MỤC LỤC

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc nhất tới cô giáo ThS. Hồ Thị Hương Thơm, cô đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của cô, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm luận án tốt nghiệp.

Em xin chân thành cảm ơn sự dạy bảo và giúp đỡ của các thầy giáo, cô giáo Khoa Công Nghệ Thông Tin – Trường Đại học Dân Lập Hải Phòng đã trang bị cho em những kiến thức cơ bản nhất để em có thể hoàn thành tốt báo cáo tốt nghiệp này.

Xin cảm ơn tới những người thân trong gia đình quan tâm, động viên trong suốt quá trình học tập và làm tốt nghiệp.

Xin gửi lời cảm ơn tất cả bạn bè, đặc biệt là các bạn trong lớp CT901 đã giúp đỡ và đóng góp ý kiến để mình hoàn thành chương trình.

Một lần nữa em xin chân thành cảm ơn !

MỞ ĐẦU

Cuộc cách mạng thông tin 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 số mang lại cũng sinh ra những thách thức cũng như cơ hội mới cho quá trình phát triển. Internet và mạng không dây đã trợ giúp cho việc chuyển phát một khối lượng thông tin rất lớn qua mạng. Tuy nhiên nó cũng làm tăng nguy cơ sử dụng trái phép, xuyên tạc bất hợp pháp các thông tin được lưu chuyển trên mạng, đồng thời việc sử dụng một cách bình đẳng, an toàn các dữ liệu đa phương tiện cũng như cung cấp một cách kịp thời tới rất nhiều người dùng cuối và các thiết bị cuối cũng là một vấn đề quan trọng và còn nhiều thách thức. 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 tác giả 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 việc 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 nhiều sự quan tâm của cả giới khoa học và giới công nghiệp nhưng cũng còn rất nhiều thách thức.

Bản báo cáo này trình bày về giấu và phát hiện ảnh có giấu thông tin. Đồng thời trình bày một số kỹ thuật giấu và phát hiện thông tin ẩn giấu trong ảnh GIF, từ đó đưa ra các thực nghiệm và đánh giá cho việc phát hiện thông tin ẩn giấu trong ảnh GIF.

Nội dung báo cáo gồm các chương:

Chương 1. Tổng quan về kỹ thuật giấu tin và giấu tin trong ảnh.

Chương 2. Ảnh và kỹ thuật nén dữ liệu LZW.

Chương 3. Một số kỹ thuật giấu tin trên ảnh GIF.

Chương 4. Kỹ thuật phát hiện thông tin ẩn giấu trên ảnh GIF.

Chương 5. Kết quả thử nghiệm.

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

1.1 Định nghĩa kỹ thuật giấu tin

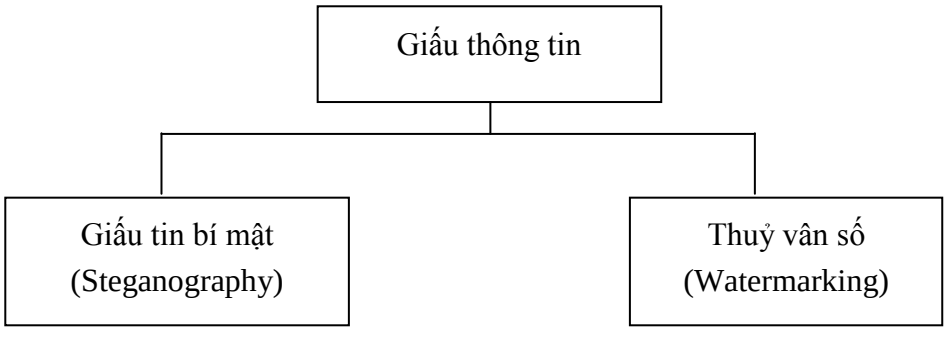
Giấu tin là kỹ thuật giấu hoặc nhúng một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác (giấu tin nhiều khi không phải là hành động giấu cụ thể mà chỉ mang ý nghĩa quy ước)

1.2 Mục đích của giấu tin

Có hai mục đích của giấu tin:

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

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



Hình 1. Hai lĩnh vực chính của kỹ thuật giấu thông tin

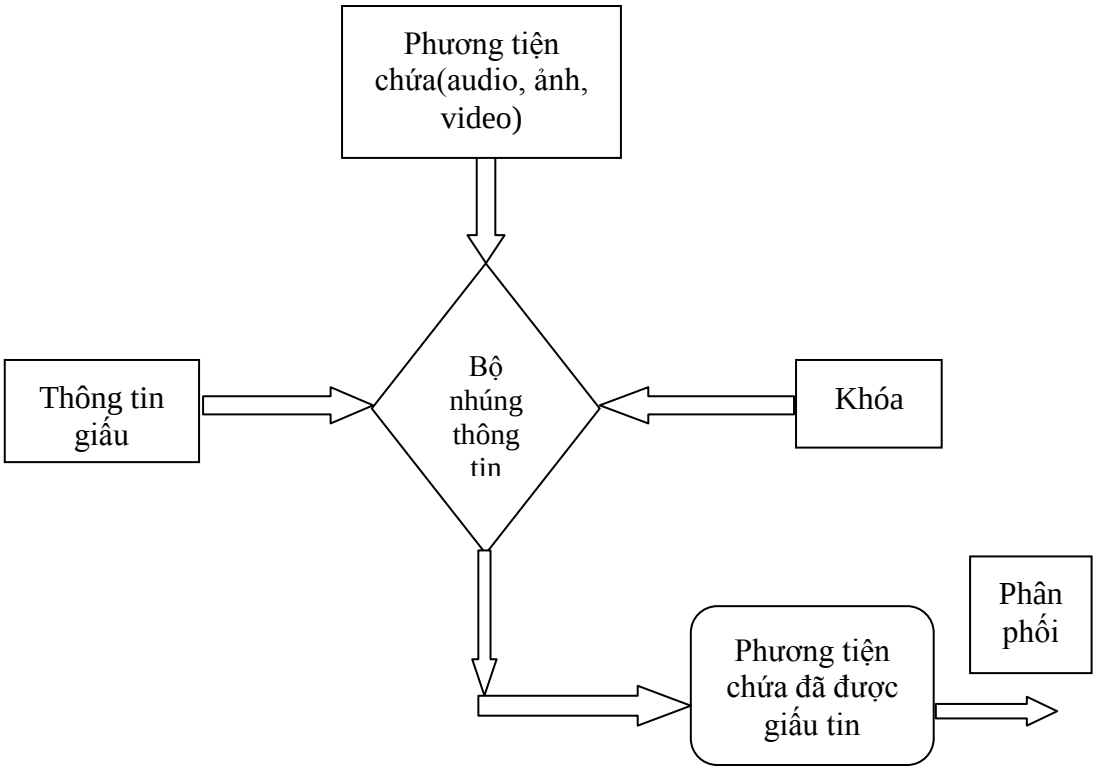
Kỹ thuật giấu thông tin bí mật (Steganography): mục đích là đảm bảo an toàn và bảo mật thông tin tập trung vào các kỹ thuật giấu tin để có thể

giấu được nhiều thông tin nhất. Thông tin mật được giấu trong đối tượng sao cho người khác không phát hiện được.

