**CHƯƠNG 1: TỔNG QUAN VỀ LĨNH VỰC GIẤU THÔNG TIN**

**Nhóm học viên thực hiện :**

**Bùi Gia Hiếu**

**Trần Thanh Lưu**

**Lớp :** **K10 – T2**

**TRƯỜNG ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Khoa Công Nghệ**

**-------oOo-------**

**j0305257**

BÀI TẬP LỚN

**ĐỒ HỌA MÁY TÍNH**

**Trình bày : Một số thuật toán giấu tin trong ảnh**

**Hà Nội – 10/2004**

**Giới thiệu chung về giấu thông tin.**

**1. Định nghĩa :**

*“Giấu thông tin là nghệ thuật nhúng mẩu tin mật vào một vật mang tin khác. Giấu tin trong ảnh số là giấu các mẩu tin cũng là dạng số trong máy tính vào các ảnh nhị phân sao cho không bị phát hiện.”*

Thuật ngữ giấu thông tin là **steganography** (bắt nguồn từ tiếng Hy Lạp - có nghĩa là covered writing).

**2. Giấu tin và mật mã :**

Có thể coi nghệ thuật giấu tin là một nhánh của ngành mật mã với mục tiêu là nghiên cứu các phương pháp che giấu thông tin mật.

**Nghành mật mã**

(Cryptology)

**Mật mã**

(Cryptography)

**Giấu thông tin**

(Data Hiding)

**Thuỷ vân số**

(Watermarking)

**Giấu tin**

(Steganography)

***Steganography (Cover writing)***

Là nghệ thuật/khoa học/công việc truyền tin mà trong đó các thông ẩn được giấu trong thông tin chính.

***Cryptography (Secret writing)***

Là nghiên cứu phương pháp gửi thông điệp dưới hình thức khác nhau sao cho chỉ người nhận mong đợi mới bỏ đi che giấu để đọc thông điệp.Thông điệp muốn gửi đi gọi là ***bản rõ***. Thông điệp bị che giấu gọi là ***bản mã hóa***. Sau khi người nhận loại bỏ che giấu để đọc thông tin thì thông điệp không còn được bảo vệ nữa.

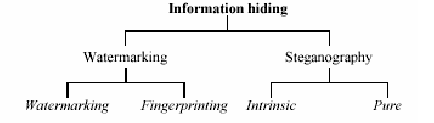
***Steganography*** giấu thông điệp trong bản rõ thay cho mã hóa thông địêp. Nó được nhúng trong dữ liệu cần bảo vệ.

Giấu tin và mật mã tuy cùng có mục đính là để đối phương không phát hiện ra tin cần giấu, tuy nhiên nó khác với mật mã ở chỗ:

+ Mật mã : Giấu đi ý nghĩa của bản thông tin.

+ Giấu tin : Giấu đi sự hiện diện của thông tin.

**3. Thủy vân số và giấu tin :**



**Watermarking** (thuỷ ấn) là lĩnh vực nghiên cứu việc nhúng các thông tin phục vụ xác thực, ví dụ như xác nhận bản quyền. Nếu thông tin giấu là một định danh duy nhất, ví dụ định danh người dùng thì khi đó người ta gọi là **Fingerprinting** (nhận dạng vân tay, điểm chỉ).

**Steganography** (giấu tin, viết phủ) là lĩnh vực nghiên cứu việc nhúng các mẩu tin mật vào một môi trường phủ. Trong quá trình giấu tin để tăng bảo mật có thể người ta dùng một khoá viết mật khi đó người ta nói về **Intrinsic Steganography** (giấu tin có xử lý). Khi đó để giải mã người dùng cũng phải có khoá viết mật đó. Chú ý rằng khoá này không phải là khoá dùng để lập mật mã mẩu tin, ví dụ nó có thể là khoá để sinh ra hàm băm phục vụ rải tin vào môi trường phủ. Ngược lại nếu không dùng khoá viết mật thì người ta chỉ giấu tin đơn thuần vào môi trường phủ thì khi đó người ta nói về **Pure Steganography** (giấu tin đơn thuần).

Xét về tính chất, thuỷ ấn giống giấu tin ở chỗ tìm cách nhúng thông tin mật vào một môi trường. Tuy nhiên xét về bản chất thì thuỷ ấn có những nét khác ở một số điểm:

+ Mục tiêu của thuỷ ấn là nhúng thông tin không lớn thường là biểu tượng, chữ ký hay các đánh dấu khác vào môi trường phủ nhằm phục vụ việc xác nhận bản quyền

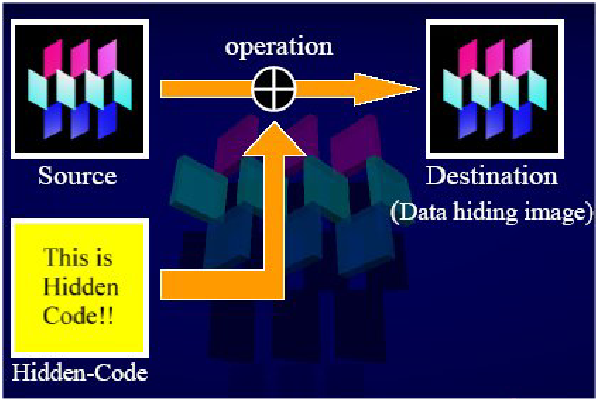
+ Khác với giấu tin ở chỗ, giấu tin sau đó cần tách lại tin còn thuỷ ấn tìm cách biến tin giấu thành một thuộc tính của vật mang

+ Chỉ tiêu quan trọng nhất của một thuỷ ấn là tính bền vững, của giấu tin là dung lượng bản tin được giấu

+ Điểm khác nữa giữa thuỷ ấn và giấu tin là thuỷ ấn có thể vô hình hoặc hữu trên ảnh mang.

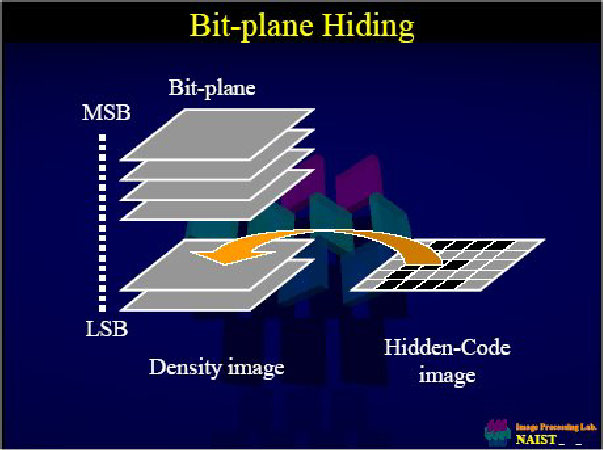
**4. Giấu tin trong ảnh số:**

Giấu tin trong ảnh được thực hiện bằng cách thay thế một vài thông tin ít quan trọng nhất của ảnh gốc. Đối với ảnh màu: Sử dụng các bit thấp (least-significant bit -LSB) của mỗi pixel để giấu thông tin.*Thí dụ, ảnh Kodak Photo CD kích thước 2048x3072x24 bit màu RGB có thể giấu tới 2.36 Mb bit thông tin.* Ảnh 2 màu đen/trắng (ảnh nhị phân) (trang fax, mã vạch...) sẽ khó khăn hơn vì khi thay đổi 1 pixel ảnh thì mắt người dễ nhận biết. Ảnh JPEG hay MP3 của âm thanh: Phức tạp hơn. Phải tìm ra các “lỗ hổng” sao cho chất lượng ảnh ít bị ảnh hưởng khi thực hiện thuật toán nén và giải nén ảnh.

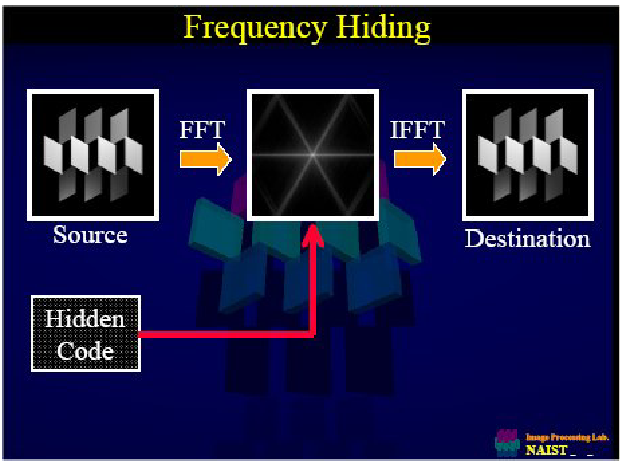


**Giấu thông tin vào trong ảnh**

**Giải pháp giấu tin trong ảnh**



**Giấu tim trong miền quan sát**

****

**Giấu tin trong miền tần số**

**5. Kỹ thuật chung giấu thông tin trong ảnh :**

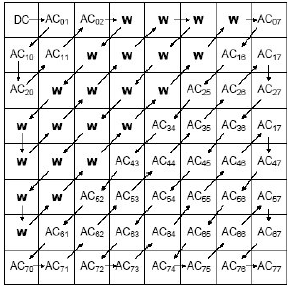
*Chọn vị trí dấu thông tin :*

* vị trí ngẫu nhiên trong ảnh gốc
* vùng tần số trung bình hay tần số cao (biên ảnh)

- Miền ảnh có tần số càng cao thì mắt người càng kém phân biệt sự thay đổi

- Chú ý: trong nén ảnh mất mát thông tin thường loại bỏ miền ảnh có tần số cao

*Chọn miền giấu thông tin :*



* Dấu thông tin trong miền quan sát: Thực

hiện trực tiếp trên ma trận ảnh

* Dấu thông tin trong miền DFT, DCT

hay DWT.

* Sau đó biến đổi ngược lại miền quan sát

*Chọn kiểu chèn thông tin giấu :*

* Cộng trực tiếp thông tin vào miền giá trị

của ảnh.

* Thay đổi cách biểu diễn giá trị ảnh theo

cách biểu diễn của thông tin ẩn

*Chọn kiểu tách thông tin ẩn :*

* Chọn kiểu tách thông tin ẩn
* Tách thông tin ẩn tương tự tách tín hiệu

nhiễu.

* Các bước tách thông tin ẩn là các bước ngược lại của tiến trình

chèn thông tin ẩn

**6. Các thành phần chính của một hệ giấu tin trong ảnh số**

Các thành phần chính của một hệ giấu tin trong ảnh số gồm :

* **Bản tin mật** (Secret Message): có thể là văn bản hoặc tệp ảnh hay bất kỳ một tệp nhị phân nào, vì quá trình xử lý chúng ta đều chuyển chúng thành chuỗi các bit.
* **Ảnh phủ** (hay ảnh gốc) (Cover Data): là ảnh được dùng để làm môi trường nhúng tin mật.
* **Khoá bí mật K** (Key): khoá viết mật tham gia vào quá trình giấu tin để tăng tính bảo mật
* **Bộ nhúng thông tin** (Embedding Algorithm)**:** Những chương trình, thuật toán nhúng tin.
* **Ảnh mang** (Stego Data): là ảnh sau khi đã nhúng tin mật vào đó
* **Kiểm định** (Control) : Kiểm tra thông tin sau khi được giải mã.

Mô hình của kỹ thuật giấu tin cơ bản được mô tả theo hai hình vẽ sau :

Phương tiện chứa

(audio, ảnh, video)

Bộ nhúng thông tin

Bản tin mật

Khoá

Phương tiện chứa đã giấu tin



**Phân phối**

**Hình 1: Lược đồ chung cho quá trình giấu tin**

Hình vẽ trên biểu diễn quá trình giấu tin cơ bản. Phương tiện chứa bao gồm các đối tượng được dùng làm môi trường giấu tin như : text, audio, video, ảnh, bản tin mật là một lượng thông tin mang một ý nghĩa nào đó như ảnh, logo, đoạn van bản… tuỳ thuộc vào mục đích của người sử dụng . Thông tin sẽ được giấu vào trong phương tiện chứa nhờ một bộ nhúng, bộ nhúng là những chương trình, triển khai các thuật toán để giấu tin và được thực hiện với một khoá bí mật giống như các hệ mật mã cổ điển. Sau khi giấu tin, ta thu được phương tiện chứa bản tin đã giấu và phân phối sử dụng trên mạng.

Phương tiện chứa

(audio, ảnh, video)

Bộ giải mã tin

Bản tin mật

Khoá

Phương tiện chứa đã giấu tin



Kiểm định mã tin

**Hình 2: Lược đồ cho quá trình giải mã**

**Hình 1: Lược đồ chung cho quá trình giải mã**

**Hình 1: Lược đồ chung cho quá trình giấu tin**

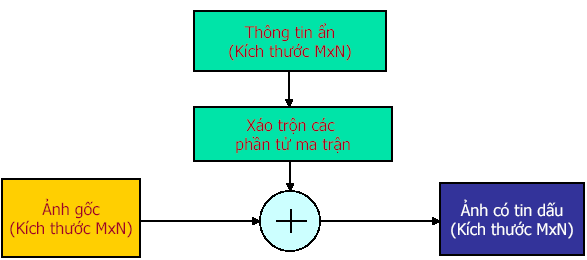
Hình vẽ trên chỉ ra các công việc giải mã thông tin đã giấu. Sau khi nhận được đối tượng phương tiện chứa có giấu thông tin, quá trình giải mã được thực hiện thông qua bộ nhúng thông tin cùng với khoá của quá trình nhúng. Kết quả thu được gồm phương tiện chứa gốc và bản tin mật đã được giấu. Bước tiếp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

Sơ đồ phân loại trên (hình 1,2) được Fabien A. P. Petitcolas đề xuất năm 1999.

