình màu. GIF là định dạng nén dữ liệu đặc biệt hữu ích cho việc truyền hình ảnh qua đường truyền lưu lượng nhỏ. Đây là một giải pháp tốt cho hình ảnh trên mạng, cho các hoạt hình nhỏ và ngắn.

GIF sử dụng thuật toán nén LOSS LESS (Không mất dữ liệu), điều đó cho phép chúng tạo ra kích thước nhỏ mà không bị mất hoặc mờ bất kỳ chi tiết nào của ảnh dữ liệu .

GIF note
GIF header (7 byte)
Globel Palette
Header Image (10 byte)
Palette of Image (nếu có)
Data of Image 1
‘,’ ký tự liên kết
……..
‘;’ terminator

Hình 4: Cấu trúc ảnh Gif

+ Chữ ký của ảnh GIF có giá trị là GIF87a. Nó gồm 6 ký tự, 3 ký tự đầu chỉ ra kiểu định dạng, 3 ký tự sau chỉ ra version của ảnh.

+ Bộ hình thị: chứa mô tả các thông số cho toàn bộ ảnh GIF:

Độ rộng hình raster theo pixel: 2 byte.

Độ cao hình raster theo pixel: 2 byte.

Các thông tin và bản đồ màu, hình hiển thị,…

Thông tin màu nền: 1 byte.

Phần chưa dùng: 1 byte.

+ Bản đồ màu tổng thể: mô tả bộ màu tối ưu đòi hỏi khi bit M=1.

Khi bộ màu tổng thể được thể hiện, nó sẽ xác định ngay bộ mô tả hiển thị ở trên và bằng 2^m, với m là lượng bit trên một pixel, 3 byte (biểu diễn cường độ màu của 3 màu cơ bản Red-Green-Blue).

Cấu trúc của khối này như sau:

Bit	Thứ tự byte	Mô tả
Màu Red	1	*Giá trị màu đỏ theo index 0*
Màu Green	2	Giá trị màu xanh lục theo index 0
Màu Blue	3	Giá trị màu xanh lơ theo index 0
Màu Red	4	Giá trị màu đỏ theo index 1
Màu Green	5	Giá trị màu xanh lục theo index 1
Màu Blue	6	Giá trị màu xanh lơ theo index 0

Hình 5: Cấu trúc của khối bản đồ màu tổng thể

Các bit	Thứ tụ byte	Mô tả
0010110	1	Ký tự liên kết ảnh (‘)
Căn trái ảnh	2,3	Pixel bắt đầu ảnh tính từ trái hình hiển thị
Căn đỉnh trên	4,5	Pixel cuối ảnh bắt đầu tính từ đỉnh trên hình hiển thị
Độ rộng ảnh	6,7	Độ rộng ảnh tính theo pixel
Độ cao ảnh	8,9	Chiều cao ảnh tính theo pixel
MI000pixel	10	Khi bit M=0 sử dụng bảng màu tổng thể. M=1 sử dụng bản đồ màu cục bộ. I = 0: định dạng ảnh theo thứ tự liên tục. I = 1: định dạng ảnh theo thứ tự xen kẽ pixel + 1: số bit/pixel của ảnh này.

Hình 6: Cấu trúc bộ mô tả ảnh

+ Bộ mô tả ảnh: định nghĩa vị trí thực tế và phần mở rộng của ảnh trong phạm vi không gian ảnh đã có trong phần mô tả hiển thị. Nếu ảnh biểu diễn theo ánh xạ màu cục bộ thì cờ định nghĩa phải được thiết lập. Mỗi

bộ mô tả ảnh được chỉ ra bởi ký tự kết nối ảnh. Ký tự này chỉ được dùng khi định dạng GIF có từ 2 ảnh trở lên. Ký tự này có các giá trị 0x2c (ký tự dấu phẩy). Khi ký tự này được đọc qua, bộ mô tả ảnh sẽ được kích hoạt. Bộ mô tả ảnh gồm 10 byte và có cấu trúc như sau:

+ Bản đồ màu cục bộ: chỉ được chọn khi bit M của byte thứ 10 là 1. Khi bản đồ màu được chọn, bản đồ màu sẽ chiếu theo bộ mô tả ảnh mà lấy vào cho đúng. Tại phần cuối ảnh, bản đồ màu sẽ lấy lại phần xác lập sau bộ mô tả hiển thị. Các tham số này không những chỉ cho biết kích thước ảnh theo pixel mà còn chỉ ra số thực thể bản đồ màu của nó.

+ Dữ liệu ảnh: chuỗi các giá trị có thứ tự của các pixel màu tạo nên ảnh. Các pixel được xếp liên tục trên một dòng ảnh, từ trái qua phải. Các dòng ảnh được viết từ trên xuống dưới.

+ Phần kết thúc ảnh: cung cấp tính đồng bộ cho đầu cuối của ảnh GIF. Cuối của ảnh sẽ xác định bởi kí tự ";" (0x3b).

Định dạng GIF có rất nhiều ưu điểm và đã được công nhận là chuẩn để lưu trữ ảnh màu thực tế (chuẩn ISO 0918-1).

2.2 Kỹ thuật nén dữ liệu LZW

2.2.1 Giới thiệu

Có 2 dạng nén ảnh: lossless (trung thực) và lossy (không trung thực). Dùng lossless, ảnh sau khi giải nén (decompressed image) hoàn toàn giống với bản ban đầu (trước khi nén). Nén kiểu lossy làm mất một số thông tin. Nghe qua thì có vẻ đáng ngại, nhưng nếu được thực thi tốt, bằng mắt thường không thể phân biệt ảnh tời với ảnh gốc, mà kỹ thuật này đảm bảo được tỉ lệ nén rất cao.