Kỹ thuật giấu thông tin theo kiểu đánh giấu (watermarking) để bảo vệ bản quyền của đối tượng chứa thông tin thì lại tập trung đảm bảo một số yêu cầu như đảm bảo tính bền vững… Đây chính là ứng dụng cơ bản nhất của kỹ thuật thuỷ vân số.

1.2.1 Mô hình kỹ thuật giấu thông tin cơ bản

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



Hình 2. Lược đồ chung cho quá trình giấu tin

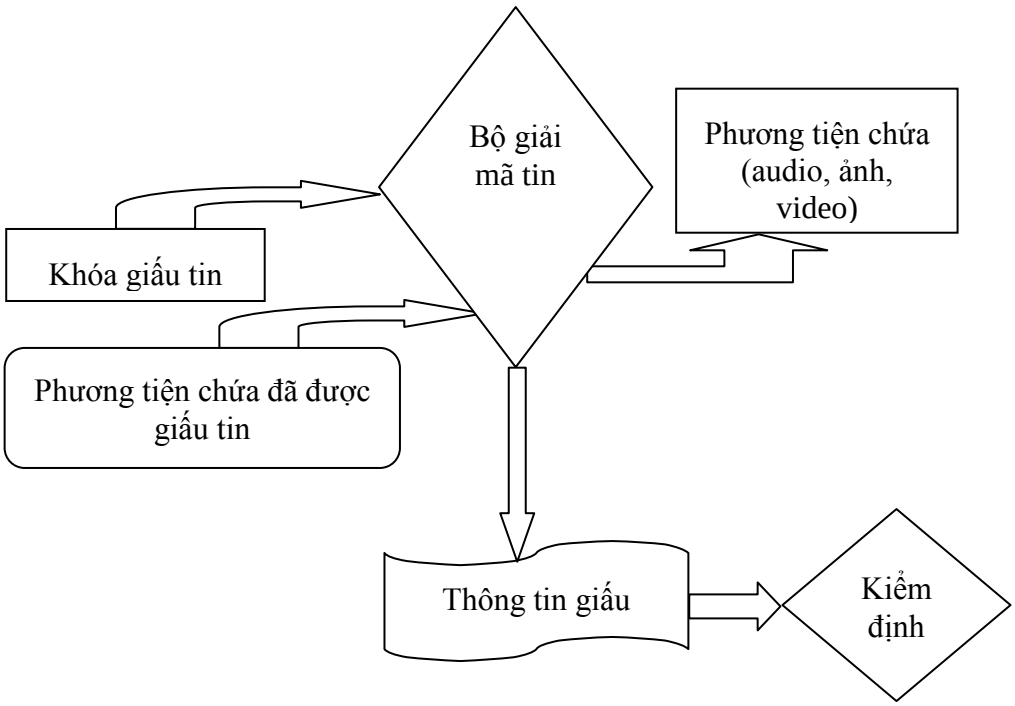
- Thông tin cần giấu tuỳ theo mục đích của người sử dụng, nó có thể là thông điệp (với các tin bí mật) hay các logo, hình ảnh bản quyền.
- Phương tiện chứa: các file ảnh, text, audio… là môi trường nhúng tin.

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

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

Tách thông tin từ các phương tiện chứa diễn ra theo một 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.

1.2.2 Mô hình kỹ thuật giải mã thông tin cơ bản



Hình 3. Lược đồ chung cho quá trình giải mã

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 được xử lý kiểm định so sánh với thông tin ban đầu.

1.3 Môi trường giấu tin.

1.3.1 Giấu tin trong ảnh

Ngày nay khi ảnh số đã được sử dụng rất phổ biến thì giấu thông tin trong ảnh đã đem lại nhiều những ứng dụng quan trọng trên các lĩnh vực trong đời sống xã hội. Ví dụ như ở các nước phát triển chữ ký tay đã được số hoá và lưu trữ sử dụng như là hồ sơ cá nhân của các dịch vụ ngân hàng tài chính.

Một đặc điểm của giấu thông tin trong ảnh nữa đó là thông tin được giấu một cách vô hình, nó như là cách truyền thông tin mật cho nhau mà người khác không thể biết được bởi sau khi giấu thông tin chất lượng ảnh gần như không thay đổi đặc biệt đối với ảnh màu hay ảnh xám.

1.3.2 Giấu tin trong audio

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

1.3.3 Giấu tin trong video

Cũng giống như giấu thông tin trong ảnh hay trong audio, giấu tin trong video cũng được quan tâm và được phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, nhận thức thông tin, bản quyền tác giả… Một phương pháp giấu tin trong video được đưa ra bởi Cox là phương pháp phân bố đều. Ý tưởng cơ bản của phương pháp là phân phối thông tin giấu dàn trải theo tần số của dữ liệu gốc.

1.3.4 Giấu thông tin trong văn bản dạng text

Giấu tin trong văn bản dạng text khó thực hiện hơn do có ít thông tin dư thừa, để làm được điều này người ta phải khéo léo khai thác các dư thừa tự nhiên của ngôn ngữ. Một cách khác là tận dụng các định dạng văn bản (mã hoá thông tin vào khoảng cách giữa các từ hay các dòng văn bản).

Kỹ thuật giấu tin đang được áp dụng cho nhiều loại đối tượng chứ không riêng gì dữ liệu đa phương tiện như ảnh, audio, video.

CHƯƠNG 2: ẢNH GIF VÀ KỸ THUẬT NÉN DỮ LIỆU LZW

2.1 Cấu trúc ảnh GIF

Ảnh GIF (Graphics Interchange Format) là định dạng tập tin hình ảnh bitmap cho các hình ảnh dùng ít hơn 256 màu 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 hơn 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.
- Bộ mô tả hiển thị.
- Bản đồ màu tổng thể.

Mô tả một đối tượng của ảnh.

- Dấu phân cách.