**7. Giải pháp giấu tin trong ảnh số :**

**Giải pháp 1 :** *Giấu tin vào miền quan sát.*

Phương pháp này đơn giản nhất vì không yêu cầu biến đổi sang miền tần số. Thông tin ẩn chèn trực tiếp vào pixel ảnh. Thông tin ẩn được trải đều trên toàn bộ mặt ảnh. Ma trận ảnh gốc và ma trận dấu ẩn phải có cùng kích thước.



+ Chèn thông tin vào miền quan sát :

- Tạo dãy số nguyên S liên tục có thứ tự ngẫu nhiên theo luật sinh xác định trước.

- Cho trước a, bvà m, xn là phần tử sinh, xn+1 là phần tử ngẫu nhiên tạo ra, theo Lehmer(1949) :

xn+1 = (a\*xn + b) mod m

- Miền giá trị của dãy là {1..MxN}

- Chuyển đổi ma trận giấu tin 2 chiều thành dãy số W.

- Xáo trộn dãy W như sau : W[i] = W[Si]

- Chuyển đổi ngược dãy W về ma trận MxN

- Cộng ma trận thông tin ẩn đã xáo trộn với ảnh gốc để có ảnh chứa thông tin ẩn.

+ Tách tin giấu trong miền quan sát :

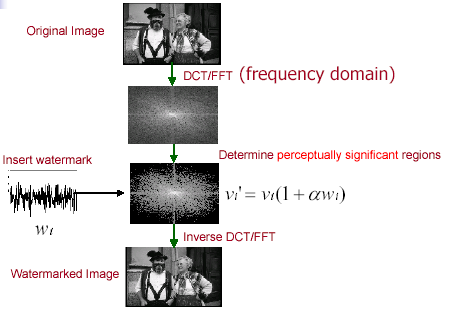
- Gọi ảnh gốc là I và ảnh có thông tin ảnh là I’.Thực hiện phép trừ các phần tử tương ứng của I’ cho I để có thông tin ẩn (ma trận W).

- Chuyển ma trận W thành dãy số W\*

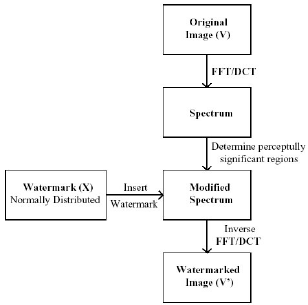
- Tạo dãy số nguyên S liên tục có thứ tự ngẫu nhiên theo luật sinh xác định ở bước chèn tin ẩn

- Sắp xếp lại dãy W\*: W\*[Si] =W[i]

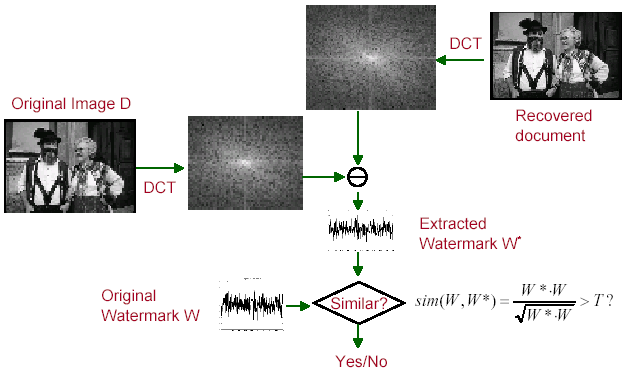
- Chuyển đổi dãy W\* thành ma trận hai chiều để có thông tin ẩn.

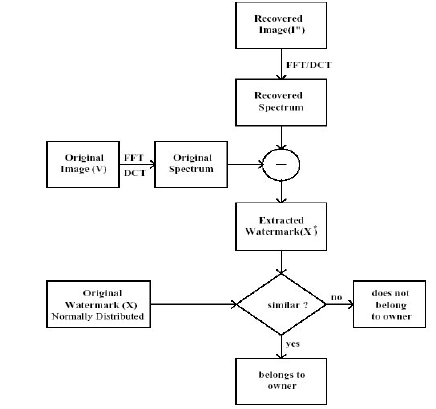
**Giải pháp 2 :***Giấu tin trong miền tần số*

+ Chèn thông tin vào miền tần số :



+ Tách thông tin trong miền tần số:



****

**8. Các tính chất giấu tin trong ảnh số:**

***a. Tính vô hình (Độ tin cậy):***

Giấu tin trong ảnh sẽ làm biến đổi ảnh mang. Tính vô hình thể hiện mức độ biến đổi ảnh mang. Một hương pháp tốt sẽ làm cho thông tin mật trở nên vô hình trên ảnh mang, người dùng không thể phát hiện trong đó có ẩn chứa thông tin.

**b. Khả năng chống giả mạo :**

Vì mục đích của một phương pháp giấu tin là chuyển đi thông tin mật. Nếu không thể do thám tin mật thì kẻ địch cũng sẽ cố tìm cách làm sai lạc thông tin mật, làm giả mạo thông tin để gây bất lợi cho đối phương. Một phương pháp giấu tin tốt sẽ đảm bảo tin mật không bị tấn công một cách có chủ đích trên cơ sở những hiểu biết đầy đủ về thuật toán nhúng tin (nhưng không biết khoá) và có ảnh mang. Đối

với lĩnh vực thuỷ ấn số thì khả năng chống giả mạo là đặc tính vô cùng quan trọng. Vì có như vậy mới bảo vệ được bản quyền, chứng minh tính pháp lý của sản phẩm.

**c. Dung lượng giấu :**

**Cân nhắc giữa chất lượng, dung lượng và tính bền vững .**

**Quality**

**Capacity**

**Robustness**

|  |
| --- |
| Dung lượng giấu được tính bằng tỷ lệ  của lượng tin giấu so với kích thước ảnh. Vì  tin mật được gửi cùng với ảnh mang qua  mạng nên đây cũng là một chỉ tiêu quan trọng.  Các phương pháp đều cố làm sao giấu được  nhiều tin trong khi vẫn giữ được bí mật. Tuy  nhiên trong thực tế người ta luôn phải cân  nhắc giữa dung lượng và các chỉ tiêu khác  như tính vô hình, tính ổn định. |

**d. Tính bền vững :**

Sau khi giấu tin vào ảnh mang, bản thân ảnh mang có thể phải qua các khâu biến đổi khác nhau như lọc tuyến tính, lọc phi tuyến, thêm nhiễu, làm sắc nét, mờ nhạt, quay, nén mất dữ liệu. Tính bền vững là thước đo sự nguyên vẹn của thông tin mật sau những biến đổi như vậy.

**e. Độ phức tạp của thuật toán :**

Chỉ tiêu độ phức tạp trong mã hoá và giải mã cũng là một yếu tố quan trọng trong đánh giá các phương pháp giấu tin trong ảnh. Yêu cầu về độ phức tạp tính toán phụ thuộc vào từng ứng dụng. Ví dụ một ứng dụng tạo thuỷ ấn để đánh dấu bản quyền cần phải có độ phức tạp tính toán cao thì mới đảm bảo chịu được sự tấn công của nhiều tin tặc nhằm phá huỷ thuỷ ấn.

**9. Ứng dụng của giấu tin trong ảnh số:**

Giấu tin trong ảnh số ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực.Các ứng dụng có sử dụng đến giấu tin trong ảnh số có thể là : **Bảo vệ bản quyền tác giả** (Copyright Protection), **Điểm chỉ số** (fingerprinting), **Gán nhãn** (Labelling), **Giấu thông tin mật** (Steganography)…

* **Bảo vệ bản quyền :** Là ứng dụng cơ bản nhất của kỹ thuật thuỷ vân số (watermarking) - một dạng của phương pháp giấu tin. Một thông tin nào đó mang ý nghĩa sở hữu quyền tác giả (người ta gọi nó là thuỷ vân - watermark) sẽ được nhúng vào trong các sản phẩm, thuỷ vân đó chỉ có một mình người chủ sở hữu hợp pháp các sản phẩm đó có và được dùng làm minh chứng cho bản quyền sản phẩm. . Giả sử có một thành phẩm dữ liệu dạng  đa phương tiện như ảnh, âm thanh, video cần được lưu thông trên mạng. Để bảo vệ các sản phẩm chống lại hành vi lấy cắp hoặc làm nhái cần phải có một kỹ thuật để “dán tem bản quyền” vào sản phẩm này. Việc dán tem hay chính là việc nhúng thuỷ vân cần phải đảm bảo không để lại một ảnh hưởng lớn nào đến việc cảm nhận sản phẩm. Yêu cầu kỹ thuật đối với ứng dụng này là thuỷ vân phải tồn tại bền vững cùng với sản phẩm, muốn bỏ thuỷ vân này mà không được phép của người chủ sở hữu thì chỉ còn cách là phá huỷ sản phẩm.
* **Điểm chỉ số :** Mục tiêu của điểm chỉ số là để chuyển thông tin về người nhận (chứ không phải chủ sở hữu) sản phẩm phương tiện số nhằm xác định đây là bản sao duy nhất của sản phẩm. Về mặt ý nghĩa điểm chỉ số tương tự như số xê ri của phần mềm
* **Gán nhãn :** Tiêu đề, chú giải và nhãn thời gian cũng như các minh hoạ khác có thể được nhúng vào ảnh, ví dụ đính tên người lên ảnh của họ hoặc đính tên vùng địa phương lên bảng đồ. Khi đó nếu sao chép ảnh thì cũng sẽ sao chép cả các dữ liệu nhúng trong nó. Và chỉ có chủ sở hữu của tác phẩm, người có được khoá mật (Stego-Key) mới có thể tách ra và xem các chú giải này. Trong một cơ sở dữ liệu ảnh, người ta có thể nhúng các từ khoá để các động cơ tìm kiếm có thể tìm nhanh một bức ảnh. Nếu ảnh là một khung ảnh cho cả một đoạn phim, người ta có thể gán cả thời điểm diễn ra sự kiện (timing) để đồng bộ hình ảnh với âm thanh. Người ta cũng có thể gán số lần ảnh được xem để tính tiền thanh toán theo số lần xem.
* **Giấu thông tin mật :** Trong nhiều trường hợp sử dụng mật mã có thể gây ra sự chú ý ngoài mong muốn. Ngoài ra việc sử dụng công nghệ mã hoá có thể bị hạn chế Một số kỹ thuật giấu tin trong ảnh mầu hoặc cấm sử dụng. Ngược lại việc giấu tin trong môi trường nào đó rồi gửi đi trên mạng ít gây sự chú ý. Có thể dùng nó để gửi đi một bí mật thương mại, một bản vẽ hoặc các thông tin nhạy cảm khác.
* **…**

**CHƯƠNG 2 : MỘT SỐ KỸ THUẬT GIẤU TIN TRONG ẢNH TĨNH**

Trong chương này, chúng ta tìm hiểu một số kỹ thuật xử lý điểm ảnh sử dụng trong kỹ thuật giấu tin và một số kỹ thuật giấu tin đang được sử dụng rộng rãi :

* Một số kỹ thuật xử lý ảnh **BITMAP**
* Mô hình của quá trình giấu tin trong ảnh số
* Một số kỹ thuật giấu tin trong ảnh số

**I. Một số kỹ thuật xử lý ảnh BITMAP trong kỹ thuật giấu tin**

Các kỹ thuật xử lý điểm ảnh được mô tả và minh hoạ bằng các câu lệnh của ngôn ngữ C.

**1). Đọc header file ảnh BITMAP**

Đọc được file dữ liệu ảnh, chúng ta cần đọc được header ảnh để biết thông tin ảnh và đọc bảng màu của ảnh (nếu có).

Ta khai báo cấu trúc header của ảnh BITMAP như sau, cấu trúc này tương ứng với những thành phần có trong header của ảnh bitmap như đã giới thiệu ở chương trước, như sau :

***Typedef struct{***

***unsigned int ID ;***

***long File\_Size ;// kich thuoc file anh***

***long Reserved ;***

***long OffsetBit ;// kich thuoc header***

***long Isize, Width, Height ;***

***unsigned int Planes,biCount ;***

***long Compression, ImageSize ;***

***long XpelsPerMeter,YpelsPerMeter ;***

***long ColorsUsed, ColorsImportant***

***} BitmapHeader;***

Khi đọc header của ảnh , ta chỉ việc dùng câu lệnh đọc một cấu trúc fread :

***fread (&bmh, sizeof(bmh), 1, picture);***

Trong đó :

+ **bmh** là một biến có kiểu là kiểu cấu trúc header đã định nghĩa.