Với các ảnh đen trắng, GIF là sơ đồ nén thực sự trung thực. Ảnh màu là vấn đề khác. GIF chỉ làm việc được với ảnh màu lập sẵn chỉ mục (indexed

color image) và một lượng lớn thông tin bị mất khi chuyển ảnh màu 24 bit thành ảnh màu 8 bit có chỉ mục - làm giảm số màu có thể từ 16,7 triệu xuống còn 256. Một ảnh nhỏ cỡ 320x240 điểm có thể nhiều màu hơn 300 lần so với trường hợp màu có chỉ mục, kết quả là ảnh GIF 8 bit hoặc 5 bit không được mịn và rõ. Tuy vậy, GIF có nhiều mặt mạnh. Trước hết, và quan trọng nhất, đây là chuẩn về thực tế, được mọi Web browser đồ họa hổ trợ. Nếu dùng GIF, chắc chắn bất cứ ai, ở bất cứ đâu, đều có thể sử dụng được tập tin đó. GIF còn là dạng thức duy nhất được chấp nhận rộng rãi cho phép sử dụng các điểm trong suốt (transparent pixels) trong file ảnh. Nó còn hỗ trợ interlacing (đan xen), một phương thức cấu trúc hóa thông tin trong tập tin, cho phép ảnh được đưa liên tục ra màn hình, ảnh cũ mờ dần, ảnh mới rõ nét lên.

2.2.2 Giải thuật

Một chuyên gia giải thích kỹ thuật nén lossless như sau: "Giả sử bạn có một ngăn kéo với 2 chiếc tất màu trắng, 2 chiếc tất màu đen. Thay vì nói: "Tôi có 1 tất trắng, 1 tất trắng nữa, 1 tất đen, 1 tất đen nữa", bạn sẽ giảm câu đi khoảng một nửa nếu nói: "Tôi có 1 cặp tất trắng và một cặp tất đen".

Phương pháp Run Length Encoding (RLE), một kiểu nén lossless đơn giản nhất, làm việc như sau: Khi nén, tìm các đoạn lặp đi lặp lại - nếu thấy có hàng gồm 9 số không, tiếp theo là 3 số một, 12 số không, tất cả sẽ được thay bằng 3 số: 9, 3, 12. Cách này hiệu quả nhất đối với các ảnh có vùng lớn đồng màu, nhưng kém hiệu lực với các ảnh phức tạp.

Phương pháp Lempel-Ziv-Welch (LZW) và kỹ thuật mã hóa kiểu Huffman phân tích và quan sát các đoạn lặp lại. Nếu LZW hoặc Huffman thấy có đoạn 010101, chúng đủ thông minh đánh dấu các đoạn như vậy và thay bằng một ký tự, bằng cách đó dữ liệu được nén lại.

GIF sử dụng LZW cho TIFF nén. Tỉ lệ nén đạt ở mức vừa phải là 2:1. Để đạt được tỉ lệ cao hơn, cần đến kỹ thuật JPEG, JPEG 2000.

2.2.3 Phương pháp nén LZW

Phương pháp nén LZW được phát minh bởi Lempel - Zip và Welch. Nó hoạt động đựa 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 một dấu hiệu (token) riêng để tạo thuận lợi cho qua trình thay thế.
+ 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 (00ffh).
+ **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: Ta cần mã hoá chuỗi "!BAN!BA!BAA!BAR!" như sau.

+ 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.
+ Bước 3: Kí tự thứ ba 'A' thêm vào sau 'B'. Chuỗi "BA" còn 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" còn chưa có trong "từ điển" nên được thêm vào "từ điển" và có dâu 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 ngoài từ điển.
+ 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!".

Stt	Bộ đệm chứa	Dữ liệu vào (8 bit)	Dữ liệu ra (12 bit)	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	-	-

Hình 7 :Các bước nén

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 dể 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.

2.2.4 Thuật toán nén LZW

Thuật toán nén:

Mã:

```
w = NIL;
while (read a char c) do
if (wc exists in dictionary) then
    w = wc;
else
    add wc to the dictionary;
    output the code for w;
    w = c;
endif
done
output the code for w;
```

Thuật toán giải nén

Mã:

```
read a char k;
output k;
w = k;
while (read a char k) do
    if (k > 255 && k exists in dictionary) then
    entry = dictionary entry for k;
else
    entry = k;
endif
output entry;
add w+entry[0] to the dictionary;
w = entry;
done
```

CHƯƠNG 3: MỘT SỐ KỸ THUẬT GIẤU TIN TRONG ẢNH GIF

3.1 Kỹ thuật giấu tin EzStego

Bước 1:Sắp xếp bảng màu

Ezstego copy bảng màu của ảnh. Sau đó sắp xếp lại bảng copy đó của bảng màu sao cho các màu được sắp xếp gần giống nhau.

Đây là 9 màu của ảnh GIF nhìn lúc trước và sau khi sắp xếp.

Bước 2: Giấu thông tin.

Ezstego đặt bit cần giấu vào LSB (least significant bit) của pixel ảnh theo các bước thực hiện sau:

- Tìm chỉ số màu RGB của pixel trong bảng được sắp.
- Lấy một bít cần giấu thay thế cho LSB của chỉ số bảng màu.
- Tìm màu RGB mới mà chỉ số bây giờ trỏ vào trong bảng màu đã có sẵn từ trước.
- Tìm chỉ số màu RGB mới này trong bảng màu ban đầu.
- Thay đổi cho chỉ số màu của màu mới.

Ví dụ:

- 17 231 31 là màu 00100101 trong bảng màu đã được sắp.
- Giá trị chỉ số của 00100101 được thay đổi thành 00100100.
- Màu 00100100 trong bảng màu đã được sắp là 179 233 36.
- 179 233 36 là màu 11101110 trong bảng màu gốc.
- Giá trị của pixel được đổi thành 11101110.

Bước 3: Khôi phục lại bít đã giấu.

Tìm chỉ số màu của RGB trong bảng màu được sắp.

Bít ít quan trọng nhất từ các bít giấu. Viết nó ở đầu ra.