- Bộ mô tả ảnh.
- Bản đồ màu cục bộ.
- Dữ liệu ảnh. Phần mô tả này lặp lại n lần nếu ảnh chứa n đối tượng.
- Phần đầu cuối ảnh GIF (terminator).
    + 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 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

+ 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 hai 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:

Các bit	Thứ tự 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

+ 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 cùng 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 ảnh GIF. Cuối của ảnh sẽ xác định bởi kí tự ";" (0x3b). Định dạng GIF có rất nhiều ưu điểm và 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 ảnh 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, ta không thể phân biệt ảnh đã giải nén với ảnh gốc, 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 – bạn 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, GIF là chuẩn về thực tế, được mọi Web browser đồ họa hỗ trợ. Nếu bạn dùng GIF, chắc chắn bất cứ ai, ở bất cứ đâu, đều có thể sử dụng được tập tin đó.

GIF 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 bạn nói: "Tôi có 1 cặp tất trắng và một cặp tất đen".

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 và 12 số không, thì tất cả sẽ 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 để đánh dấu các đoạn đó 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 độ 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 dữ liệu LZW được phát minh bởi Lempel – Zip và Welch. Nó hoạt động dựa trên một ý tưởng rất đơn giản là người mã hoá và người giải mã cùng xây dựng bản mã.

Nguyên tắc hoạt động của nó như sau:

- Một xâu kí tự là một tập hợp từ hai kí tự trở lên.
- Nhớ tất cả các xâu kí tự đã gặp và gán cho nó một dấu hiệu (token) riêng.
- Nếu lần sau gặp lại xâu kí tự đó, xâu kí tự sẽ được thay bằng dấu hiệu của nó.

Phần quan trọng nhất của phương pháp nén này là phải tạo ra 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 được giữ lại trong bộ đệm chứa (Accumulator) và 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 có trong bộ đệm chứa đã lưu 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 hiên việc nén dữ liệu theo thuật toán LZW là:

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

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

Qui tắc 3: Các kí tự đầu vào nhận từ tập tin sẽ được nén được bổ sung vào bộ đệm chứa cho 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 tạo thành chuỗi mới tiếp theo.

Ví dụ: Các bước để mã hoá chuỗi "!BAN!BA!BAA!BAR!" như sau (Bảng 1):

- Bước 1: Kí tự thứ nhất '!' được cất vào bộ đệm chứa 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 thêm vào "từ điển" và được gán dấu hiệu là 100h (vì khoảng 000h ÷ 0ffh được dành 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" 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" chưa có trong "từ điển" nên được thêm vào "từ điển" và có dấu hiệu là 102h. 'N' ở lại trong bộ đệm chứa còn 'A' được gửi ra.
- Bước 5: Kí tự thứ năm '!' thêm vào sau 'N' để tạo thành chuỗi "N!", "N!" được thêm vào "từ điển" với dấu hiệu là 103h. '!' ở lại còn 'N' được gửi ra.
- Bước 6: Kí tự thứ sáu 'B' thêm vào sau '!'. Lần này thì chuỗi "B!" đã có trong "từ điển" nên không có kí tự nào được gửi ra. "B!" tiếp tục ở lại trong "từ điển" để tạo ra chuỗi mới.

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

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

Trong thuật toán nén này, phần lớn thời gian khi bắt đầu nén chủ yếu mất vào việc tạo "từ điển". Khi "từ điển" đủ lớn, xác suất gặp chuỗi trong bộ đệm chứa ở "từ điển" tăng lên và càng nén được nhiều hơn. Một điều cần chú ý ở đây là mỗi một dấu hiệu, ta phải lưu một chuỗi trong "từ điển" để so sánh.

Vì dấu hiệu được biểu diễn bằng một số 12 bits nên "từ điển" sẽ có 4096 lối vào, khi tăng số bit để biểu diễn dấu hiệu lên thì hiệu quả nén sẽ tốt hơn nhưng lại bị giới hạn bởi bộ nhớ của máy tính. Vì dụ, khi ta sử dụng 16 bits để biểu diễn một dấu hiệu thì "từ điển" phải có đến 65536 lối vào, nếu mỗi lối vào có 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 mình xây dựng bảng mã mà không cần bên gửi phải gửi kèm theo bản tin nén.

STT	Bộ đệm chứa	Dữ liệu vào (8 Bits)	Dữ liệu ra (12 Bits)	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	-	-

Bảng 1. Ví dụ

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:

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: KỸ THUẬT GIẤU TIN TRÊN ẢNH GIF

3.1 Khái niệm bit có trọng số thấp LSB (Least Significant Bit)

Least Significant Bit là bit có ảnh hưởng ít nhất đến việc quyết định tới màu sắc của mỗi điểm ảnh. Vì vậy khi thay đổi bit ít quan trọng của một điểm ảnh thì màu sắc của mỗi điểm ảnh mới sẽ tương đối gần với điểm ảnh cũ. Ví dụ, đối với ảnh 16 bit thì sẽ có 15 bit là biểu diễn 3 màu RGB của điểm ảnh còn bit cuối cùng không dùng đến thì ra sẽ tách bit này ra ở mỗi điểm ảnh để giấu tin… Như vậy kỹ thuật tách bit trong xử lý ảnh sẽ được sử dụng rất nhiều trong quy trình giấu thông tin. Việc xác định LSB của mỗi điểm ảnh phụ thuộc vào định dạng của ảnh và số bit màu dành cho mỗi điểm của ảnh đó.

3.2 Kỹ thuật giấu tin EzStego

Thuật toán EzStego được đề xuất bởi Romana Machado vào tháng 10 năm 1996

Ý tưởng: Sắp xếp bảng màu copy của ảnh gốc sao cho các màu được sắp xếp gần giống nhau. Sau đó thực hiện giấu thông điệp trên LSB của pixel ảnh.

Thuật toán:

Input: Ảnh gốc và tệp tin cần giấu.

Output: File ảnh có chứa thông tin ẩn giấu.

Các bước thực hiện:

B1: 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.