+ **picture** là biến file ảnh.

Sau khi đọc xong header của ảnh thì những thông tin về tính chất ảnh sau đây là cần thiết :

+ **bmh.Width** : Độ rộng của ảnh .

+ **bmh.Height** : Chiều cao của ảnh .

+ **bmh.bitCount** : Số bít cho một điểm ảnh (nhờ thông số này mà ta biết ảnh có bảng màu hay không) để đọc dữ liệu ảnh tiếp theo.

**2). Đọc bảng màu của ảnh :**

Chỉ có ảnh mà mỗi điểm ảnh được biểu diễn bởi số không lớn hơn 8 bits thì có bảng màu (pallete), trong trường hợp đó ta khai báo một cấu trúc màu và đọc bảng màu như sau :

***typedef struct {signed char Red, Green, Blue, Reserved ;}TRGB;***

Cáu trúc bảng màu gồm 4 thành phần : Red, Green, Blue và Reserved, mỗi thành phần là một byte. Khi đó kích thước của bảng màu được tính bằng công thức :

***TableSize = power2(bmh.biCount)\*sizeof(TRGB);***

Với power2(x) là hàm tính 2x , hàm power2(bmh.biCount) sẽ cho ta số màu để biểu diễn điểm ảnh. Chẳng hạn , nếu mỗi điểm ảnh được biểu diễn bằng 8 bits thì số màu để biểu diễn điểm ảnh sẽ là 28 = 256 màu. Mỗi màu lại được biểu diễn bởi một byte như cấu trúc ở trên nên kích thước bảng màu là : 256\*4 =1024 byte.

Tiếp theo , ta khai báo một mảng các màu của ảnh , nếu ảnh có 256 màu thì kích thước mảng là 256, nếu có 16 màu thì mảng có 16 phần tử hay nếu là ảnh đen trắng thì mảng có 2 phần tử. Tổng quát , ta khai báo mảng như sau :

***TRGB \* color;***

Cuối cùng, ta đọc bảng màu của ảnh :

***TableSize = power2(bmh.biCount)\*sizeof(TRGB);***

***fread(color,TableSize,1,Picture);***

**3). Chia ảnh thành các khối m x n .**

Trong nhiều kỹ thuật giấu tin trong ảnh, ảnh ban đầu thường được chia nhỏ thành các khối có kích thước m x n . Đây chính là phần xử lý dữ liệu ảnh sau khi ta đã đọc đuợc header và bảng màu.

Ta định nghĩa khối ảnh m x n là một ma trận hai chiều kích thước m x n, mỗi phần tử của mảng hai chiều có giá trị tương ứng là giá trị các điểm ảnh .

**Ảnh được chia thành các khối m\*n.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Các kỹ thuật giấu tin thường chia nhỏ ảnh ra thành các khối như trên hình vẽ, sau đó giấu tin vào các khối, cuối cùng ghép các khối lạ với nhau để thu được ảnh ban đầu (đã giấu thông tin).

**4). Kỹ thuật ghép ảnh mới :**

Sau khi giấu dữ liệu xong ta ghép ảnh mới theo thứ tự sau:

* Ghi header ảnh mới lấy từ ảnh cũ.
* Ghi bảng màu vào ảnh mới sang ảnh cũ
* Đưa dữ liệu ảnh mới (đã giấu tin) vào trong ảnh.

**5). Kỹ thuật xử lý điểm ảnh**

Xử lý các điểm ảnh là kỹ thuật được sử dụng thường xuyên trong các kỹ thuật giấu tin trong ảnh. Các giá trị điểm ảnh được lấy ra rồi biến đổi theo kỹ thuật giấu tin. Tuy nhiên, miền giá trị của các điểm ảnh lại khác nhau phụ thuộc vào các loại ảnh, chính vì thế ta cần dùng đến *kỹ thuật tách thông tin* từ giá trị điểm ảnh .

**Bảng : Số bít biểu diễn một điểm ảnh đối với các ảnh khác nhau**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kiểu ảnh** | Đen trắng | ảnh 16 màu | ảnh 256 màu | ảnh High color | ảnh True color |
| **Số bit/pixel** | **1** | **4** | **8** | **16** | **24** |

Kỹ thuật này được sử dụng nhiều trong kỹ thuật giấu tin, sử dụng các bit ít quan trọng nhất của điểm ảnh (gọi là **LSB** – Least Significiant Bit). Kỹ thuật LSB là kỹ thuật sử dụng các bít ít quan trọng về thị giác nhất trong các bít mang giá trị điểm ảnh để giấu tin. Ví dụ, với ảnh 256 màu thì bít cuối cùng trong 8 bít biểu diễn một điểm ảnh được coi là bít ít quan trọng nhất theo nghĩa là nếu thay đổi bít này thì ảnh hưởng ít nhất đến cảm nhận của mắt người về điểm ảnh. Hay đối với ảnh 16 bít thì 15 bít là biểu diễn màu RGB của điểm ảnh của điểm ảnh, còn bít cuối cùng không dùng đến thì ta sẽ tách bít này ra ở mỗi điểm ảnh để giấu tin. Như vậy, kỹ thuật tách bít trong xử lý điểm ảnh được sử dụng rất nhiều trong quy trình giấu tin, sau đây ta sẽ khảo sát một số kỹ thuật tách bít ít quan trọng trên một số loại ảnh phổ biến:

**Tách bít cuối cùng trong 8 bít biểu diễn mỗi điểm ảnh của ảnh 256 màu**

|  |  |  |
| --- | --- | --- |
| **1001110 0** | **1001010 1** | **110001 0** |

**Hình 1.1: Bít LSB của mỗi điểm ảnh trong ảnh 256 màu.**

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

Ta thực hiện tách như sau:

// c là một giá trị điểm ảnh

c = getc(picture);

c1 = c & 1;

Sau hai câu lệnh này thì c1 sẽ mang giá trị là 0 hoặc 1 ứng với bít cuối cùng của biến c. Thật vậy, c là giá trị của một điểm ảnh nào đó, giả sử c = 01100101 (đối với ảnh 8 bit), c1 =c& 1; đây là phép toán nhân theo bít thông thường :

01100101

& 00000001 (giá trị của 1 lưu trên một byte – 8 bit)

c1 = 00000001 (c1 bằng giá trị của bít cuối cùng)

**6). Kỹ thuật đảo bít**

Giả sử 1 bít có giá trị là b, kỹ thuật đảo bít được thực hiện qua phép toán là 1 – b.

**7). Kỹ thuật tính hệ số phân bố bít**

Giả sử ta có một ma trận điểm ảnh 0,1 cỡ m\*n (D)

Ta định nghĩa : **hệ số phân bố bít** D là đại lượng đặc trưng cho mức độ rời rạc của các bít 0,1 trên D và được tính theo công thức:

**D = Dh + Dv + Dc + Da**

Trong đó :

Dh  là hệ số phân bố bít theo chiều ngang :

**Dh =(pi,j ≠ pi,j+1)**

Dv là hệ số phân bố theo chiều dọc:

**Dv = (pi,j ≠ pi+1,j)**

Dc là hệ số phân bố bít theo đường chéo 1:

**Dc =(pi,j ≠ pi-1,j+1)**

Da là hệ số phân bố bít theo đường chéo 2:

**Da =(pi,j ≠ pi+1,j+1)**

Với hàm số I(Pi,j ⊕ Pi’,j’) là hàm số cho hai giá trị : là 1 nếu phần tử thứ i, j khác với phần tử thứ i’ ,j’ và bằng 0 nếu phần tử thứi, j bằng với phần tử thứ i’,j’.

Ví dụ : Cho một ma trận nhị phân D 4\*4 như sau :

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |

Khi đó ta sẽ có các hệ số phân bố bít theo các chiều là :

Dh = 2 + 2 + 3 + 2 = 9

Dv = 2 + 1 + 1 + 1 = 6

Dc = 1 + 2 + 1 +1 +1 = 7

Da = 1 + 1+ 2 + 1 = 5

Suy ra hệ số phân bố bit 0,1 trên D là : D = Dh + Dv + Dc + Da = 9 + 6 + 7 + 5 = 27

**II. Mô hình của quá trình giấu tin và giải mã trong ảnh :**

Ảnh gốc

File dữ liệu ra các bit LSB

Thông tin cần giấu

File nhị phân chứa thông tin cần giấu

Các khối bit

Các khối đã giấu thông tin

File dữ liệu chứa các Bit LSB

Ảnh đã được giấu tin

Tách bit LSB

Tách khối

Giấu tin

Mã hoá

Giấu tin

Ghép khối

**Hình 2.1 : Quá trình giấu tin**

Ảnh chứa thông tin giấu

File dữ liệu các bit LSB

Các khối bit

Thông tin được giấu

File thông tin giấu ban đầu

File thông tin giấu ban đầu

Tách bit LSB

Tách khối

Lấy tin

Ghép khối

Giải mã

**Hình 2.3 : Quá trình giải mã thông tin**

**III. Một số kỹ thuật giấu tin trong ảnh đen trắng và ảnh màu .**

Các kỹ thuật giấu tin dựa trên giải pháp giấu tin vào miền quan sát :

**Kỹ thuật 1 - Giấu tin đơn giản :**

**1).Ý tưởng :**

Đây có thể là kỹ thuật đơn giản nhất trong các kỹ thuật giấu tin . Ý tưởng cơ bản của thuật toán là chia một ảnh thành các khối nhỏ và với mỗi khối nhỏ đó sẽ giấu được một bít thông tin. Các khối ảnh này chỉ gồm các bít 0 và 1, và bít thông tin được giấu thì chỉ có giá trị 0 hoặc 1. Vì vậy ý tưởng của kỹ thuật này là sử dụng tính chẵn lẻ của tổng số bit 1 trong khối để quy định bít được giấu là 0 hay 1.

**2). Thuật toán giấu tin :**

**Input :**

* **Một file ảnh Bitmap đen trắng I**
* **Một file thông tin cần giấu F**

**Output :**

* **Một file ảnh đã giấu thông tin I’**
* **Một khoá để giải tin K**

**Cách thức thực hiện :**

*Các bước tiền xử lý trước khi thực hiện thuật toán :*

* Chuyển file thông tin cần giấu F sang dạng nhị phân, bởi thuật toán sẽ giấu từng bít thông tin vào trong ảnh. Quá trình giải tin là biến đổi ngược, thu được file thông tin cần giấu.
* Đọc header của ảnh để lấy thông tin ảnh. Sau đó, đọc toàn bộ dữ liệu ảnh vào một mảng hai chiều để sử dụng cho việc giấu tin.

*Quá trình thực hiện giấu tin :*

* Giả sử ảnh gốc ban đầu có kích thước là (M \* N), được chia thành các khối nhỏ (m\*n), khi đó tổng các khối nhỏ sẽ là (M\*N)/(m\*n) khối. Vì ảnh sử dụng làm môi trưòng là ảnh đen trằng nên mỗi khối là một ma trận hai chiều chứa các giá 0,1.

**Hình 3.1 : Một ví dụ về dữ liệu ảnh được chia thành các khối kích thước 4\*4 :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *1* | *0* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *1* | *0* |  |  |  |
| *1* | *0* | *0* | *0* | *1* | *1* | *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *1* | *1* | *0* | *0* | *1* | *1* |  |  |  |
| *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *1* | *0* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *1* | *0* |  |  |  |
| *1* | *0* | *0* | *0* | *1* | *1* | *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *1* | *1* | *0* | *0* | *1* | *1* |  |  |  |
| *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *1* | *0* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *1* | *0* |  |  |  |
| *1* | *0* | *0* | *0* | *1* | *1* | *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *1* | *1* | *0* | *0* | *1* | *1* |  |  |  |
| *0* | *1* | *0* | *1* | *1* | *1* | *1* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *1* | *0* | *0* |  |  |  |
| *1* | *1* | *0* | *0* | *1* | *1* | *1* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *1* | *0* | *0* |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Sau khi phân tích thành các khối nhỏ , ta chọn các khối để giấu tin, ta có thể chọn ngẫu nhiên các khối nhưng để cho đơn giản ta coi như các khối được chọn tuần tự từ khối đầu tiên cho đến khi hết thông tin cần giấu.

Mỗi khối nhỏ này sẽ được dùng để giấu một bit thông tin theo quy tắc sau: Sau khi giấu thì tổng số bit 1 trong khối và bit thông tin cần giấu sẽ có cùng tính chẵn lẻ. Nghĩa là , nếu giấu bít 1 vào trong một khối thì khối đó phải thoả mãn tính chất tổng số bit 1 trong khối là số lẻ. Ngược lại, cần giấu bit 0 vào một khối thì khối đó phải thoả mãn tính chất tổng số bít 1 trong khối là chẵn.