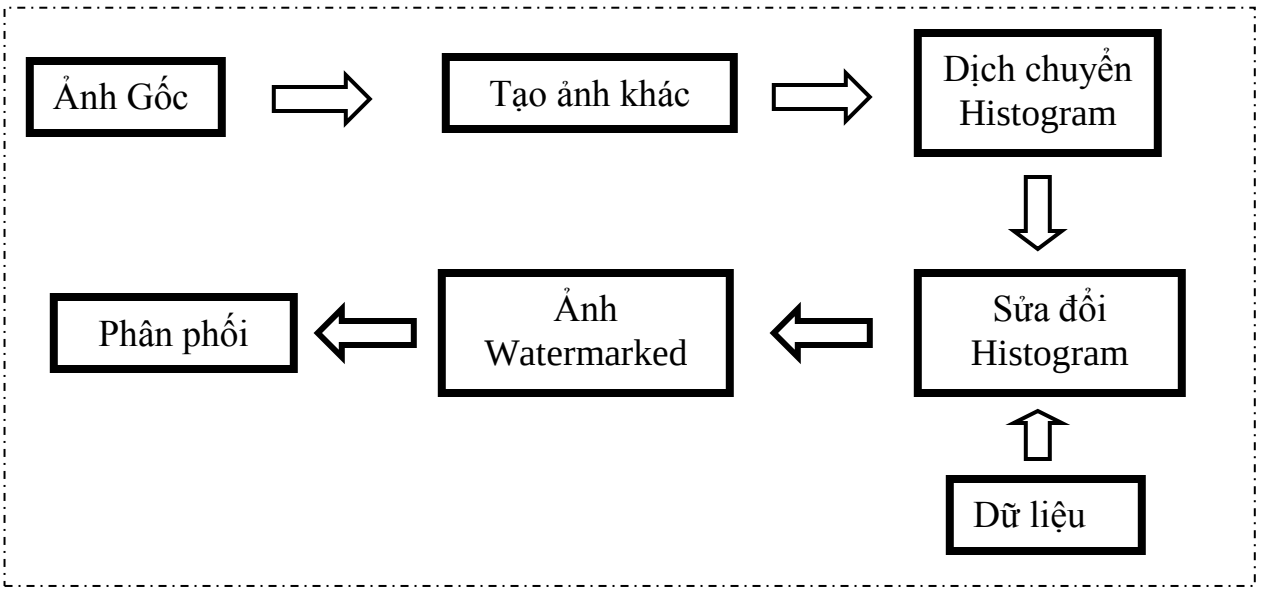
Ví dụ:

- 179 233 36 là màu 11101110 trong bảng màu đã đươc sắp.
- Bít ít quan trọng nhất là bít 0.
- Viết 0 cho đầu ra.

3.2 Kỹ thuật giấu tin DIH

Thuật toán Difference Image Histogram (DIH) được đề xuất bởi Sang-Kwang Lee, Young-Ho Suh, và Yo-Sung Ho năm 2003. Thuật toán này nhúng thông điệp cần giấu vào histogram của difference image sửa đổi. Chuỗi thông điệp giấu được giấu vào các pixel có giá trị 1 hoặc -1 trong *difference image* sửa đổi. Số lượng pixel có giá trị 1 hoặc -1 thể hiện khả năng giấu lượng bit thông điệp vào ảnh gốc.

3.2.1 Quá trình giấu thông tin



Hình 8: Lược đồ quá trình giấu tin DIH

Các bước thực hiện:

- Bước 1: Tạo ảnh khác D

+ Với mỗi hình ảnh xám *(grayscale)* *I(i, j)* của kích thước M × N pixel, ta hình thành nên ảnh khác *D(i, j)* của kích thước M × N / 2 từ bản gốc hình ảnh

$$D(i, j) = I(i, 2j + 1) - I(i, 2j),\ 0 \le i \le M - 1,\ 0 \le j \le N/2 - 1 \quad \textbf{(1)}$$

Trong đó I(i, 2j + 1) và I(i, 2j) là các trường lẻ và chẵn tương ứng*(odd line field and the even line field)*

- Bước 2: Dịch chuyển và thay đổi Histogram

+ Khi nhúng watermark, ta làm rỗng các vùng -2 và 2 bằng việc thay đổi một vài giá trị điểm ảnh trong ảnh khác. Nếu các giá trị trong ảnh khác lớn hơn hoặc bằng 2, ta cộng thêm 1 vào những điểm hàng lẻ. Nếu các giá trị trong ảnh khác nhỏ hơn hoặc bằng -2, ta trừ 1 trong những điểm hàng lẻ

$$\tilde{D}(i, j) = \tilde{I}(i, 2j+1) - I(i, 2j)$$

$$\tilde{I}(i, 2j+1) = \begin{cases} I(i, 2j+1) + 1 \text{ if } D(i, j) \ge 2 \\ I(i, 2j+1) + 1 \text{ if } D(i, j) \ge 2 \\ I(i, 2j+1) \end{cases}$$