Cho bảng màu gốc *Cold = {ci, i = 0,..., n − 1}*. Cho *I (ci,Cold) ≡ i* là chỉ số của *ci* trong bảng màu cũ và cho *δ(a, b)* là khoảng cách giữa 2 màu *a* và *b*

- Sắp xếp:

1. D ← {c0}, C← Cold\{c0}; c ← c0

2. Tìm màu d € C là khoảng cách từ c

3. D ← {c0, d} ≡ {d0, d1}, C← C\{d}

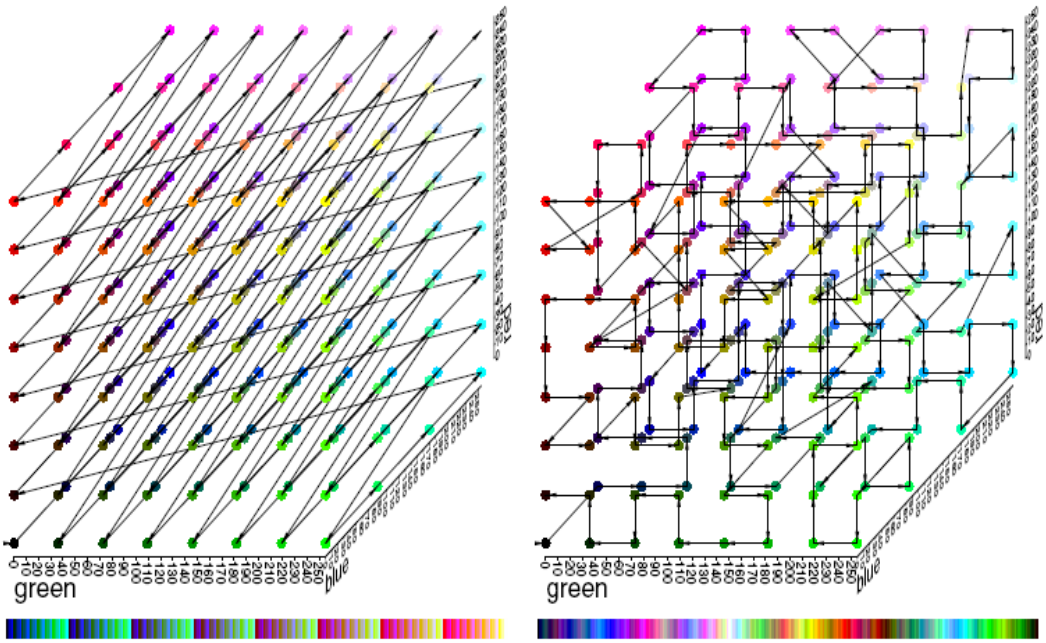
4. Trong khi C _= ∅ thực hiện.

5. Tìm màu d € C là khoảng cách từ c.

6. Tìm 2 màu {di, di+1} € D và δ{di, d} + δ{d, di+1} là nhỏ nhất.

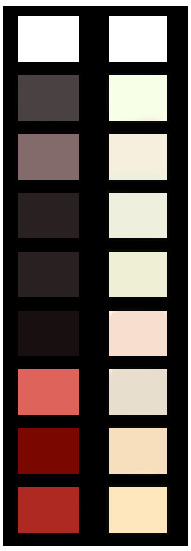
7. D ← {d0,..., di, d, di+1,...}, C← C\ d, c ← d

8. Kết thúc.



Bảng màu cũ Bảng màu sau khi sắp xếp

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



B2: Giấu thông tin.

Ezstego đặt bít 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.
- Tìm chỉ số màu RGB mới này trong bảng màu ban đầu.
- Thay đổi cho chỉ số màu của màu mới.

Ví dụ:

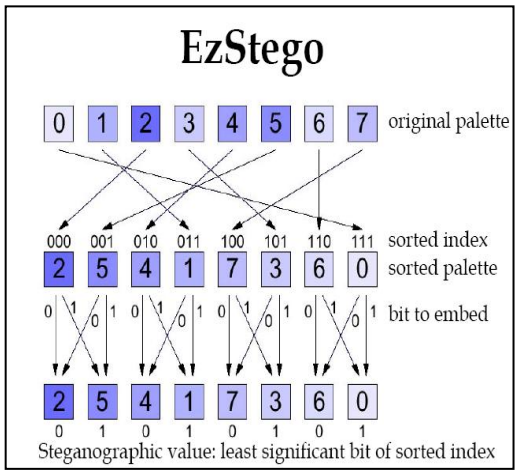
- 17 231 31 là màu 00100101 trong bảng màu đã được sắp.
- Giá trị chỉ số của 00100101 được thay đổi thành 00100100.
- Màu 00100100 trong bảng màu đã được sắp là 179 233 36.

- 179 233 36 là màu 11101110 trong bảng màu gốc.

- Giá trị của pixel được đổi thành 11101110.

B3: 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.

3.3 Thuật toán giấu DIH

Thuật toán Difference Image Histogram (DIH) được đề xuất bởi Sang-Kwang Lee, Young-Ho Suh, và Yo-Sung Ho năm 2003.

Ý tưởng: 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.

Thuật toán:

Input: Ảnh gốc và file thông điệp cần giấu.

Output: File ảnh đã giấu tin.

Các bước thực hiện:

Bước 1. Quá trình thực hiện giấu tin:

+ Quá trình đọc ảnh đầu vào và xử lý Histogram của ảnh:

- Lấy dữ liệu ảnh đầu vào (histogram ảnh gốc)

- Tạo DI (Defference Image). D(i, j) được tính như sau:

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

Với *MxN* là kích cỡ ảnh gốc.

- Tính DI điều chỉnh bằng cách tăng thêm 1 nếu $D(i, j)$ có giá trị ≥ 2, giảm đi 1 nếu $D(i, j)$ có giá trị ≤ -2. Ngược lại các giá trị được giữ nguyên. Ta có:

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

Trong đó:

$$\tilde{I}(i, 2j + 1) = \begin{cases} I(i, 2j + 1) + 1 & \text{if } D(i, j) \ge 2 \\ I(i, 2j + 1) - 1 & \text{if } D(i, j) \le -2 \\ I(i, 2j + 1) & \text{otherwise} \end{cases} \quad (3)$$

+ Thực hiện giấu thông điệp:

W(m, n) là thông điệp được giấu. Nếu bit được giấu là 1 thì $\tilde{D}(i, j)$ có giá trị *1* sẽ thành *2*, *–1* thành *–2*. Nếu bit được giấu là 0 thì $\tilde{D}(i, j)$giữ nguyên.

$$I_w(i, 2j + 1) = \begin{cases} \tilde{I}(i, 2j + 1) + 1 & \text{if } \tilde{D}(i, j) = 1 \text{ and } W(m, n) = 1 \\ \tilde{I}(i, 2j + 1) - 1 & \text{if } \tilde{D}(i, j) = -1 \text{ and } W(m, n) = 1 \\ \tilde{I}(i, 2j + 1) & \text{otherwise} \end{cases} \quad (4)$$

$$I_w(i,2j) = I(i,2) \quad (5)$$

Bước 2. Tách thông điệp được giấu và khôi phục ảnh gốc:

Thực hiện quá trình đọc ảnh đầu vào và xử lý Histogram của ảnh ta thu được ảnh $I_e(i, j)$, DI của ảnh vừa giấu tin $D_e(i, j)$ và DI điều chỉnh $\tilde{D}_e(i,$

j). Trong $\tilde{D}_e(i, j)$, bit được giấu là *0* nếu $\tilde{D}_e(i, j) = 1$ hoặc *-1*, bit được giấu là *1* nếu $\tilde{D}_e(i, j) = 2$ hoặc *-2*.

$$W_e(m,n) = \begin{cases} 0 & \text{if } D_e(i,j) = -1 \text{ or } 1 \\ 1 & \text{if } D_e(i,j) = -2 \text{ or } 2 \end{cases} \quad (6)$$

Để khôi phục được ảnh gốc, ta dịch chuyển một số pixel trong DI như sau: nếu pixel trong DI có giá trị ≤ *-2* thì tăng thêm *1* giá trị, nếu pixel trong DI có giá trị ≥ *2* thì giảm *1* giá trị. Cuối cùng ta sẽ thu được ảnh gốc ban đầu:

$$I_r(i,2j+1) = \begin{cases} I_e(i,2j+1) - 1 & \text{if } D_e(i,j) \geq 2 \\ I_e(i,2j+1) + 1 & \text{if } D_e(i,j) \leq -2 \\ I_e(i,2j+1) & \text{otherwise} \end{cases} \quad (7)$$

$$\boldsymbol{Ir(i,2j) = Ie(i,2j)} \quad (8)$$

Bước 3. Xử lý vượt ngưỡng:

Phương pháp giấu DIH có thể không trả về 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:

$$\boldsymbol{I(i,2j+1) +_c 1 = ((i,2j+1) + 1) \bmod c} \quad (9)$$

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

$$\boldsymbol{I(i,2j+1) -_c 1 = ((i,2j+1) + 1) \bmod c} \quad (10)$$

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, nó là cần thiết để phân biệt giữa các trường hợp, ví dụ, $I_e\ (i,\ 2j\ +1)\ =\ 255$ có được như: $I(i,\ 2j\ +1)\ +1$ và $I(i,\ 2j\ +1)\ -_{256}\ 1$. Và họ cho rằng không có sự thay đổi xảy ra giữa hai điểm kề bên. Nếu có một sự khác biệt đáng kể giữa $I_e(i,\ 2j\ +\ 1)$ và $I_e(i,\ 2j)$, ta sẽ ước lượng $(i,\ 2j\ +\ 1)$ vận dụng modulo số học.

$$\begin{cases} I(i,2j+1)+1 & \text{if } |I_e(i,2j+1)-I_e(i,2j)| \le \tau \\ I(i,2j+1)-_{256}1 & \text{otherwise} \end{cases} \tag{11}$$

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

$$\begin{cases} I(i,2j+1)-1 & \text{if } |I_e(i,2j+1)-I_e(i,2j)| \le \tau \\ I(i,2j+1)+_{256}1 & \text{otherwise} \end{cases} \tag{12}$$

CHƯƠNG 4: KỸ THUẬT PHÁT HIỆN TIN ẨN GIẤU TRÊN ẢNH GIF

4.1 Tổng quan về kỹ thuật phát hiện tin ẩn giấu trong ảnh

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

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

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

Các phương pháp phân tích có thể phân thành 3 nhóm:

- Phân tích trực quan
- Phân tích theo dạng ảnh
- Phân tích theo thống kê

Westfeld đã đề xuất phương pháp steganalytic, nó chính là cơ sở của phương pháp PoVs.

Tính chất ẩn thường hay xảy ra trong quá trình lập mã của phương pháp PoVs trong ảnh stego ngang bằng trong khi tính chất đó trong phương pháp PoVs với ảnh có bề mặt ảnh sáng rõ lại không ngang bằng.

Đây là một phương pháp có thể tìm ra những bức ảnh bị nghi ngờ mà không có những bức ảnh nguyên bản (sự dò tìm không rõ ràng).

4.2 Kỹ thuật phát hiện DIH và ước lượng tin ẩn giấu bằng DIH

Kỹ thuật phát hiện DIH ước lượng histogram của ảnh cover và ảnh stego và thống kê sự khác biệt đó. Kết quả cho thấy rằng nó có thể nhận một tỉ lệ thay đổi T_0 trong ảnh có sử dụng kỹ thuật giấu DIH.

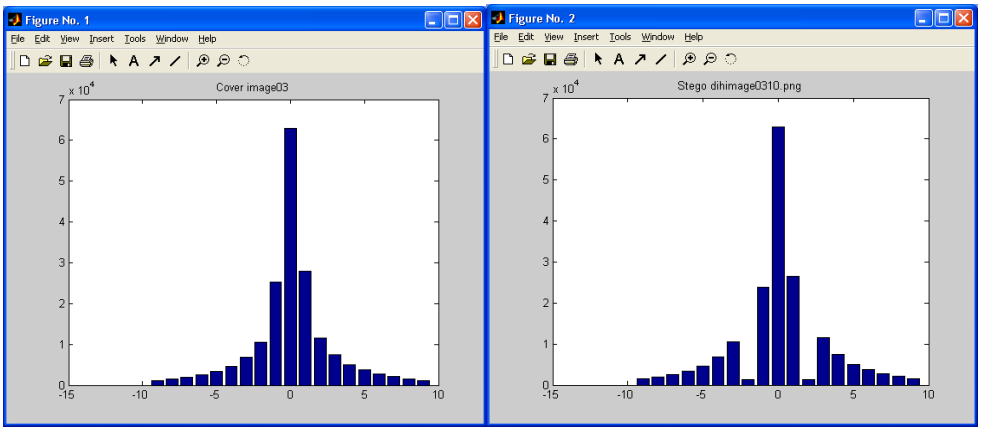
Qua thực nghiệm cho thấy, sau khi nhúng thông điệp bằng thuật toán DIH sẽ làm thay đổi tổng số histogram $h_{\pm 2}$ của ảnh (Bảng 2).

Đối với ảnh không giấu tin, tổng số histogram :

$$h_1 + h_{-1} > h_2 + h_{-2} > h_3 + h_{-3} > \ldots\ldots > h_{10} + h_{-10} \quad (1)$$

Đối với ảnh có giấu tin, ta có:

$$h_2 + h_{-2} \leq h_3 + h_{-3} \quad (2)$$



Ảnh gốc Ảnh có giấu tin

Ý tưởng: Xét tỷ lệ của (h_2 + h_{-2}) với (h_3 + h_{-3}). So sánh tỷ lệ này với hệ số thực nghiệm T_0.