Như vậy, mỗi lần giấu 1 bit ta có hai trường hợp xảy ra sau đây :

* Khối đã thoả mãn tính chất để giấu một bít thông tin : Chẳng hạn như khi cần giấu bít 1 thì tổng số bit 1 đã là lẻ rồi, hoặc khi cần giấu bit 0 vào khối thì khối đó đã có tổng số bít 1 là chẵn. Trong những trường hợp như thế, ta không cần thay đổi và xem như một bit thông tin đã được giấu .
* Trong trường hợp ngược lại, tính chất của khối chưa thoả mãn yêu cầu để giấu thông tin, nghĩa là khi cần giấu bit 1 vào trong khối thì tổng số bít 1 có sẵn là chẵn, hoặc khi cần giấu bít 0 thì tổng số bít 1 trong khối lại đang lẻ. Khi đó, ta sẽ chọn thay đổi một bít bất kì trong khối, từ 0 sang 1(hoặc ngược lại), để nhận được tổng số bit 1 trong khối thoả mãn yêu cầu.

Giả sử ta phải giấu bit 1 vào khối B có kích thước 4\*4 như sau :

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

Trong trường hợp trên, khối bít B có tổng số 8 bít 1, như vậy chưa thoả mãn yêu cầu giấu bit 1 vào trong khối , như vậy ta cần thay đổi bằng cách chọn một bít bất kỳ và đổi từ 0 sang 1 hoặc từ 1 sang 0. Giả sử ta đổi lại giá trị một bit trong khối bít đó như sau :

Bít thay đổi từ 0 sang 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **0** | **1** | **1** |
| **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** |

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **0** | **1** | **1** |
| **0** | **1** | **0** | **0** |
| **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** |

Ngược lại nếu thực hịên giấu bít 0 vào trog khối bít trên thì ta không phải thay đổi đối với khối bít ban đầu , do khối bít ban đầu đã thỏa mãn tính chất để giấu bit 0.

Như vậy, mỗi lần giấu một bit ta lại lấy một khối để thực hiện giấu bít theo quy tắc trên đến khi hết lượng thông tin cần giấu. Sau khi giấu xong ta được một ma trận hai chiều dữ liệu ảnh mới. Bước tiếp theo là xây dựng ảnh mang tin bằng cách : Ghi header ảnh gốc đã đọc lúc đầu vào ảnh mới , ghi bảng màu đã đọc vào file ảnh mới, cuối cùng ghi dữ liệu ảnh mới sau khi đã giấu tin vào ảnh ta sẽ thu được ảnh mới sau khi giấu tin.

Trong thuật toán này, khoá đơn giản chỉ là kích thước của khối , nếu biết kích thước của khối thì dễ dàng giải mã tin theo quy tắc sau :

*Quá trình giải mã tin :*

Sau khi nhận được ảnh đã giấu tin, quá trình giải mã tin sẽ được thực hiện theo các bước sau đây :

* Đọc header và bảng màu của ảnh để biết các thông tin về ảnh.
* Đưa phần dữ liệu ảnh vào mảng hai chiều .

Các bước này giống với quá trình giấu tin. Sau khi đã có được dữ liệu ảnh , ta chia ảnh thành các khối có kích thước giống kích thước khối khi thực hiện giấu, đây chính là khoá để giải mã. Chọn ra các khối đã giấu và giải tin theo quy tắc : đếm số bít 1 trong khối, nếu tổng số bít 1 là lẻ thì thì thu được bit 1, ngược lại thu được bit 0. Cứ tiếp tục cho đến khi hết các khối đã giấu tin.

Như vậy, sau khi hết các khối đã giấu tin, ta thu được một chuỗi bít đã đem giấu . Bước tiếp theo ta chuyển từ file nhị phân sang file văn bản .

*Phân tích thuật toán*

Đây là thuật toán rất đơn giản thực hiện một cách thức giấu tin trong ảnh, sau khi nghiên cứu thuật toán này chúng ta có thể đưa ra một số bình luận và đánh giá như sau :

* Việc chọn kích thước khối để giấu tin tuỳ thuộc vào kích thước ảnh và lượng thông tin cần giấu sao cho giấu dàn trải trên toàn ảnh. Ví dụ, nếu ta có một ảnh có kích thước 512\*512 pixel và có một lượng thông tin cần giấu là 100 ký tự. Như vậy, file nhị phân thông tin cần giấu sẽ là 100\*8 = 800 bít 0/1, vì mỗi kí tự mã ASCII biểu diễn bởi 1 byte. Ta có thể thấy rằng : để giấu được hết thông tin thì cần ít nhất 800 khối, vậy thì ta nên chia khối như thế nào để đủ khối giấu và dàn trải rộng trên ảnh. Lấy (512\*512)/800 =327 dư 544. Với kết quả này, kích thước khối tối đa là 327 vậy thì ta có thể chọn các kích thước phù hợp với con số này (phù hợp theo nghĩa đủ lớn và không vượt quá 327), chẳng hạn như 20\*15, 16\*16.
* Sở dĩ ta nên chọn khối có kích thước lớn vì như vậy nếu như trong trường hợp các khối bị thay đổi thì khoảng cách bít bị biến đổi sẽ xa nhau (thưa) lám cho anh sau khi giấu khó bị nhận biết hơn.
* Độ an toàn của thuật toán này là không cao, vì ta chỉ cần biết được kích thước của các khối giấu tin là ta có thể giải mã được nhanh chóng.
* Thuật toán ở trên hoàn toàn có thể áp dụng được đối với ảnh màu hoặc ảnh đa mức xám. Các loại ảnh này có giá trị của mỗi điểm ảnh được biểu diễn bởi nhiều bít. Vậy làm thế nào để có được một ma trận điểm ảnh 0,1 để thực hiện giấu tin như thuật toán trên. Rất đơn giản , ta chỉ việc chọn từ một điểm ảnh đúng một bít và lưu vào trong ma trận hai chiều các bít 0,1 . Việc chọn này được thực hiện theo quy tắc chọn bít quan trọng nhất **LSB – Least Significiant Bit**
* Đối với ảnh màu và ảnh đa mức xám ta không cần quan tâm nhiều đến việc chọn điểm cần giấu vì ta đã dùng những bít ít quan trọng nhất để giấu rồi. Do vậy, tại mỗi bước giấu ta có thể chọn một bít bất kỳ để thay đổi.

*Cải tiến thuật toán :*

Với thuật toán này việc chọn khối khá đơn giản, ta bắt đầu từ khối đầu tiên và những khối liên tiếp phía sau một cách tuần tự. Tuy nhiên, ta có thể cải tiến thuật toán bằng cách chọn ngẫu nhiên một khối chưa giấu ở mỗi lần giấu. Khi đó, ta đã làm tăng được độ an toàn của thuật toán vì khoá bậy giờ còn có thêm cả chỉ số khối đã giấu tin cho từng bit. Hoặc ta có thể thay đổi kích thước khối ở mỗi lần giấu, chẳng hạn như lần 1 có kích thước khối là 8\*8, lần 2 là 8\*12, trong trường hợp này thì khoá sẽ là kích thước khối ở mỗi lần giấu.

Một cách cải tiến thuật toán khác nữa là ta sẽ tính hệ số phân bố bit của một ma trận điểm ảnh. Hệ số phân bố bit là đại lượng đặc trưng cho mức độ rời rạc của bit 0 và 1 của ma trận đó.Việc chọn bít nào để đảo giá trị sẽ tuỳ thuộc vào hệ số bít của ma trận đó lớn hay nhỏ.Với cách này, việc giấu bít vào trong ảnh đen trắng là rất hiệu quả.

**Kỹ thuật 2 - Kỹ thuật giấu tin của WU\_LEE :**

**1. Một số khái niệm cơ bản :**

\* Phép nhân bit (AND)

Gọi a và b là bít tuỳ ý, phép tính toán nhân bít AND, ký hiệu là ^ trên hai bít a và b cho ta giá trị 1 khi và chỉ khi a=b=1, trong các trường hợp còn lại a^b =0

\* Phép cộng loại trừ (XOR)

Phép toán cộng trừ (còn gọi là phép toán so khác) XOR, ký hiệu là ⊕ trên hai bít a và b cho ta giá trị 1 nếu a ≠ b và giá trị 0 nếu a=b.

\* Bảng giá trị chân lý của hai phép toán trên:

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | a^b | a⊕b |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 |

\* Phát triển 2 phép toán trên đối với 2 ma trận

Cho A và B là hai ma trận bít cùng cấp. Ta thực hiện các phép toán như sau:

- Nếu A= (aij), B = (bij), C= (cij), D = (dij)

thì A ^ B = C với cij= aij ^ bij

và A ⊕ B = D với dij = aij ⊕ bij

\* Tổng giá trị các phân tử trong ma trận

Ta định nghĩa SUM(X) là tổng ác giá trị trên ma trận X. Chú ý rằng nếu X là một ma trận bít thì SUM(X) chính là tổng số bít 1 trong X.

2. Ý tưởng của thuật toán Wu\_Lee

- Sử dụng ma trận khoá bí mật K là một ma trận nhị phân có kích thước m x n (bằng kích thước của khối ảnh giấu tin) nhằm làm tăng độ antoàn của thuật toán. Nếu trước đây chỉ biết kích thước khối là mxn thì đói phương rất dễ khai thác được bản tin mật, nay ngoài kích thước này còn phải biết giá trị cụ thể của khoá K.

- Sử dụng phép toán AND giữa ma trận điểm ảnh và ma trận khoá (Fi^K) nhằm quy định thuật toán chỉ được phép sửa các bít trong khối Fi ứng với bít 1 trong khoá K. Như vậy khoá K được xem nhu một mặt nạ, tạo ra khung hình cho thuật toán, tăng độ an toàn.

- Sử dụng phép SUM (Tính giá trị các bít 1 trong các ma trận nhị phân) để kiểm tra điều kiện anh toàn thì thông tin được đấu. Điều kiện an toàn là 0<SUM (Fi^K)<SUM(K) có nghĩa là quy định nếu khối Fi^K toàn 0 hoặc giống nhu khoá K thì không được giấu tin để tránh bị lộ.

- Thông tin được giấu vào mỗi khối Fi các bít điểm ảnh chỉ là 1 bít. Khi thông tin được giấu, khối bìt F’i sau khi được giấu luôn đảm bảo tính bất biến: SUM(F’i) sau khi được giấu luôn đảm bảo tính bất biến: SUM (F’i^K) mod2=b (b chính là bít được giấu).

3. Thuật toán

**Input:**

- Một ảnh gốc nhị phân F.

- Một khoá bí mật K: là một ma trận nhị phân có kích thước m\*n.

- Một file thông tin cần giấu P

**Output:**

- Một ảnh đã được giấu thông tin

**Giấu tin:**

Để cho đơn giản chúng ta coi kích cỡ của ảnh F là bội của m\*n. Việc nhúng thông tin giấu vào trong ảnh sẽ được thực hiện bằng cách thay đổi một số bít của ảnh F theo quy tắc.

*Bước 1*: Chia ảnh F thành các khối nhỏ, mỗi khối có kích thước là m\*n.

*Bước 2*: Với mỗi khối ảnh nhỏ Fi thu được từ bước S1, ta kiểm tra điều kiện an toàn khi giấu tin:

0 < SUM (Fi^K) < SUM(K)

Nếu đúng thì chuyển tới bước 3 để giấu thông tin vào trong khối Fi, còn nếu không thì không giấu dữ liệu vào trong khối Fi, khối Fi sẽ được giữ nguyên.

*Bước 3*: Gọi bít cần giấu vào trong khối Fi là b, thực hiện các bước sau để thay đổi Fi:

If(SUM(Fi^K)mod2=b) then

Giữ nguyên Fi

Else if (SUM(Fi^K)=1) then

Chọn ngẫu nhiên một bít (j,k) thoả mãn đồng thời [Fi]ik = 0và [K]ik = 1 sau đó, chuyển giá trị của bít [Fi]ik thành 0. else

Chọn ngẫu nhiên một bit mà [K]jk = 1 chuyển giá trị của bít [Fi]jk từ 0 thành 1 hoặc từ 1 thành 0.

End if;

Việc chọn bít nào trong F để đảo cần tuân thủ theo nguyên tắc: Nếu Fi^K có nhiều bít 1 (SUM(Fi^K)= SUM(K)-1) thì chọn bít 1, ngược lại nếu FiAK có quá ít bít 1 (SUM (Fi^K)= 1) thì chọn 0 bít. Nguyên tắc này làm giảm khả năng bít đảo bị phát hiện.

**Giải mã:**

Nhờ bất biến có được khi giấu tin, ta dễ dàng giải mã để lấy lại thông tin đã giấu như sau. Duyệt lần lượt các khối Fi của ảnh đích F. Nếu Fi thoả điều kiện 0<SUM(Fi^K) <SUM(K) thì tính bít b đã được giấu vào trong khối bằng công thức b = SUM (F’i^K) mod 2.

**4. Minh hoạ thuật toán.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F1 | | |  | | F2 | | |  |  | Thông tin giấu B=011 | | |  | F’1 | | |  | F’2 | |
| 1 | 1 | 0 | | 1 | | 1 | 1 | | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | | 1 | | 1 | 0 | | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | | 0 | | 0 | 0 | | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | | 0 | | 0 | 0 | | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | | 0 | | 1 | 0 | | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

F3 F4 K F’3 F’4

**Hình : Mô tả quá trình đảo bít để giấu tin của thuật toán trên 4 khối.**

Giả sử một ảnh F có kích thước 6x6 và một ma trận khoá K có kích thước 3x3 như trong hình vẽ. Ta chia ảnh F thành 4 khối nhỏ mỗi khối sẽ có kích thước là 3x3 ta thu được F1, F2, F3, F4.