- Bước 3: Thực hiện giấu thông điệp

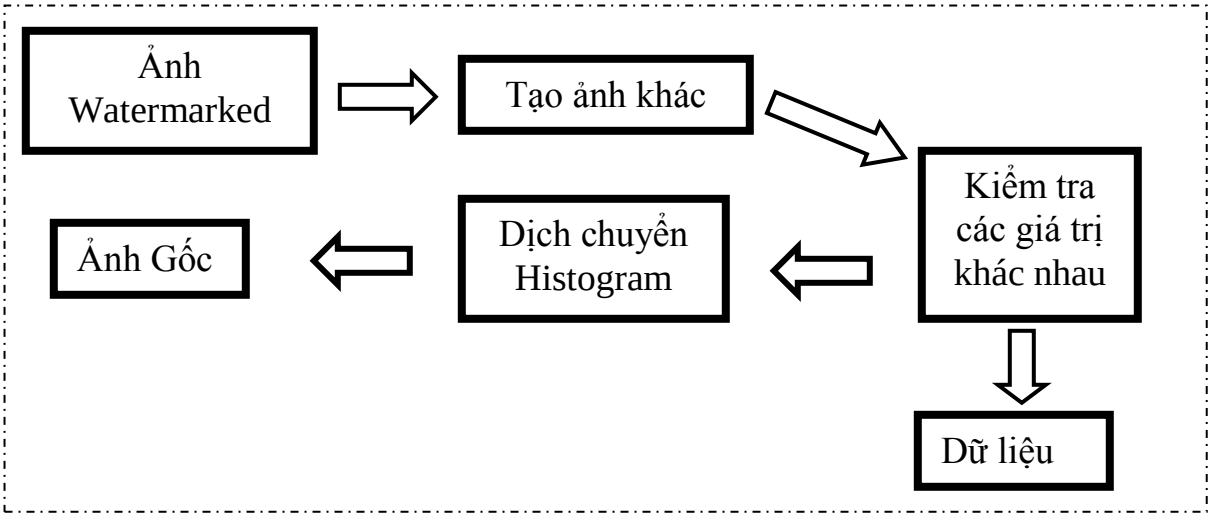
+ W(m, n) là thông điệp được giấu. Sau khi ta gặp một điểm ảnh $\tilde{D}(i,j)$ có giá trị -1 hoặc 1,ta kiểm tra watermark để nhúng vào. Nếu $\tilde{D}(i,j) = 1$ và $W(m,n) = 1$ thì $I_w(i,2j+1) = I_e(i,2j+1) + 1$. Nếu $\tilde{D}(i,j) = -1$ và $W(m,n) = 1$ thì $I_w(i,2j+1) = I_e(i,2j+1) - 1$. Còn các bit được nhúng vào là 0, ta bỏ qua các điểm ảnh của ảnh khác cho đến khi ta gặp một điểm ảnh có giá trị -1 hoặc 1.

Trong trường hợp này, không có sự thay đổi trong biểu đồ. Do đó I_w (i, 2j + 1) và I_w *(i, 2j)* được tạo lên:

$$I_w(i,2j+1) = \begin{cases} \tilde{I}(i,2j+1)+1 \text{ if } \tilde{D}(i,j)=1 \text{ and W(m,n) = 1} \\ \tilde{I}(i,2j+1)-1 \text{ if } \tilde{D}(i,j)=-1 \text{ and W(m,n) = 1} \\ \tilde{I}(i,2j+1) \qquad \text{otherwise} \end{cases}$$

$$I_w(i,2j) = \mathrm{I}\ (i,2j)$$

3.2.2 Quá trình lấy thông tin



Hình 9: Lược đồ quá trình lấy tin DIH

- Sau khi có được ảnh watermarked $I_e(i, j)$, Áp dụng công thức (1) ta được $D_e(i, j)$. Nếu gặp các điểm ảnh có giá trị -1 hoặc 1, thì bit 0 được lấy. Nếu gặp các điểm ảnh có giá trị -2 hoặc 2 thì bit 1 được lấy. Bằng cách này $W_e(m,n)$ có thể được lấy ra:

$$W_e(m,n) = \begin{cases} 0 \text{ if } D_e(i,j) = -1 \text{ or } 1 \\ 1 \text{ if } D_e(i,j) = -2 \text{ or } 2 \end{cases}$$

- Để khôi phục được ảnh gốc, ta dịch chuyển một số pixel trong I_e như sau: nếu D_e (I,j) có giá trị ≤ -2 thì tăng thêm *1* vào I_e *(I,2j+1)*, nếu D_e có giá trị ≥ 2 thì giảm *1* tại I_e *(I,2j+1)*. Cuối cùng ta sẽ thu được ảnh gốc ban đầu:

$$I_r(i,2j+1) = \begin{cases} I_e(i,2j+1)-1 \text{ if } D_e(i,j) \geq 2 \\ I_e(i,2j+1)+1 \text{ if } D_e(i,j) \leq -2 \\ I_e(i,2j+1) \text{ otherwise} \end{cases}$$

$$I_r(i,2j) = I_e(i,2j)$$

- Phương pháp giấu DIH có thể không trả về được ảnh gốc hoàn toàn đúng như ban đầu bởi việc mất mát thông tin xảy ra trong quá trình cộng trừ tại biên của vòng xám (mức xám là từ 0 ÷ 255). Để khắc phục vấn đề này, họ đưa ra modul số học cho các phép cộng và trừ thủy vân. Đối với trường lẻ I(i,2j+1), phép cộng modul c như sau:

$$I(i,2j+1) +_c 1 = ((i,2j+1) + 1) \bmod c$$

Với c là độ dài của vòng xám. Đối với phép trừ modul *c* được định nghĩa như sau:

$$I(i,2j+1) -_c 1 = ((i,2j+1) + 1) \bmod c$$

- Những vấn đề thuận nghịch được phát sinh từ sự thừa, thiếu hụt pixel. Vì vậy, ta sử dụng $+_c$ và $-_c$ thay vì + và – chỉ khi bỏ bớt do thừa hay thiếu hụt xảy ra. Nói cách khác, ta chỉ để xem xét $255 +_c 1$ và $0 -_c 1$.

- Khi nhận được, ta cần phân biệt giữa các trường hợp, ví dụ $I_e(i, 2j+1) = 255$ có được như: $I(i, 2j+1) + 1$ và $I(i, 2j+1) -_{256} 1$. Nếu có một sự khác biệt đáng kể giữa $I_e(i, 2j+1)$ và $I_e(i, 2j)$, ta ước lượng $(i, 2j+1)$ vận dụng modulo số học.

$$I(i,2j+1) = \begin{cases} I(i,2j+1)+1 & \text{if } \left|I_e(i,2j+1) - I_e(i,2j)\right| \leq \tau \\ I(i,2j+1) -_{256} 1 & \text{otherwise} \end{cases}$$

Trong đó τ là giá trị ngưỡng,Tương tự $I_e(i, 2j+1) = 0$ được ước lượng bằng cách:

$$I(i,2j+1) = \begin{cases} I(i,2j+1)-1 & \text{if } \left|I_e(i,2j+1) - I_e(i,2j)\right| \leq \tau \\ I(i,2j+1) +_{256} 1 & \text{otherwise} \end{cases}$$

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1 Môi trường cài đặt

- Cài đặt chương trình trên môi trường Java, sử dụng bộ soạn thảo JCreator_Pro_v4.5 và thông dịch JDK-6u10.