Thuật toán:

Input: Một tập ảnh Q với kích thước chung của các ảnh là 512x512

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

Các bước thực hiện như sau:

Bước 1. Tính Histogram của ảnh cần kiểm tra

Bước 2. So sánh tỷ lệ giữa $h_{\pm 2}$ và $h_{\pm 3}$:

Nếu $(h_2 + h_{-2}) / (h_3 + h_{-3}) \geq T_0$ thì ảnh kiểm tra là ảnh gốc.

Ngược lại, nếu $(h_2 + h_{-2}) / (h_3 + h_{-3}) < T_0$ ta có ảnh là ảnh đã giấu thông tin (T_0=1.15 là hệ số được xác định bằng thực nghiệm).

Bước 3. Ước lượng độ dài thông điệp giấu:

Bằng các thực nghiệm khoa học về ngôn ngữ tự nhiên, người ta đã chứng minh được rằng, trong mỗi thông điệp ẩn giấu bất kỳ đều có tỷ lệ tổng số bit 0 và 1 là sấp sỉ 50 : 50. Từ đó, một cách tính ước lượng thông điệp ẩn giấu được đề xuất như sau:

- Gọi L là độ dài ước lượng thông điệp ẩn giấu trên tổng histogram $h_{\pm 1}$, và L_i là tỷ lệ thông điệp đã giấu so với khả năng giấu của ảnh, được tính như sau:

$$L = 2*(h_2 + h_{-2}) \quad (3)$$

$$L_i = [2(h_2 + h_{-2})/ (h_1 + h_2 + h_{-1} + h_{-2})] * 100 \quad (4)$$

- Gọi [p, q] là kích thước ảnh. Ước lượng tỷ lệ phần trăm của ảnh có chứa thông điệp ẩn giấu như sau:

$$E = [2(h2 + h–2)/(p*q)] * 100 \quad (5)$$

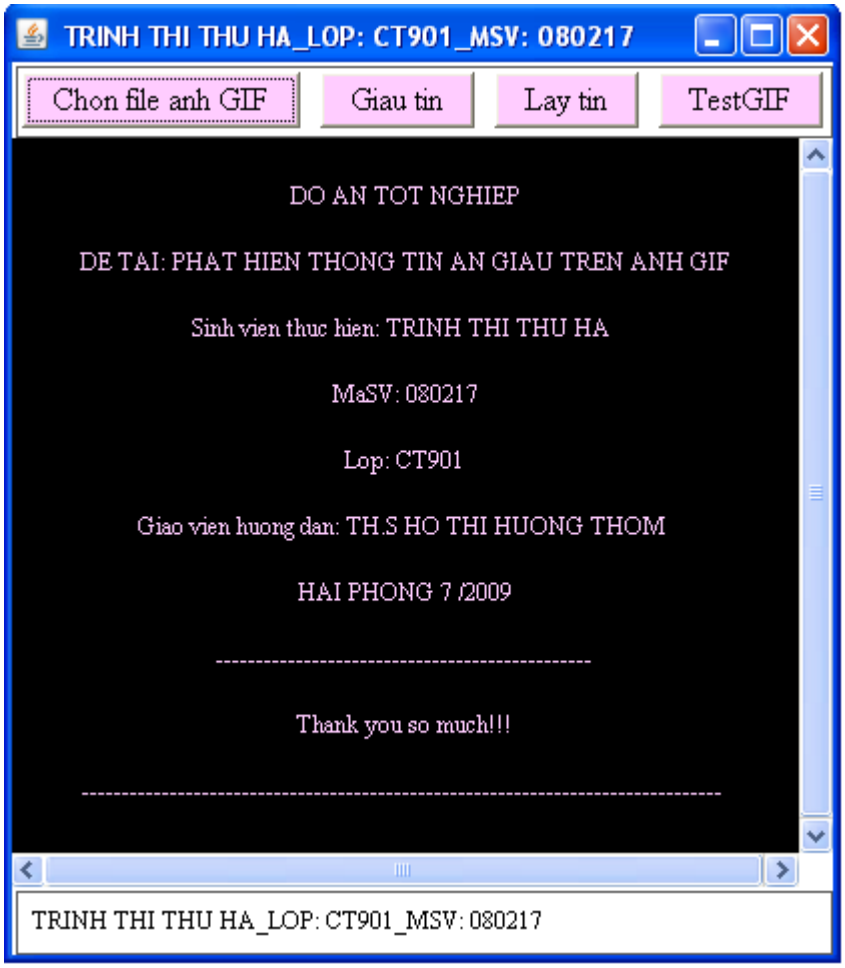
CHƯƠNG 5: KẾT QUẢ THỬ NGHIỆM

5.1 Môi trường cài đặt

Thực hiện cài đặt thuật toán phát hiện trên môi trường cài đặt là Java (bộ soạn thảo Jcreator 4.5).

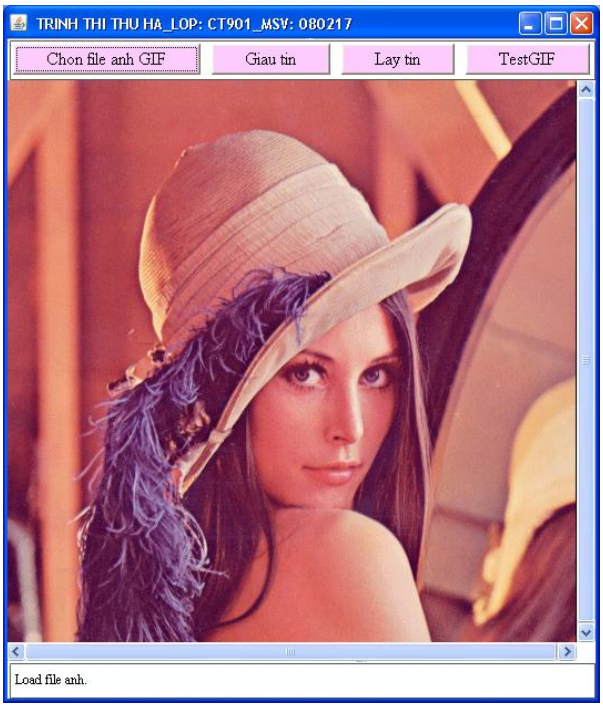
Yêu cầu cấu hình tối thiểu: Bởi JCreator khá nhẹ nên việc cài đặt và thi hành trên các thế hệ máy gần đây là vô cùng đơn giản.