- Vì SUM(F1^K) =SUM(K) nên không giấu dữ liệu vào trong F1.

- Vì SUM(F2^K) =3 nên một bít có thể được giấy vào khối 2. Theo ví dụ trên bít đầu tiên được giấy là bít 0. Nên theo S3 ta sẽ chọn một bít có [F2]ij = 0 và [K]ij = 1 và đổi giá trị [F2]ij thành 1, F2 chuyển thành F2 như trên hình vẽ (bít đổi được đánh giấu xám).

- Với F3 SUM(F3^K) = 3 nhưng bít cần giấu là bít 1 nên theo Sl ta giữ nguyên F3 nhưng thực tế F3 vẫn được dấu một bít 1.

- Tương tự đối với F4, SUM (F1^K) = 4, và bít cần giấu là bít1 nên theo S3 ta chọn một bít ở [F4]ij = 1 và [K]ij= 1 rồi chuyển [F4]ij = 0. Bít thay đổi được đánh giấu xám.

**Ví dụ minh hoạ:**



**Hình : Ảnh trước và sau khi giấu các bít thông tin**

5. Phân tích đánh giá thuật toán

- Vì khoá K bí mật nên thông tin đã được nhúng là bí mật. Thuật toán này làm thay đổi nhiều nhất một bít của khối Fi khi giấu một bít thông tin vào trong khối nên với một khối có kích thức m\*n đủ lớn thì sự thay đổi của Fi là nhỏ để đảm bảo được tính an toàn của thuật toán.

- Vì phép toán AND được sử dụng để tính Fi^K, nên giá trị lớn nhất của SUM(FI^K) không thể vượt quá SUM(K) và do tính chất của phép toán AND, nếu có một khối nào thay đổi thì vị trí thay đổi chỉ xảy ra ở phần tử có giá trị l trong khoá K. Vì thế, nếu một ảnh F hoàn toàn trắng nào đó được truyền đi thi kẻ thù khi bắt được thông tin sẽ dễ dàng tìm ra được vị trí l của khoá K, đó là lí do mà ta không dùng trường hợp SUM (F1^K) - 0. Đây là một kẽ hở của thuật toán đối với khoá.

- Với trường hợp SUM (F1^K) = SUM (K) cũng tương tự nếu F hoàn toàn đen thì vị trí của bít thay đổi thì cũng là vị trí mà bít tương ứng ở khoá là 1.

Để tránh những trường hợp trên thuật toán đã phải đưa ra phụ thuộc 0 < SUM(F1^K) < SUM (K). Nhưng cho dù như thế đi chăng nữa thì vị trí tương ứng với bít bị thay đổi vũng tương ứng với bít ở vị trí đó trong khoá K có giá trị l, và bít không bao giờ bị thay đổi tương ứng sẽ là bít 0 ở vị trí đó trong khoá K. Và như thế việc chọn khoá K như thế nào là một công việc hết sức quan trọng.

- Nếu ảnh F được lựa chọn để giấu thông tin có quá nhiều điểm trắng hoặc quá nhiều điểm đen thì tỉ lệ bít giấu được sẽ thấp.

-Nói chung đối với ảnh đen trắng thuật toán này vẫn chưa đạt được những yêu cầu cần thiết về khả năng giấu. Độ an toàn thông tin đã phần nào được cải thiện hơn so với kỹ thuật thứ nhất. Vì vậy, chất lượng ảnh hưởng cũng tốt hơn. Tuy nhiên về vấn đề số lượng thông tin giấu lại giảm đi vì có những khối sẽ không được giấu tin .

6. Cải tiến thuật toán

Ta có thể cải tiến thuật toán này khi áp dụng đối với ảnh màu bằng cách khi ma trận bít không thoả mã điều kiện an toàn 0 < sum (Fi^K) < sum(K) ta vẫn thực hiện giấu tin theo ý tưởng của thuật toán, vì đối với ảnh màu ta đã áp dụng kỹ thuật tách các bit LSB, nên không ảnh hưởng đến chất lượng ảnh mà số lượng tin giấu sẽ nhiều hơn.

**Kỹ thuật 3 - Kỹ thuật giấu tin CHAN\_PAN\_TSENG**

Trong mục này đề cập đến một kỹ thuật đơn giản và đáng tin cậy để giấu những thông tin quan trọng vào trong một ảnh đen trắng (ảnh nhị phân) bằng cách sử dụng một khoá bí mật K (Private Key) và một ma trận trọng số do Yu-Yuan Chen, Hsiang – Kuang Pan và Yu – Chee Lseng thuộc khoa Công nghệ thông tin và Khoa học máy tính, Đại học Quốc gia Đài Loan nghiên cứu. Phương pháp này được chứng minh là có độ an toàn dữ liệu cao, bảo đảm chất lượng ảnh gốc và có tỷ lệ giữa kích thước thông tin giấu được với kích thước ảnh môi trường tương đối lớn so với các phương pháp khác và cho phép giấu được tới [log2 (m\*n +1)] bít dữ liệu vào trong mỗi khối ảnh có kích thước m\*n mà chỉ cấn thay đổi nhiều nhất 2 bít trong khối ảnh đó.

***1.Một số khái niệm dùng trong thuật toán***

* ***Khoá bí mật*** : Khoá là một ma trận nhị phân có cùng kích thước m\*n với kích thước của khối ảnh. Khoá được dùng một cách bí mật giữa người gửi và người nhận.
* ***Ma trận trọng số cấp r :*** Ma trận trọng số W cấp r là một ma trận số nguyên có kích thước bằng kích thước của khối ảnh m\*n và thoả mãn các điều kiện sau :
  + Wcó các phần tử nằm trong khoảng giá trị (0..2r-1) với r cho trước thoả mãn điều kiện sau : 2r  < m\*n
  + Mỗi một phần tử có giá trị từ 1..2r-1 xuất hiện ít nhất một lần.

Ví dụ : Xây dựng một ma trận trọng số kích thước 4\*4, ma trận này chứa các giá trị nằm trong khoảng 0..15 và mỗi giá trị từ 1,2,3..7 xuất hiện ít nhất một lần.

Một ví dụ ma trận trọng số W :

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 1 |
| 2 | 3 | 4 | 5 |
| 4 | 5 | 7 | 1 |

Các phép toán trên ma trận dùng trong thuật toán :

Phép toán XOR hai ma trận : A ⊕ B.

Phép toán nhân hai ma trận A ⊗ B.

Phép toán tính tổng giá trị trong ma trận SUM(X).

***2. Ý tưởng của thuật toán CHEN\_PANG\_TSENG***

Thuật toán sử dụng một ma trận khoá và một ma trận trọng sô để giấu thông tin. Việc sử dụng thêm một ma trận trọng sô W và phép toán XOR giữa ma trận điểm ảnh và ma trận khoá sẽ làm cho thuật toán đảm bảo được tốt an toàn thông tin và cũng giấu được nhiều thông tin hơn trong mỗi khối ảnh bằng cách thay đổi nhiều nhất 2 bít mỗi khối ảnh .

***3. Thuật toán giấu tin trong ảnh CHEN\_PANG\_TSENG***

**Input:**

**- F :** Một ma trận ảnh gốc dùng để giấu thông tin . F được chia thành các khối Fi , mỗi ma trận điểm ảnh Fi  có kích thước là m\*n, để cho đơn giản ta giả sử rằng F là bội của các Fi.

- **K** : Một ma trận khoá cấp m\*n

- **W** : Ma trận trọng số cấp m\*n

- **r** : Số lượng bít sẽ nhúng trong mỗi khối ảnh m\*n

- **B** : Là lượng thông tin cần giấu gồm k\*r bít, k sẽ là số khối ảnh được giấu.

- **d** : Độ chênh lệch trọng số.

**Output :**

Ảnh đã nhúng tin F’ chứa B, F’ được tạo từ các khối đã nhúng tin F’i . Mỗi khối F’i thu được từ khối Fi tương ứng sau khi đã giấu r bít thông tin từ B.

**Giấu tin :**

Thuật toán sẽ thực hiện việc giấu tin bằng cách biến đổi mỗi khối bít Fi thành F’i sao cho luôn thoả mãn điều kiện sau :

SUM(Fi ⊕ K) ⊗ W) ≡ b1b2…br (mod 2r) (\*)

Trong đó, b1b2…br là dạng biểu diễn nhị phân tạo từ dãy r bít liên tiếp trong B.

Mỗi khối bít Fi bị biến đổi nhiều nhất là 2 bít thông tin (Vì cần tạo ra điều kiện có trường hợp sẽ phải thay đổi trong khối ảnh Fi 2 bit thông tin).

Quá trình biến đổi được thực hiện gồm 4 bước sau đây :

*Bước 1:*

Tính ma trận T = Fi ⊕ K

Tính ma trận P = T ⊗ W

*Bước 2:*

Tính tổng giá trị trong ma trận P, Sum = SUM(P).

*Bước 3:*

Với ma trận T và với mọi w = 1,2,…,2r -1, ta xác định tập hợp Sw như sau :

Sw = {(j,k) | (W[j,k] = w ∧ T[j,k] = 0) ∨ (W[j,k] = 2r – w ∧ T[j,k] = 1)}

Dễ nhận thấy rằng : Sw là tập hợp các toạ độ (j,k) của ma trận Fi sao cho khi đảo bít Fi [j,k] thì Sum ở bước 2 tăng thêm w đơn vị. Thực vậy, ta có :

* Trường hợp 1 : Nếu W[j,k] = w và T[j,k] = 0 , khi đó đảo bít Fi [j,k] sẽ làm cho bít T[j,k] = 1, do đó Sum tăng lên w.
* Trường hợp 2 : Nếu W[j,k] = 2r – w và T[j,k] = 1, khi đó đảo bít Fi [j,k] sẽ làm cho T[j,k] = 0, do đó Sum sẽ giảm đi 2r – w, tức là tăng lên w theo mod 2r . Từ định nghĩa của tập Sw  , ta có :  Sw’ = Sw

*Bước 4 :*

Ký hiệu d = (b1b2…br) – SUM(P) (mod 2r).

Ta cần thực hiện việc đảo bít Fi để được F’i  , sao cho tổng Sum tính được ở bước 2 khi thay Fi bởi F’i sẽ tăng lên d.

* Nếu d = 0, không thay đổi Fi
* Nếu d ≠ 0, ta thực hiện các công việc sau :

1 .Chọn h bất kỳ thuộc tập {1,2,….2r - 1} sao cho Shd ≠ ∅ và S-(h-1)d ≠ ∅

2. Chọn phần tử (j,k) bất kỳ thuộc Shd và đảo bít Fi [j,k] (nếu là bit 0 thì đổi thành 1 và ngược lại)

3. Chọn phần tử (j,k) bất kỳ thuộc S-(h-1)d và đảo bít Fi [j,k]

Rõ ràng, để tăng Sum lên d, ta có thể chon hai tập khác rỗng là Shd và S-(h-1)d .Thật vậy, hai tập này chứa các vị trí bít trong khối Fi mà ta có thể đảo để tăng Sum lên hd và –(h-1)d một cách tương ứng , kết quả cuối cùng là Sum sẽ tăng lên hd + (-(h-1)d) = d.

Tương tự như các tập Sw khác, ta có thể coi tập S0 là chứa các vị trí mà khi đảo những bít có vị trí này trên Fi thì sẽ tăng Sum lên 0. Kết quả này cũng đạt được nếu ta không đảo bất kỳ một bít nào trên Fi . Vì vậy, ta có thể coi S0 là tập trống và khi nói đảo 1 bít có vị trí thuộc tập S0 có nghĩa là không cần làm gì cả.

Ví dụ minh hoạ :

Giả sử ta có một ma trận ảnh F 8\*8 được chia thành 4 ma trận khối ảnh F1, F2, F3, F4 có cùng cỡ 4\*4, một ma trận khoá K 4\*4 và một ma trận trọng số có cùng cỡ như sau:

Ma trận F 8\*8 :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

F1

F2

F3

F4

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 1 |
| 2 | 3 | 4 | 5 |
| 6 | 7 | 1 | 2 |

K =

W =

Trong ví dụ này, ta chọn m = n = 4, chọn r = 3, ta giấu bít sau : B = 001010000001 vào trong ảnh F. Như vậy, đoạn bít 001 sẽ được giấu vào khối F1, 010 vào trong F2, 000 vào trong F3, 001 vào trong F4.