- Cầu hình máy tính tối thiểu để chạy chương trình:Hệ diều hành Windown XP hoặc các hệ điều hành tương tự, Chip PIII 500 trở lên, Ram >=128, ổ cứng còn trống 400 Mb.

- Chương trình gồm các chức năng sau:

+ **Giấu tin**: Quá trình thực hiện như sau:

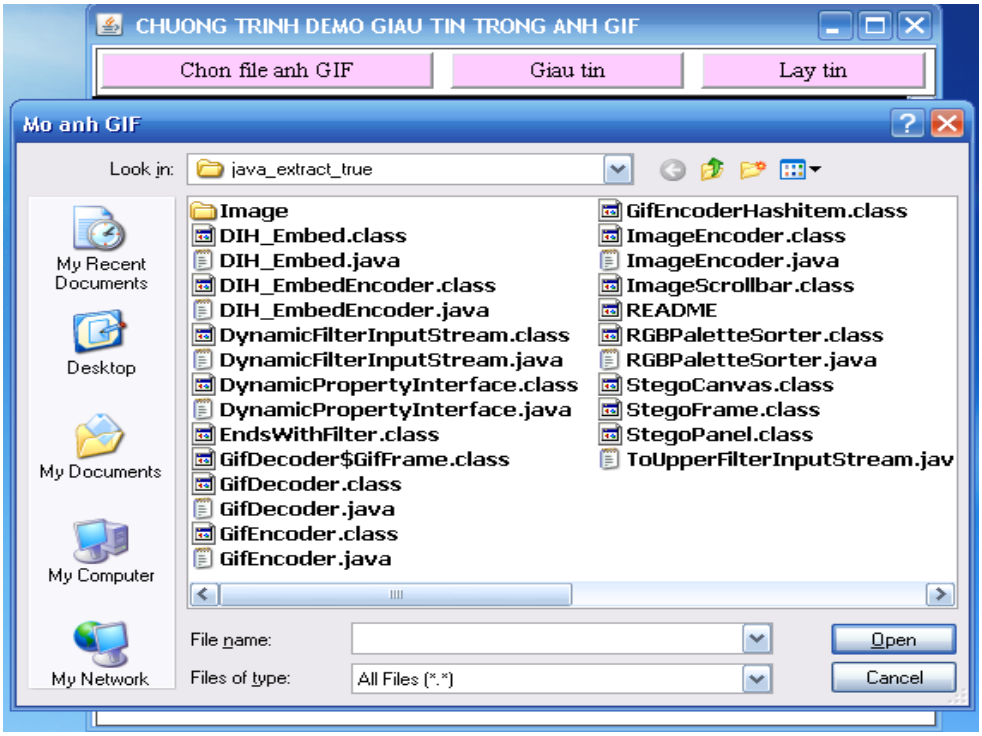
Chọn file ảnh GIF → Giấu tin → Chọn vị trí lưu file ảnh output.gif → Chọn file text cần giấu.

+ **Lấy tin:** Quá trình thực hiên như sau:

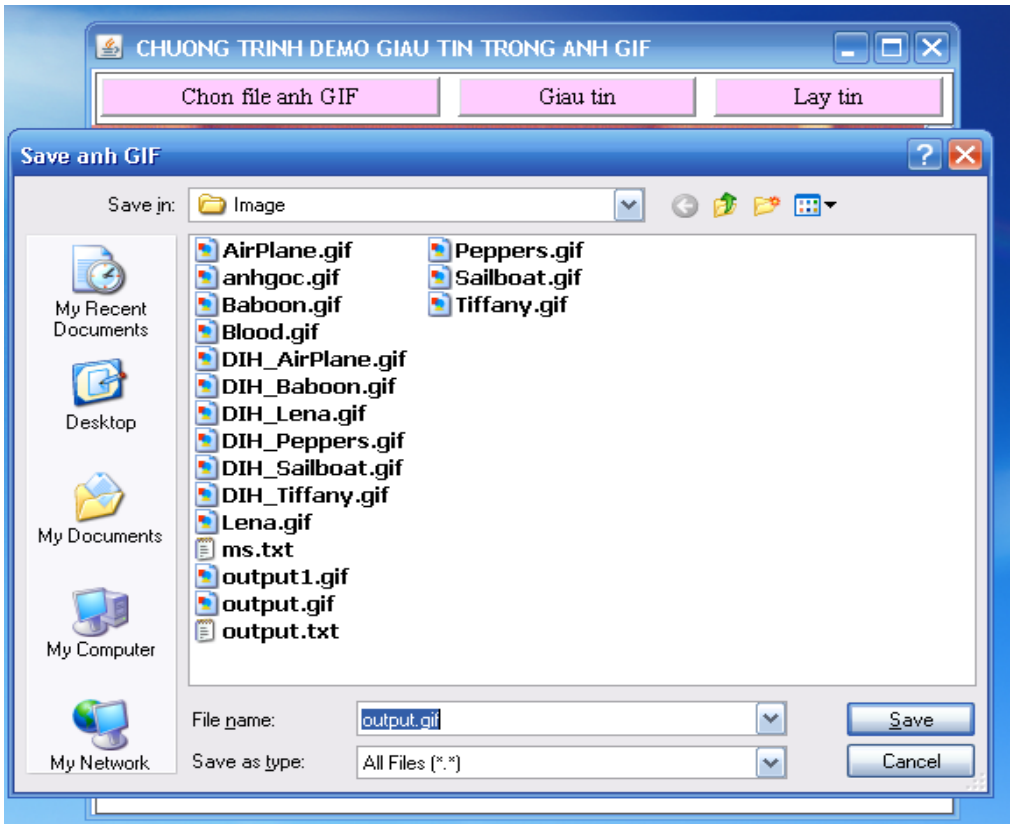
Chọn file ảnh GIF → Lấy tin → Chọn vị trí lưu file text output.txt → Chọn vị trí lưu file ảnh gốc anhgoc.gif