Cài đặt: Dưới đây là giao diện chương trình



Hình 7. Giao diện chính

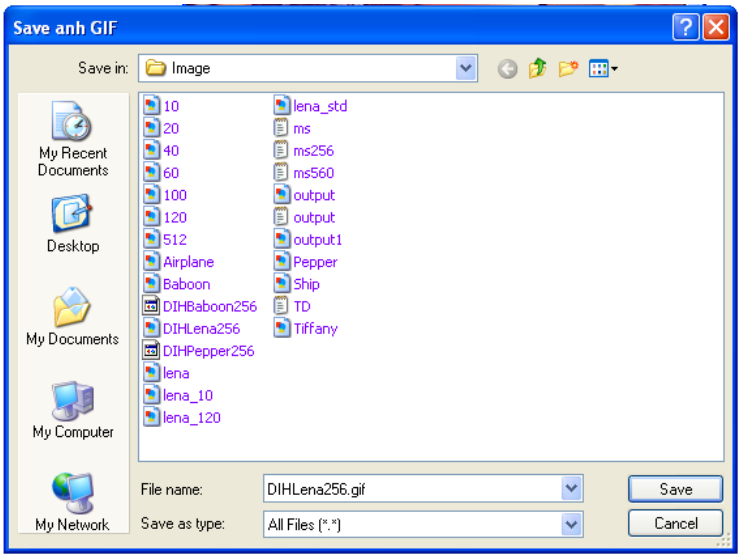
+ Giao diện Load ảnh GIF (Hình 8)



Hình 8. Ảnh gốc

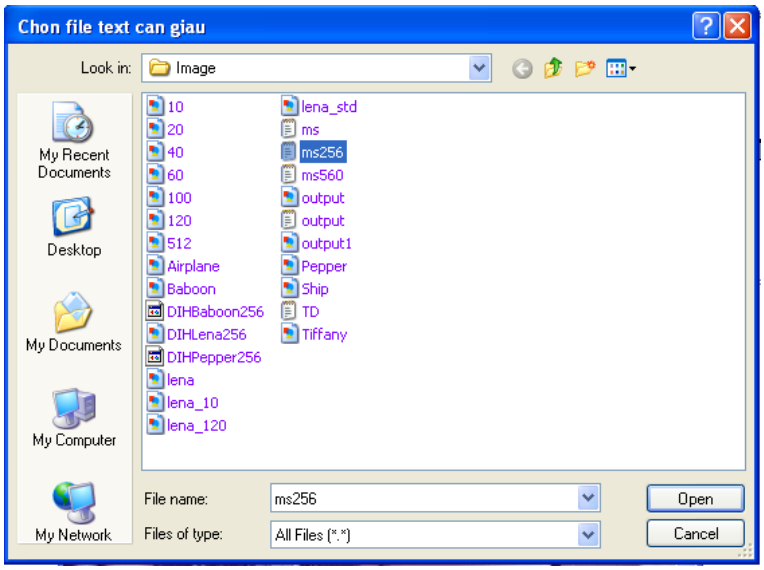
+ Giao diện giấu tin vào File ảnh GIF vừa được load

- Giao diện lưu ảnh GIF mới (Hình 9):



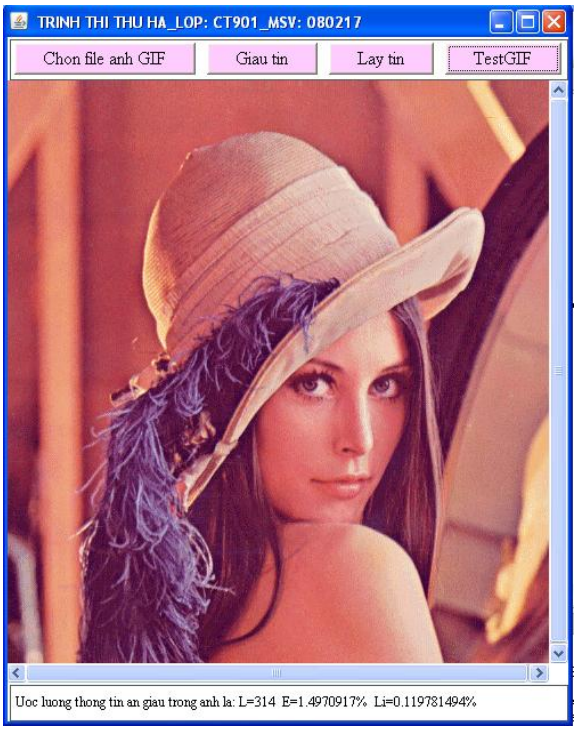
Hình 9. Lưu ảnh mới

- Giao diện chọn File thông điệp giấu (Hình 10):



Hình 10. Chọn file thông điệp

+ Giao diện TestGIF và kết quả kiểm tra ảnh đã chọn (Hình 11):



Hình 11. Ảnh có giấu tin

5.2 Thử nghiệm

Có một tập cơ sở dữ liệu ảnh gồm 100 ảnh JPEG kích cỡ 512x512 pixel được download từ [7], [8] chi tiết theo Bảng 2. Dùng Paint Shop Pro Photo X2 đề chuyển đổi sang dạng ảnh GIF. Tập ảnh này được đặt tên từ Image01.GIF đến Image100.GIF.

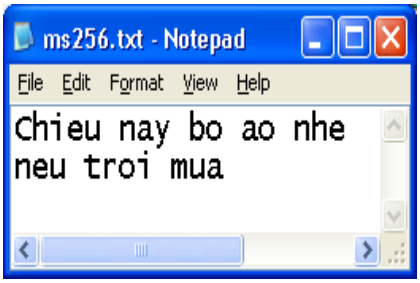
 Image01	 Image02	 Image03	 Image04	 Image05
 Image06	 Image07	 Image08	 Image09	 Image10
 Image11	 Image12	 Image13	 Image14	 Image15
 Image16	 Image17	 Image18	 Image19	 Image20
 Image21	 Image22	 Image23	 Image24	 Image25
 Image26	 Image27	 Image28	 Image29	 Image30

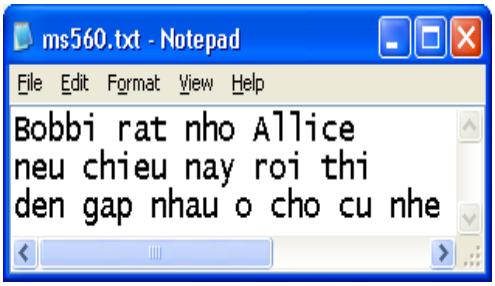
 Image31	 Image32	 Image33	 Image34	 Image35
 Image36	 Image37	 Image38	 Image39	 Image40
 Image41	 Image42	 Image43	 Image44	 Image45
 Image46	 Image47	 Image48	 Image49	 Image50
 Image51	 Image52	 Image53	 Image54	 Image55
 Image56	 Image57	 Image58	 Image59	 Image60
 Image61	 Image62	 Image63	 Image64	 Image65
 Image66	 Image67	 Image68	 Image69	 Image70

 Image71	 Image72	 Image73	 Image74	 Image75
 Image76	 Image77	 Image78	 Image79	 Image80
 Image81	 Image82	 Image83	 Image84	 Image85
 Image86	 Image87	 Image88	 Image89	 Image90
 Image91	 Image92	 Image93	 Image94	 Image95
 Image96	 Image97	 Image98	 Image99	 Image100

Bảng 2. Tập ảnh thử nghiệm

Sự dụng kỹ thuật giấu tin DIH để giấu 2 thông điệp mật gồm: 32 byte (256 bit) và 70byte (560 bit) (xem Bảng 3) vào tập cơ sở dữ liệu trên ta được kết quả như Bảng 4.