***Bước 1:***

Ta thực hiện phép toán XOR của Fi và K. Kết quả như sau :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

F1 ⊗ K

F2 ⊗ K

F3 ⊗ K

F4 ⊗ K

**Bước 2:**

- Ta thực hiện phépnhân từng khối bít ma trận kết quả ở trên với ma trận trọng số :

(F2 ⊕ K) ⊗ W

(F4 ⊕ K) ⊗ W

(F3 ⊕ K) ⊗ W

(F1 ⊕ K) ⊗ W

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 4 | 1 | 0 | 3 | 0 |
| 5 | 6 | 0 | 0 | 5 | 0 | 7 | 1 |
| 0 | 3 | 4 | 0 | 2 | 3 | 4 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 0 | 0 | 3 | 0 | 0 | 2 | 0 | 0 |
| 0 | 6 | 7 | 1 | 5 | 6 | 7 | 0 |
| 0 | 0 | 4 | 5 | 2 | 0 | 0 | 5 |
| 6 | 0 | 0 | 2 | 0 | 7 | 0 | 2 |

***Với F1 :***

Chú ý rằng : Ta có 23 =8

Tính SUM ((F1 ⊕ K) ⊗ W) = 0 (mod 8). Vì chuỗi 3 bít cần giấu đầu tiên là 001 nên ta phải thay đổi để tăng trọng số lên 1(Vì d =1- 0(mod 2r) = 1)

Ta xây dựng tập S1 , với h =1 :

Ta nhận thấy , tại ô (2,4) thì W[2,4] = 0 và T[2,4] = 0, thoả mãn điều kiện theo thuật toán . Vậy S1 = {(2,4)} ≠ ∅, ta chọn luôn ô này để đảo bít . Khi đó ma trận khối ảnh F1 là :

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

F’1

***Với F2:***

Tính SUM((F2 ⊕ K) ⊗ W) = 2 (mod 8)

Và vì chuỗi 3 bit tiếp theo cần giấu là 010 = 2 nên d = 0, vậy không cần thay đổi F2 nữa.

***Với F3:***

Tính SUM ((F3 ⊕ K) ⊗ W) = 2 (mod 8)

Và vì chuỗi 3 bít tiếp theo cần giấu là 000 = 0, nên ta cần thay đổi F3 để tăng trọng số lên d = (0-2) mod 8 = 6.

Ta xây dựng tập S6 : Với h = 1:

Ta nhận thấy W[4,4] = 8-6 =2 và T[4,4] =1, thoả mãn điều kiện thuật toán, nên S6 = {(4,4)} ≠ ∅, ta chọn luôn ô này để đảo bít . Khi dó, ma trận khối ảnh F3 là :

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |

F’3

***Với F4:***

Tính SUM ((F4 ⊕ K) ⊗ W) = 4 (mod 8)

Và vì chuỗi 3 bít tiếp theo cần giấu là 001 = 1, nên ta cần thay đổi F4 để tăng trọng số lên 5,d = (1-4) mod 8 = 5.

Ta xây dựng tập S5 : Với h = 1: S5 = ∅.

Với h = 2 : S10 = S2 = { (2,2)}; S(-5) = S3 = { (1,3), (2,1),(3,2), (3,4)}. Ta chọn đảo bít ở hai ô [F4]2,2  và [F4]3,2

Khi đó ma trận khối ảnh của F4 là :

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |

Ảnh tạo thành sau khi ghép 4 khối điểm ảnh F11;F22;F33:F44 như sau:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |

F’1

F’2

F’3

F’4

Như vậy, ta đã giấu xong thông tin B vào trong các khối theo thuật toán CHEN\_PAN\_TSENG.

**4. Phân tích đánh giá thuật toán :**

*Độ an toàn của thuật toán :*

Đánh giá về độ an toàn của kỹ thuật giấu tin trong ảnh như đã trình bày ở trên, giả sử thuật toán lập mã là công khai, cũng giả sử thêm rằng ảnh môi trường F, giá trị r, kích thước khối m\*n không còn là bí mật. Hơn nữa , nếu người thám tin còn có cả bản mã (ảnh kết quả )F nhưng chưa biết khoá và ma trận trọng số , thì khi đó việc tìm ra thông tin giấu trong F bằng thuật toán đã nêu ở trên với các tham số được biết vẫn gần như là không thể được.

Thật vậy, ta có gần t1­ = **2mn** khả năng lựa chọn khoá K và gần

t2 = **C\*(2r -1)!\* (2r -1)mn -()**

khả năng lựa chọn ma trận trọng số W và do đó có tới t1\*t2 cách kết hợp K với W. Khi (m\*n) đủ lớn thì số lựa chọn này là rất lớn và gần như không thể tìm ra được bản tin mật. Chẳng hạn, với m = n = 4,r = 4, ta có t1=65.536,t2 = 16\*15!\*15 = 313.841.848.320.000

Ttrong truờng hợp một phần thông tin B bị lộ và người thám tin biết được hai khối ảnh Fi, Fj và hai khối ảnh tương ứng sau khi đã lần lượt giấu Bi và Bj vào trong Fi và F­j , thì khả năng giải mã được thông tin là có thể xảy ra nếu có thêm một số điều kiện .

Nếu Fi = Fj thì sự khác nhau giữa Bi và Bj sẽ cho biết mối quan hệ của trọng số tại vị trí mà Fi khác F’j và Fj khác F’j . Hơn nữa, nếu có thêm rằng Fi = F’i = F’j và chỉ có một bít tại vị trí (a,b) trong Fj bị đảo, thì khi đó giá trị của W[a,b] =Bj - Bi (mod 2r) hoặc Bi - Bj (mod 2r). Điều này có thể dễ dàng thấy được nếu ta đặt :

di = Bi – SUM((Fi ⊕ K) ⊗ W) (mod 2r) = 0

dj = Bj – SUM((Fj ⊕ K) ⊗ W) (mod 2r)

Nếu mỗi phần tử của W đều có thể được xác định chỉ nhaqnj một trong hai giá trị trên thì số khả năng có thể cho W chỉ còn là 2mn , giảm đi đáng kể so với ban đầu. Khi ma trận trọng số W bị xác định thì việc tìm khoá trở nên dễ dàng hơn . Chẳng hạn, như với giả thiết Fi = F’i và Fj và Fj khác F’j tại một vị trí duy nhất (a,b) thì khi đó K[a,b] có thể được tính bằng cách :

* Nếu Bj - Bi = W[a,b] ≠ 2r-1 thì (Fj ⊕ K) [a,b] = 0 suy ra K[a,b] = Fj [a,b].
* Nếu không Bj - Bi = -W[a,b] ≠ 2r-1 thì (Fj ⊕ K) [a,b] = 1, suy ra K[a,b] = 1 - Fj [a,b]

Tóm lại, việc giải mã thông tin càng khó khăn hơn khi kích thước khối m\*n đủ lớn và khoá K, ma trận trọng số W được cất giữ an toàn . Nếu coi đây là một hệ mã thì hệ mã có khoá bí mật giống như các hệ mã cổ điển.

*Phân tích đánh giá thuật toán:*

- Thuật toán có thể giấu được r bít vào trong một khối m\*n với điều kiện là 2r < m\*n mà cần thay đổi nhiều nhất là 2 bít trên một khối . Như vậy, thuật toán này đã có những cải tiến rất lớn so với các thuật toán khác - chỉ giấu được một bít vào mỗi khối, số lượng thông tin giấu đã nhiều hơn.

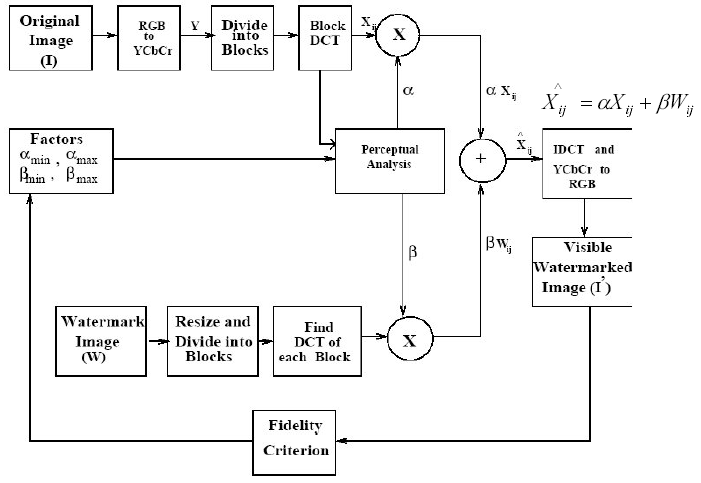
- Độ an toàn của thuật toán cũng rất cao thông qua hai ma trận dùng làm khoá để giải tin, đó là ma trận trọng số và ma trận khoá.

- Thuật toán này đương nhiên có thể áp dụng cho ảnh màu và ảnh đa mức xám. Ta cũng sẽ sử dụng kỹ thuật chon ra bít ít quan trọng nhất LSB của mỗi điểm ảnh để xây dựng ma trận hai chiều các bit 0,1 như trong thuật toán với ảnh đen trắng.

**5. Cải tiến thuật toán :**

Ta nhận thấy trong tình huống phải thay đổi 2 bít trên một khối ảnh cũng dẫn tới chất lượng ảnh sẽ bị ảnh hưởng. Vì vậy, ta có thể cải thiện thuật toán bằng cách thay đổi giá trị của ma trận trọng số W sao cho trong mọi tình huống ta chỉ cần thay đổi 1 bít trên một khối ảnh , như vậy chất lượng ảnh sẽ sau khi giấu tin sẽ tốt hơn. Mỗi lần thuật toán cần thay đổi giá trị của ma trận trọng số, ta cần lưu lại giá trị thay đổi đó và giá trị này cũng được coi là khoá của thuật toán, như vậy độ an toàn của thuật toán càng tăng thêm.

Kỹ thuật giấu tin dựa vào giải pháp giấu tin vào miền tần số :



**Kỹ thuật giấu tin J.COX**

**Thuật toán J.COX :**

Các bước của thuật toán chèn thông tin ẩn :

* Tính DCT cho toàn bộ ảnh gốc
* Tìm ra vùng có ý nghĩa nhất để chèn thông tin ẩn

*Tác giả chọn 1000 hệ số DCT lớn nhất*

* Thông tin ẩn là dãy số thực ngẫu nhiên trung bình 0 thuộc miền [-1, 1]: X=x1, x2,..., xn
* Chèn thông tin ẩn vào miền DCT theo công thức : C = C(1+α.W)

C : Hệ số của ảnh chứa thông tin ảnh .

C : Hệ số của ảnh gốc

W: Thành phần thông tin ẩn tương ứng với hệ số (u,v).

α : Hệ số cân đối giữa tính bền vững với tính ẩn, chọn 0.1

Các bước của thuật toán tách thông tin ẩn :

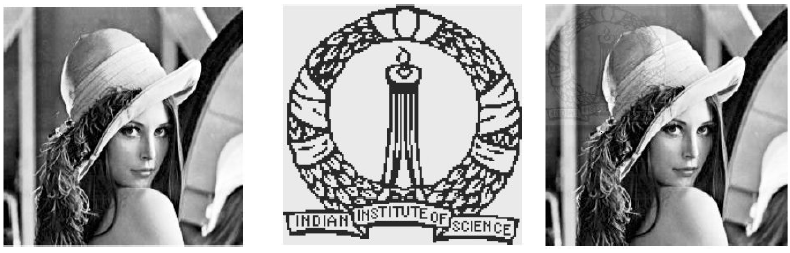
* Tính toán DCT cho toàn bộ ảnh gốc
* Tính toán DCT cho toàn bộ ảnh nhúng thông tin ẩn
* Thông tin ẩn được tách nhờ công thức biến đổi như sau:

W = 

* Sử dụng hàm “tương đương” để xác thực thông tin ẩn.

sim (W,W\*) = 

Kết quả của thuật toán J.COX :



**IV. Đánh giá về các thuật toán đã trình bày.**

Với kỹ thuật giấu tin : Giấu tin đơn giản và giấu tin bằng phương pháp WU\_LEE, thì các thuật toán này chỉ cho phép giấu một bít vào trong một khối bít. Do đó, kích thước khối bít càng lớn thì dẫn tới số bít được giấu trong ảnh là càng nhỏ.

Còn đối với thuật toán giấu tin trong ảnh CHENG\_PAN\_TSENG cho phép giấu nhiều bít hơn với [log2 (mn+1)] bit dữ liệu vào trong một khối ảnh m\*n, tức là với m\*n bít môi trưòng có thể giấu được [log2(mn+1)] bit dữ liệu. Do đó, với kích thước khối cố định thì tỷ lệ giấu bít dữ liệu lớn nhất có thể là :

f = 

Nhận thấy đây là một hàm đơn điệu giảm theo m,n, vì vậy muốn có tỷ lệ giấu tin càng lớn thì m\*n càng phải nhỏ.

Tuy nhiên, ta lưu ý rằng, độ an toàn của thông tin lại phụ thuộc tỷ lệ thuận với kích thước khối bít sử dụng để giấu bít thông tin : kích thước khối càng lớn thì độ an toàn càng cao. Do đó, tỷ lệ giấu tin sẽ phụ thuộc tỷ lệ nghịch với độ an toàn và việc chọn kích thước khối lớn sẽ làm tăng độ an toàn nhưng lại giảm tỷ lệ giấu tin và ngược lại, kích thước khối nhỏ sẽ làm tăng tỷ lệ giấu tin nhưng lại làm giảm độ an toàn.