- Giao diện của chương trình:

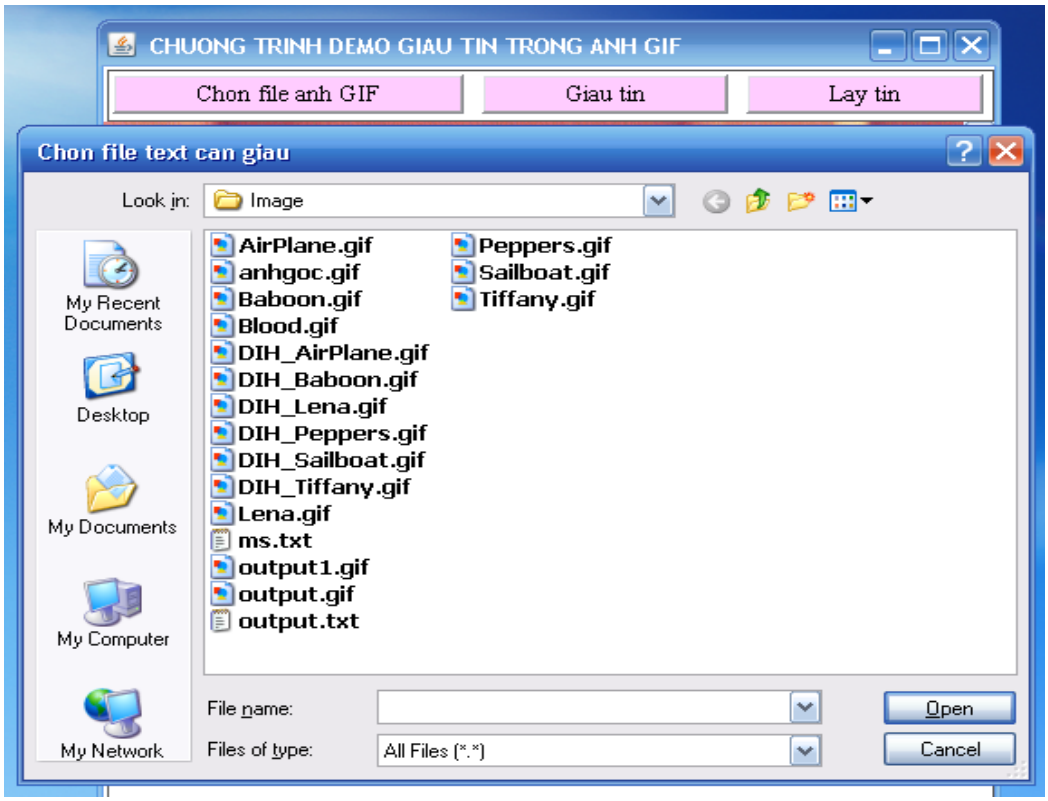
Hình 10: Giao diện chính của chương trình