Bảng 3. Thông điệp giấu

Ảnh #	Ước lượng không giấu (ảnh gốc)	Ước lượng giấu trên 256 bit	Ước lượng giấu trên 560 bit
Image01	12710	298	542
Image02	0	256	500
Image03	0	318	562
Image04	0	240	484
Image05	12948	244	486
Image06	0	522	766
Image07	0	1070	1314
Image08	0	244	490
Image09	0	324	568
Image10	0	246	490
Image11	0	310	554
Image12	0	604	848
Image13	0	242	486
Image14	0	1474	1718
Image15	0	316	560
Image16	0	348	590
Image17	0	320	562
Image18	0	382	626
Image19	0	656	900
Image20	0	266	508
Image21	0	304	548
Image22	0	238	482
Image23	0	284	528
Image24	0	240	484
Image25	0	248	492
Image26	0	252	496
Image27	0	238	482
Image28	0	280	524

Image29	0	236	474
Image30	0	234	478
Image31	0	246	490
Image32	0	360	600
Image33	0	440	682
Image34	0	242	486
Image35	0	254	496
Image36	0	1286	1500
Image37	0	470	566
Image38	0	668	896
Image39	0	914	1148
Image40	0	3076	3294
Image41	16790	262	338
Image42	0	366	578
Image43	0	748	990
Image44	0	506	584
Image45	0	804	1032
Image46	0	1284	1490
Image47	0	2688	2920
Image48	0	252	484
Image49	0	256	500
Image50	0	256	500
Image51	14388	268	512
Image52	0	254	498
Image53	0	226	464
Image54	0	262	506
Image55	0	392	572
Image56	0	304	548
Image57	0	526	770
Image58	18592	244	488
Image59	0	236	480
Image60	13132	324	568
Image61	15720	246	490
Image62	9654	260	450
Image63	0	240	468
Image64	0	248	492
Image65	0	244	488
Image66	0	254	494
Image67	0	240	484
Image68	0	256	500

Image69	15620	274	518
Image70	0	244	488
Image71	0	242	486
Image72	0	254	498
Image73	0	258	626
Image74	0	520	756
Image75	0	242	486
Image76	0	304	548
Image77	0	920	1164
Image78	0	322	566
Image79	0	436	680
Image80	0	240	484
Image81	0	244	488
Image82	0	358	602
Image83	0	252	496
Image84	0	18	210
Image85	0	264	508
Image86	0	244	488
Image87	0	256	500
Image88	0	318	562
Image89	0	238	482
Image90	0	252	496
Image91	0	410	654
Image92	15672	258	502
Image93	0	278	522
Image94	0	260	504
Image95	17276	378	622
Image96	18686	278	522
Image97	13528	266	510
Image98	0	242	486
Image99	0	268	512
Image100	0	240	484
Ước lượng trung bình	86% Ước lượng chính xác ảnh không có giấu tin	412.14	648.38

Băng 4. Kết quả thử nghiệm

5.3 Đánh giá thuật toán

Qua kết quả thực nghiệm ở Bảng 4 ta thấy với một số ảnh nhiễu (chưa giấu tin) thì chương trình phát hiện DIHAttack vẫn có kết quả là phát hiện có giấu tin (Image01, Image05, Image41, Image51, Image58, Image60, Image61, Image62, Image69, Image92, Image95, Image96 và Image97).

Cũng từ bảng kết quả thực nghiệm trên cho thấy rằng, với cùng một ảnh gốc, số lượng bit giấu khác nhau, thì khả năng phát hiện chính xác lượng thông điệp ẩn giấu là cao hơn đối với thông điệp ngắn hơn.

Thuật toán phát hiện ảnh gốc cho kết quả chính xác là khá cao (86/100 ảnh phát hiện chính xác trong tập ảnh thử 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 trình thực hiện báo cáo tốt nghiệp, báo cáo đã được hoàn thành.

Tuy nhiên, giấu và phát hiện tin ẩn giấu vẫn là vấn đề mới mẻ, phức tạp và thời gian thực có hạn, nhất là lĩnh vực phát hiện tin ẩn giấu, cộng với khả năng và kinh nghiệm còn hạn chế nên về mặt thiết kế chương trình còn thô sơ, chưa giải quyết được vấn đề giấu và phát hiện thông tin ẩn giấu trên ảnh GIF động.

Vì vậy em rất mong nhận được sự đóng góp ý kiến của các thầy giáo cô giáo trong khoa, cũng như các thấy các cô giáo trong hội đồng phản biện để bài báo cáo tốt nghiệp của em được hoàn thiện hơn.

Em xin chân thành cảm ơn các thầy các cô!

TÀI LIỆU THAM KHẢO

[1].Lee, J., Hwang, S., Jeong, S., Yoon, K., Park, C., Ryou, J.: A DRM framework for distributing digital contents through the Internet. ETRI Journal (2003) 423–436.

[2]. Fridrich, J., Goldjan, M., Du, R.: Invertible authentication. Proc. SPIE, Security and Watermarking of Multimedia Contents (2001) 197–208.

[3]. Honsinger, C., Jone, P., Rabbani, M., Stoffel, J.: Lossless recovery of an original image containing embedded data. US Patent: 6,278,791 B1 (2001).

[4]. Ni, Z., Shi, Y., Ansari, N., Su, W.: Reversible data hiding. Proc. ISCAS (2003) 912–915.

[5]. Goldjan, M., Fridrich, J., Du, R.: Distortion-free data embedding. Proc. 4th Information Hiding Workshop (2001) 27–41.

[6]. Xuan, G., Zhu, J., Chen, J., Shi, Y., Ni, Z., Su, W.: Distortionless data hiding based on interger wavelet transform. IEE Electrionics Letters (2002) 1646–1648.

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

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

[8].USC-SIPI Image Database,

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