Thông thường, ta nên chon kích thước khối bít sao cho [log2(mn+1)] = 8 hoặc bằng 4, tức là giấu được 8 bít hay 4 bít dữ liệu vào mỗi khối bít ảnh có kích thước m\*n.

Đối với từng loại ảnh thì lượng thông tin có thể giấu cũng như các yếu tố khác là khác nhau :

- Đối với ảnh đen trắng, kích thước của ảnh thường nhỏ. Vì vậy, số lượng tin giấu cũng rất ít và chất lượng ảnh sau khi giấu tin thường không tốt.

- Đối với ảnh màu, nhất là các ảnh màu có kích thước lớn thì số lượng tin giấu được nhiều hơn (kích thước file thông tin có thể tính bằng KB) và ảnh sau khi giấu tin thường không thấy được sự thay đổi bằng hệ thống mắt thường.

**V. Một số kỹ thuật giấu tin trong ảnh khác :**

**1. Kỹ thuật giấu tin dựa trên bảng màu :**

Kỹ thuật giấu tin sử dụng kỹ thuật LSB như đã nói ở những thuật toán trên thường được áp dụng cho các loại bảng màu. Vấn đề nảy sinh khi thay đổi giá trị của các bít LSB, tức là làm thay đổi con trỏ chỉ đến bảng màu.Nếu màu không được sắp xếp, sẽ xảy ra trường hợp các màu đối lập đứng kề nhau, sẽ dẫn đến tình trạng bít có tin giấu sẽ dễ bị phát hiện. Có thể giải quyết vấn đề này theo một trong hai cách sau đây:

* Sắp xếp lại bảng màu, như vậy các gam màu giống nhau sẽ nằm cạnh nhau trong bảng màu, và việc lật một bít của điểm ảnh sẽ khó bị phát hiện.
* Mở rộng bảng màu bằng cách đưa thêm vào bảng màu các màu lân cận.

Tuy nhiên, cả hai cách này đều có thể tạo ra những kẽ hở cho việc thám tin. Ở cách thứ nhất điểm dễ phát hiện là bảng mầu được sắp xếp. Còn cách thứ hai điểm yếu là xuất hiện thêm các mầu không dùng đến.

Một số phương pháp cải tiến nhắm vào việc tính các mầu kế cận sao cho ít bị phát hiện.

Trình tự kỹ thuật sắp xếp lại bảng mầu gồm các bước:

*1. Copy bảng mầu gốc*

*2. Sắp xếp bảng mầu Copy được bảng có trật tự*

*3. Ánh xạ ma trận các chỉ số của các điểm ảnh đến bảng mầu có trật tự*

*4. Nhúng thông tin mật vào các LSB bằng cách thay thể mầu*

Nếu sắp xếp bảng mầu theo độ chói (**luminance**) sẽ thành các nhóm điểm mầu không phân biệt được bằng mắt thường.

Độ chói (**L**)được tính theo công thức: **L = 0.299R + 0.587G + 0.114B**

Bản chất của của kỹ thuật này cũng là kỹ thuật giấu tin theo LSB, chỉ thêm vào các bước sắp xếp hay mở rộng bảng màu nên ngoài phần kỹ thuật thì các tính chất khác đều giống kỹ thuật giấu tin dùng bít có trọng số thấp.

**2. Kỹ thuật giấu tin dựa trên trải phổ :**

Kỹ thuật này hiện đang được ứng dụng rộng rãi trong thông tin vô tuyến của quân đội, khả năng giấu tin lớn, rất bền vững chống lại phát hiện và thám tin.Nếu coi ảnh và tin như các dải phổ thì việc giấu tin giống như trải một phổ có năng lượng thấp (tin mật) vào một dải phổ năng lượng cao (tín hiệu). Đây là một phương pháp đã được nghiên cứu kỹ lưỡng nhằm đặt mục tiêu là khó bị phát hiện, khó trích tin, dung lượng giấu cao và tính bền vững cao chống mọi phép biến đổi.

Trọng tâm của phương pháp là bộ rải tín hiệu. Thiết bị này điều chế một băng tín hiệu hẹp trên một dải mang. Tần số của sóng mang được dịch chuyển liên tục nhờ vào một bộ sinh nhiễu giả ngẫu nhiên với một khoá mật. Bằng cách này phổ năng lượng của tín hiệu được rải đều trên băng rộng với mật độ thấp, thường là dưới mức nhiễu. Để lấy lại thông tin, bên thu phải có khoá của bộ sinh giả số ngẫu nhiên để có thể chỉnh đúng tần số và giải chế tín hiệu gốc. Một người thứ ba không thể phát hiện ra tin mật vì nó ở dưới mức nhiễu.Bộ rải tín hiệu thực hiện thêm một số bước để khai thác triệt để dải phổ.

*1. Lập mã mẩu tin m với một khoá key1, thu được e (tuỳ chọn)*

*2. Chuỗi bít e sau đó được đưa qua một bộ mã hoá tự sửa lỗi cấp thấp (Low - Rate Error Correction Code) để tăng tính bền vững chống tấn công và chống nhiễu, trở thành c*

*3. Điều chế dải phổ sử dụng một bộ sinh nhiễu giả ngẫu nhiên với khoá là key2, để thu được s*

*4. Xen và trải s sử dụng khoá key3 thu được i*

*5. Cộng i vào ảnh f, thu được g 6. Một công đoạn lượng hoá được thực hiện nhằm giữ vùng phạm vi động ban đầu của ảnh phủ. Chúng ta vẫn gọi nó là g.Chúng ta giả thiết rằng ảnh mang sẽ được truyền qua một kênh truyền có nhiễu đến người nhận, và họ nhận được g'.*

Quá trình giải mã chỉ đơn thuần lặp lại các bước trên theo trật tự ngược.

*1. Dùng các kỹ thuật phục chế ảnh để thu được ảnh f' xấp xỉ ảnh gốc f*

*2. f' được trích ra từ ảnh mang g' để lộ ra dữ liệu nhúng i'*

*3. i' được đưa vào một bộ tách xen dùng khoá key3 để xây dựng lại xấp xỉ của*

*tín hiệu mật s'*

*4. s' được giải chế với key2 để được xấp xỉ tin giấu c'*

*5. c' được giải mã hoá qua bộ giải mã sửa lỗi ECC để có e'*

*6. Nếu m được mã hoá thì e' được giải mã với khoá key1 để thu lại m'.*

Đánh giá kỹ thuật giấu tin dựa vào trải phổ :

***Dung lượng giấu***

Dung lượng thông tin giấu theo phương pháp này là tương đối cao, nhưng phụ thuộc vào các tham số khác nhau dùng trong quá trình mã hoá. Nếu nén mẩu tin trước khi nhúng sẽ tăng thêm dung lượng giấu. Bộ mã hoá sửa lỗi ECC sẽ chèn thêm các dữ liệu dư thừa vào chuỗi dữ liệu mật để nhằm mục đích tự sửa lỗi. Khả năng tự sửa lỗi càng cao thì càng thêm nhiều bit kiểm tra. Chúng ta phải cân nhắc giữa khả năng sửa lỗi và khả năng giấu. Khi chèn nhiều dữ liệu vào ảnh thì lượng nhiễu càng tăng. Các bức ảnh sặc sỡ cho phép giấu nhiều dữ liệu hơn. Mật độ giấu tin cho phép của phương pháp này rất khác biệt, từ 1 byte ẩn/ 50 byte dữ liệu cho đến 10 byte ẩn/ 50 byte dữ liệu.

***Tính bền vững***

Kỹ thuật trải phổ tương đối bền vững. Mọi phép biến đổi ảnh, cộng nhiễu vào ảnh sẽ không thể phá hỏng tin giấu. Tuy vậy một kẻ tấn công ngoan cố vẫn có thể phá tin nhúng bằng cách thực hiện một số kỹ thuật xử lý ảnh số, ví dụ dùng lọc nhiễu, là cách mà chúng ta dùng để tách ra ảnh gốc.

***Khả năng giấu tin***

Kỹ thuật trải phổ được dùng nhiều trong liên lạc quân sự vì có tính bảo mật cao. Thông thường một kẻ tấn công không thể biết là có tin giấu trong đó, và ngay cả nếu phát hiện ra thì họ cũng khó có thể thám tin nếu không biết key2 và key3.

***Sự phù hợp cho giấu tin và thuỷ ấn***

Do có dung lượng giấu cao và khó bị phát hiện và thám tin nên phương pháp này rất phù hợp cho giấu tin.

***Vấn đề và giải pháp***

Chấp nhận mức độ sửa lỗi vừa phải chúng ta có thể tăng lượng tin giấu, và nhờ vậy tác động lên khả năng của bộ cộng nhiễu. Nếu dùng bộ mã hoá sửa lỗi ECC thì sẽ tăng khả năng sửa lỗi nhưng lại làm giảm dung lượng tin giấu. Một hướng cải tiến là nâng cao chất lượng ảnh gốc sao cho giảm tỷ lệ bít lỗi của tín hiệu phục hồi, và như vậy có thể giảm bớt các bít dư thừa dùng tự sửa lỗi. Alexander Herrigel đã phát triển các kỹ thuật tăng tính bền vững chống các phép biến đổi như cắt xén ảnh, hay biến đổi hình học khác.

* Thứ nhất là ảnh phủ được chia làm các khối và cùng một thông tin được nhúng nhiều lần vào các khối đó, nhờ vậy thông tin mật có thể được trích từ một phần bất kỳ của ảnh.
* Thứ hai là một mẫu dược cộng thêm vào dải phổ qua một phép biến đổi lôga-cực áp dụng trên ảnh, xác định hệ số tỷ lệ và định hướng ảnh, qua đó tin mật trở nên bền vững đối với các phép co giãn và xoay.

Cuối cùng có thể cộng các bộ lọc giác quan thích nghi trước khi chèn tín hiệu của mẩu tin mật, nhờ đó các nhiễu cộng thêm vào sẽ chắc chắn nằm dưới ngưỡng giác quan. Tuy nhiên chính điều này lại làm tăng tỷ lệ lỗi trong khi phục hồi vì nó làm giảm năng lượng của tín hiệu nhúng.

**3. Kỹ thuật giấu tin dựa vào mã khối bề mặt**

Kỹ thuật mã khối bề mặt do Bender phát triển và công bố trên tạp chí IBM Systems Journal vào năm 1996 [17]. Đây là một kỹ thuật chỉ chèn được rất ít dữ liệu, tương đối bền vững chống lại các phép biến đổi hình học. Nhược điểm lớn nhất của phương pháp này là cần đến sự can thiệp trực tiếp của con người trong quá trình mã hoá.

Mã hoá khối bề mặt hoạt động theo cách sao chép một vùng có mẫu hoa văn bề mặt ngẫu nhiên tìm thấy trong ảnh đến một vùng có hoa văn tương tự. Quá trình mã hoá được thực hiện thủ công bằng cách chọn vùng làm việc, sau đó sử dụng một số mặt nạ để chọn vùng sẽ sao chép, ví dụ một khối văn bản đồ hoạ, để sau khi giải mã mặt nạ trở nên hiện diện.

Quá trình giải mã hoá được thực hiện các bước sau:

1.Tính tương quan (autocorrelate) của ảnh với chính nó để làm nổi lên các vùng mà các khối giống nhau nằm chồng lên nhau. Càng nhiều vùng được sao chép thì càng nổi rõ sau phép tương quan ảnh với chính nó

2. Dịch chuyển các vùng ảnh được chỉ ra trong bước 1. Trích ảnh ra khỏi phần dịch chuyển

3. Bình phương kết quả và dùng ngưỡng để phân lại những giá trị sát 0. Các vùng sao chép sẽ để lộ ra các tin giấu.

Đánh giá về kỹ thuật giấu tin dựa trên mã khối bề mặt:

***Dung lượng giấu***

Có thể thấy ngay rằng khối sao chép không được quá lớn (16x16 điểm ảnh), hơn nữa trên đó có một số mặt nạ nên lượng tin giấu được rất ít. Nếu chúng ta sửa quá nhiều vùng ảnh theo cách này thì dễ bị lộ. Hơn nữa không phải tất cả các vùng đều phù hợp với kỹ thuật này, ví dụ như các vùng mầu đồng nhất, vì chúng sẽ làm loang to các thay đổi.

***Tính bền vững***

Nếu khối bề mặt sao chép đủ lớn (trên 16 x 16 điểm ảnh), thì bản thân khối cũng sẽ thay đổi theo ảnh trong các phép biến đổi phi hình học như lọc, nénn và quay. Mặt nạ nhúng vẫn hiện diện, chỉ có sẽ bị xoay hay lọc so với ban đầu. Cắt xén ảnh sẽ phá huỷ khối bề mặt giấu tin nếu một trong hai bản sao nằm ngoài vùng cắp. Các biến đổi hình học cũng sẽ biến đổi mặt nạ làm cho nó bị mất.Nếu chỉ một trong hai vùng giống nhau bị kẻ tấn công phá huỷ thì thông tin nhúng sẽ bị mất.

***Khả năng giấu tin***

Nếu chỉ một trong hai vùng giống nhau bị kẻ tấn công phá huỷ thì thông tin nhúng sẽ bị mất.