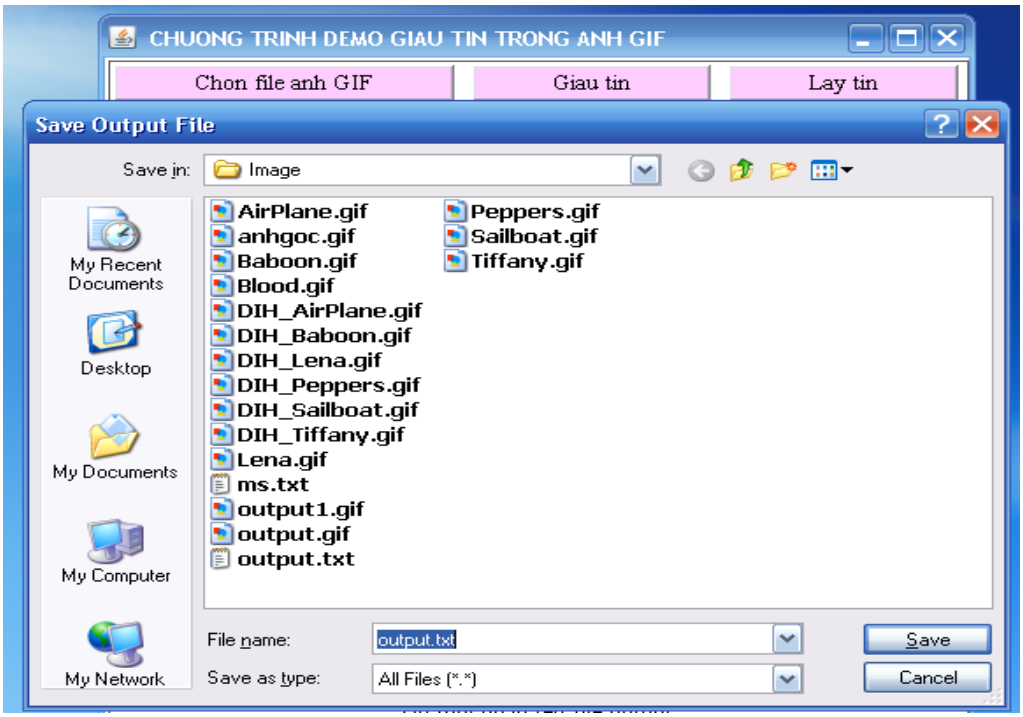
Hình 11: Chọn file ảnh GIF cần giấu



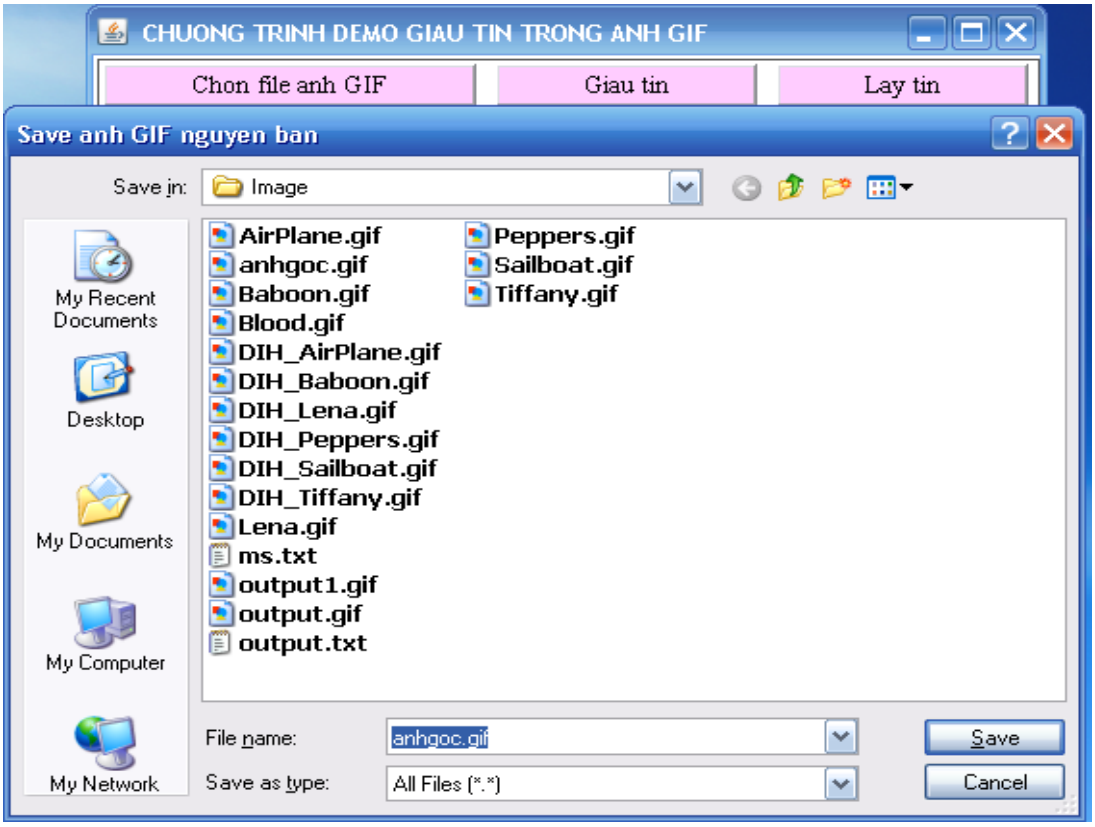
Hình 12: Chọn vị trí lưu file ảnh mới output.gif



Hình 13: Chọn file text cần giấu



Hình 14: Chọn vị trí lưu file text output.txt

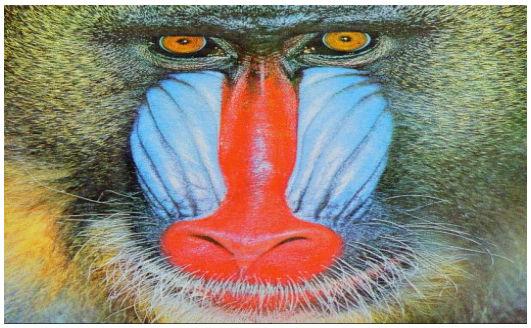


Hình 15: Chọn vị trí lưu file ảnh gốc anhgoc.gif

4.2 Cơ sở dữ liệu thử nghiệm

Có một tập cơ sở dữ liệu ảnh gồm 6 ảnh GIF chuẩn được download từ [5] và [6] có kích cỡ 512x512 pixel.

Airplane.gif



Baboon.gif

Peppers.gif

Lena.gif

Sailboat.gif

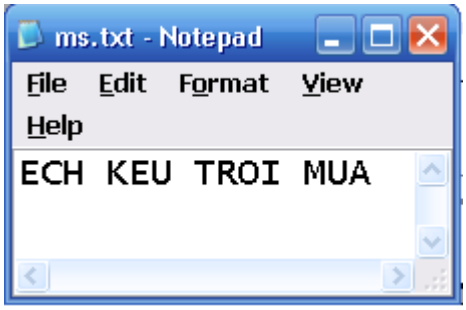
Tiffany.gif

Hình 16: Các hình ảnh GIF thử nghiệm

4.3 Kết quả thử nghiệm và đánh giá thuật toán bằng (PSNR)

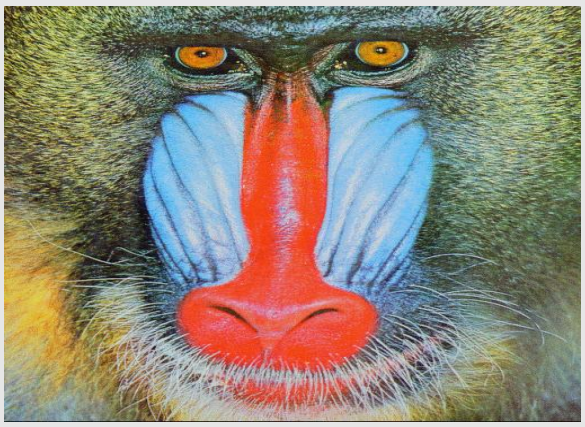
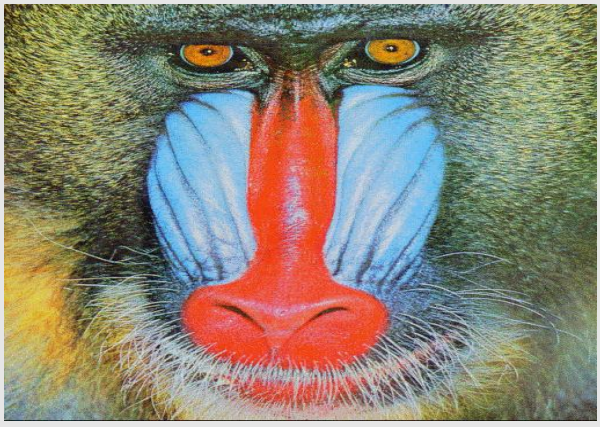
Để đánh giá hiệu quả hoạt động của phương pháp đề xuất, chúng ta thực hiện trên nhiều máy tính mô phỏng trên một vài ảnh GIF kích thước 512 × 512 pixels.

-Chuỗi ký tự cần giấu:



Hình 17: Chuỗi kỹ tự cần giấu

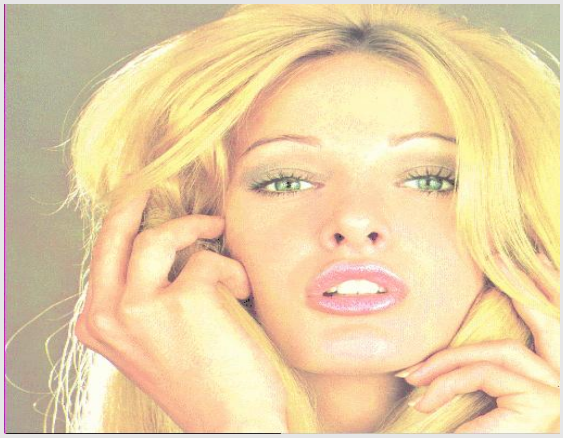
- Kết quả thực nghiệm:

Ảnh gốc	Ảnh Wartermared
	
Airplane.gif	Airplane.gif
	
Sailboat.gif	DIH_Sailboat.gif
	
Baboon.gif	DIH_Baboon.gif

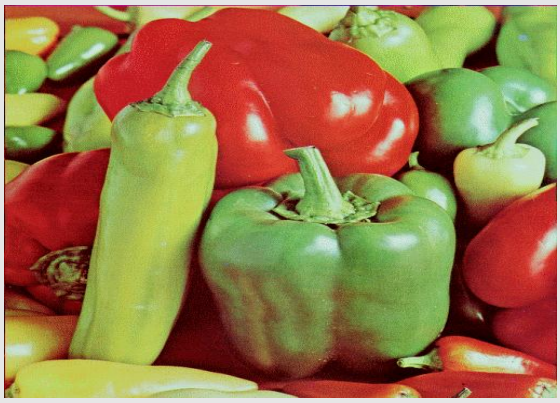
Lena.gif

DIH_Lena.gif

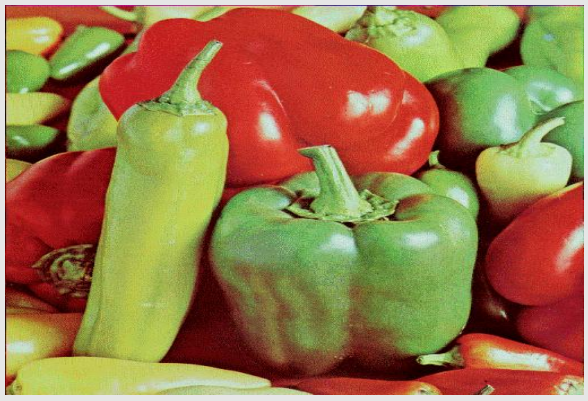
Tiffany.gif



DIH_Tiffany.gif



Peppers.gif



DIH_Peppers.gif

Hình 18:Ảnh trước và sau khi giấu tin

- Đánh giá thuật toán bàng PSNR

Bảng 1 tóm tắt các kết quả thử nghiệm. Bảng này cho thấy rằng các giá trị PSNR của tất cả các hình ảnh watermarked đang ở trên 51,14 dB.Khả năng dao động từ 8 kbits đến 30 kbits của 512 × 512 × 8 bits.

Ảnh (512x512x8)	**PSNR (dB)**	**Khả năng giấu (bit)**	**Vượt ngưỡng (pixels)**
Airplane	58.78	13,551	0
Baboon	51.49	14,111	2
Lena	55.63	16.379	0
Peppers	55.74	23,725	2
Sailboat	55.55	17,719	14
Tiffany	55.20	20,497	1

Bảng 19:Bảng tóm tắt kết quả thực nghiệm.

KẾT LUẬN

Sau một thời gian học tập và tìm hiểu, dưới sự hướng dẫn tận tình của cô giáo hướng dẫn ThS. Hồ Thị Hương Thơm cùng sự giúp đỡ của các thầy cô bộ môn tin trong trường, trong quá trinh thực hiện báo cáo tốt nghiệp, báo cáo đã được hoàn thành.

Tuy nhiên, giấu và phát hiện tin ẩn giấu vẫn là cón là vấn đề mới mẻ, phức tạp, nhất là lĩnh vực phát hiện tin ẩn giấu, cộng với khả năng và kinh nghiệm còn hạn chế nên em còn gặp một số khó khăn trong việc nghiên cứu các kỹ thuật giấu tin trên ảnh GIF.

Vì vậy em rất mong nhận được sự đóng góp ý kiến của các thầy cô giáo trong khoa để báo của em được hoàn thiện hơn

Hải Phòng, ngày ... tháng 07 năm 2009

SINH VIÊN

Mạc Như Hiển

TÀI LIỆU THAM KHẢO

[1]. Nguyễn Xuân Huy, Trần Quốc Dũng, Giáo trình giấu tin và thủy vân số.

[2]. Lương Mạnh Ba, Nguyễn Thanh Thủy, Nhập môn xử lý ảnh số

[3]. Romana Machado. How Stego Online Works

http://www.fqa.com/stego_com/howto.html , http://www.fqa.com/romana/

[4]. Sang-Kwang Lee, Young-Ho Suh and Yo-Sung Ho: "Lossless Data Hiding Based on Histogram Modification of Difference Images "

http://www.springerlink.com/content/yd16tlbmqxkbx16l/

[5].CBIR image database, University of Washington, available at:

http://www.cs.washington.edu/research/imagedatabase/groundtruth/

[6].USC-SIPIImage Database

http://www.sipi.usc.edu/services/database/Database.html