***Sự phù hợp cho giấu tin và thuỷ ấn***

Do dung lượng tin giấu là rất ít, có tính bền vững đối với các biến đổi vô tình và dễ trích suất nên kỹ thuật này phù hợp cho thủy ấn.

***Vấn đề và giải pháp***

Nhược điểm lớn nhất của phương pháp này là cần có sự hỗ trợ thủ công của con người để chọn vùng sao chép. Một hướng giải quyết là để máy tính tự làm việc này, có thể có sự giám sát của con người.Các hướng nghiên cứu khác tìm các khối trong vùng tần số ít bị để ý và bền vững đối với các thuật toán nén.

**4. Kỹ thuật giấu tin dựa vào mảnh chắp**

Đây một kỹ thuật giấu tin ra đời sớm, làm việc theo nguyên lý giấu một bít đơn vào một cặp ảnh chắp (tập điểm) lấy từ hai phần khác nhau của ảnh Kỹ thuật Patchwork sử dụng trong dựa vào một quá trình ngẫu nhiên mà quá trình này lại dựa trên yếu điểm của mắt người là phân biệt độ chói. Sử dụng phương pháp mã hoá mẫu dư thừa (redundant pattern encoding) để phân bổ các thông tin trên ảnh phủ giống như công việc chắp vá. Patchwork có thể giấu một mẫu tin nhỏ nhiều lần trong một ảnh. Tư tưởng chủ đạo là chọn ngẫu nhiên hai mảnh trên tệp ảnh mang, thay đổi độ sáng, một theo chiều tăng, một theo chiều giảm. Patchwork (chắp mảnh) cũng do Bender phát triển và công bố trên tạp chí IBM Systems Journal vào năm 1996 [17]. Trong kỹ thuật này, n cặp điểm ảnh (a,b) được chọn ngẫu nhiên. Độ sáng của a được tăng một đơn vị và độ sáng của b được giảm một đơn vị. Như vậy tổng sự chênh lệch của n cặp điểm là 2n.

*Thuật toán làm việc như sau:*

* *Tạo một chuỗi bít giả ngẫu nhiên để chọn các cặp điểm từ ảnh phủ*
* *Với mỗi cặp, gọi d là sự khác biệt giữa hai điểm (ví dụ độ sáng)*
* *Mã hoá một bít thông tin vào trong cặp. Ví dụ nếu d<0 thể hiện bit 0, d>0 thể hiện bit 1. Nếu các điểm không đúng trật tự đó, đổi chúng cho nhau*
* *Nếu d lớn hơn một ngưỡng nào đó hoặc bằng 0 thì bỏ qua cặp điểm đó và tìm cặp mới.*

Vì trong thuật toán có sự hoán đổi vị trí điểm ảnh nên có thể dẫn đến làm biến đổi ảnh. Vì vậy, thuật toán này chỉ phù hợp cho nhúng một lượng tin nhỏ. Một lợi điểm của thuật toán là sự bền vững với các thuật toán nén mất dữ liệu. Để xác định các điểm sẽ dùng trong ảnh, một bộ sinh số giả ngẫu nhiên được sử dụng, với một khoá bí mật được chia xẻ giữa người gửi và người nhận, vì thuật toán giải mã cũng sẽ phải tới thăm cùng các điểm ảnh đó để trích suất thông tin.

Nếu chúng ta tính tổng: ***S(n)=sum(a[i]-b[i], i=1..n)***

trong đó ***a[i]*** và ***b[i]*** là điểm thứ i tương ứng trong mảnh A và B, nó bằng 0 trên một ảnh bình thường. Sau khi thực hiện n lần ta được:

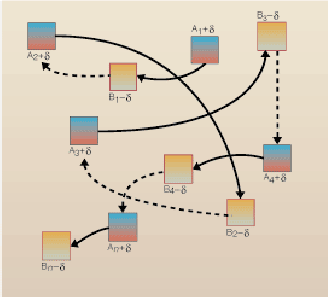
***a[i]=a[i]+delta***

***b[i]=b[i]+delta***

Giá trị kỳ vọng của S(n) là: ***2\*delta\*n***

Với ***n*** đủ lớn thì sau khi tính ***S(n)*** cho một ảnh chúng ta có thể chắc chắn rằng ảnh có thuỷ ấn hay không trong mảnh chắp.

Sau nhiều lần lặp trong phương pháp Patchwork, tại bước lặp thứ i hay mảnh chắp Ai và Bi được chọn giả ngẫu nhiên trong ảnh. Ai được tăng độ sáng δ còn Bi giảm đi δ.



**Hình : Lặp trong phương pháp Patchwork**

Đánh giá kỹ thuật giấu tin dựa trên phương pháp Patchwork :

***Dung lượng giấu***

Lượng thông tin giấu theo phương pháp này là rất thấp.

***Tính bền vững***

Một trong những đặc tính quan trọng của phương pháp Patchwork là bền vững chống cắt xén và các phép hiệu chỉnh độ tương phản và tông mầu. Cắt xén ảnh dẫn đến độ chíng xác giảm theo cấp lô-ga-rít vì kích thước ảnh bị thu nhỏ. Biến đổi thứ hai thường làm thay đổi độ sáng của các điểm ảnh, không ảnh hưởng đến tin mật vì Patchwork hoạt động theo nguyên lý khác. Một số thử nghiệm đã cho thấy rằng sau khi nén ảnh theo phương pháp JPEG với hệ số chất lượng là 75%, vẫn có thể giải mã lại thông tin với hệ số 85%.

Tuy nhiên Patchwork dễ bị huỷ bởi các phép biến đổi hình học như dịch, xoay, co giãn Một sự tấn công vào hệ thống giấu tin dùng phương pháp này sẽ thành công nếu chúng có thể thay đổi vị trí dữ liệu nhằm thay đổi chúng theo cách vô hình và như vậy làm thay đổi vị trí các điểm sử dụng trong Patchwork.

***Khả năng giấu tin***

Các vùng chắp mà thông tin được giấu trong đó gần như và vô hình vì vậy không cần đến khoá và hàm sinh số giả ngẫu nhiên để tìm các điểm giấu tin. Mặt khác nếu dùng cùng một khoá để mã hoá một số lớn các ảnh cùng kích thước thì chỉ cần lấy trung bình của các ảnh là sẽ tìm được mảnh chắp.

***Vấn đề và giải pháp***

Vấn đề đầu tiên là nếu chúng ta sử dụng điểm làm đơn vị của mảnh chắp thì các nhiễu được cộng thêm vào ảnh có tần số cao. Dễ bị lọc khi qua một loạt các công đoạn xử lý như nén mất dữ liệu … Một giải pháp là dùng các mảnh chắp là một vùng nhỏ thay vì một điểm, với sự phân bổ của delta nó sẽ rải nhiễu ra dải rộng hoặc đưa xuống vùng tần số thấp và như vậy thông tin sẽ không bị lọc mất.Một vấn đề nữa đó là sự nhạy cảm đổi với các phép biến đổi hình học. Khắc phục điều nằy bằng cách thêm các đặc tính nhận dạng thống kê.

**5. Kỹ thuật giấu tin dựa vào hệ số của phép chiếu trực giao:**

Kỹ thuật này cũng thuộc nhóm phương pháp biến đổi cosin rời rạc (DCT). Nó chủ yếu được áp dụng trong thuỷ ấn. Cách tiếp cận là xử lý các hệ số của phép chiếu của ảnh trên một hệ trực giao, nhờ đó các thông tin mật sẽ được ẩn trong các phần quan trọng của dữ liệu và khó bị phá huỷ bởi các phép xử lý ảnh thông thường cũng như nén mất dữ liệu.

Kỹ thuật này trên phương diện nào đó gần giống kỹ thuật LSB ở chỗ nó thay đổi chút ít các hệ số lớn.

Đây là một sự cải tiến của kỹ thuật DCT cổ điển, với kết quả là mức độ bảo mật cao hơn. Có thể coi nó là một sự tổng quát hoá của phương pháp DCT vì cơ sở của phép chiếu trực giao không sử dụng hàm cosin nữa. Các véc tơ được tạo từ một khoá bí mật, còn lại các bước khác giống như DCT.

*Tóm tắt quá trình này qua các bước sau:*

*1. Sinh ra một tập n các véc tơ trực giao (ánh xạ hai chiều) từ một khoá key*

*2. Chiếu ảnh lên một hệ toạ độ được một tập các hệ số {c[i], i=1..n}*

*3. Sửa các hệ số lớn nhất để mã hoá các bit tin m(k) với một mật độ mong muốn*

***a: c'[j]=c[j] (1+a\*m(k))***

*4. Tái tạo lại ảnh bằng cách ngịch đảo lại phép chiếu*

Quá trình giải mã ngược lại, tính sự chênh lệch giữa các hệ số tính toán trên ảnh mang và ảnh gốc. Như vậy trong phương pháp này chúng ta cần ảnh gốc để thu tách thông tin. Phần chính của kỹ thuật này là chọn một khoá để tính véc tơ cho phép chiếu trực giao. Có phương pháp sử dụng một hàm giả ngẫu nhiên sau đó dùng thuật toán Gramm-Schmidt.

Đánh giá kỹ thuật giấu tin dựa trên hệ số của phép chiếu trực giao:

***Dung lượng giấu***

Dung lượng giấu phụ thuộc vào ảnh gốc và véc tơ cơ sở, vì chúng ta cần các hệ số lớn để giấu tin. Một ảnh có thể có giấu lượng tin khác nhau tuỳ vào khoá được chọn.

***Tính bền vững***

Phương pháp minh hoạ đã chứng minh sự bền vững với các tấn công, bao gồm cả nén mất dữ liệu, lọc bằng các chương trình xử lý ảnh. Để có thêm sự bền vững chống các biến đổi cắt xén ảnh, người ta sử dụng mã hoá dư thừa, nhúng cùng một thông tin vào nhiều khối ảnh khác nhau.

***Khả năng giấu tin***

Vì véc tơ cơ sở dùng cho kỹ thuật này luôn thay đổi nên khó có thể tách thông tin nếu không biết khoá. Tuy nhiên nếu dùng lại một khoá nhiều lần thì kẻ tấn công có thể dùng các phương pháp thống kê để dò ra khoá, vì vậy để đảm bảo tính bí mật phải thường xuyên thay đổi khoá.

***Sự phù hợp cho giấu tin và thuỷ ấn***

Do có tính bền vững cao và cần phải có ảnh gốc nếu muốn tách tin nên kỹ thuật này phù hợp cho thuỷ ấn. Tuy nhiên kỹ thuật DCT, là một biến thể của kỹ thuật này lại được dùng chủ yếu cho giấu tin vì khả năng giấu một lượng thông tin lớn.

**6. Kỹ thuật giấu tin dùng hệ số DCT**

Các ảnh JPEG có tỷ lệ nén cao, chất lượng tốt, do đó chúng được sử dụng nhiều trên mạng. Tuy nhiên các tệp ảnh JPEG không phù hợp với xử lý bít như các ảnh dựa trên bảng mầu, tuy vậy vẫn có thể dùng chúng để giấu dữ liệu. Ảnh JPEG sử dụng biến đổi Cosin rời rạc để thực hiện nén ảnh. Biến đổi cosin rời rạc là phép biến đổi mất dữ liệu vì không thể tính chính xác các giá trị cosin, cũng như có thể có các lỗi làm tròn. Độ lệch giữa dữ liệu gốc và dữ liệu phục hồi lại sau khi biến đổi phụ thuộc vào các giá trị và phương pháp sử dụng để tính các trị cosin rời rạc. Cũng có thể xử lý ảnh dùng biến đổi Fourier nhanh hoặc biến đổi con (wavelet transformation).

Thuật toán JPEG làm việc bằng cách chia ảnh ra thành các ma trận 8x8. Sau đó tính hệ số biến đổi cosin rời rạc cho từng ma trận. Bước tiếp theo các hệ số này được nhân với một ma trận lượng hoá. Kết quả thu được sẽ được làm tròn đến số nguyên gần nhất, cuối cùng các số nguyên này được nén và lưu lại. Các cấu tử DCT chính là nơi chúng ta có thể giấu dữ liệu. Cách tiếp cận phổ biến là chọn các hệ số DCT lớn và sử đi chút ít. Vì hệ số lớn tức mức "năng lượng" cao nên ít làm thay đổi ảnh nhất. Một hướng khác là chọn các hệ số DCT trong các vùng mà mắt người không nhìn thấy. Các thuật toán JPEG nổi tiếng áp dụng trong F5 và JSteg đều dùng cách sửa DCT để nhúng dữ liệu. Cả hai phương pháp này đều qua được mắt thường nhưng không qua được các phương pháp phân tích thống kê.

Đây là kỹ thuật giấu tin mà dung lượng bản tin giấu không lớn nên thường dùng nhiều hơn trong công việc thuỷ ấn.

**7. Kỹ thuật giấu tin dựa vào lượng hoá lỗi :**

Ý tưởng của kỹ thuật này là lượng hoá lỗi theo một sơ đồ cho trước. Mức chênh lệch tín hiệu Δi được tính bằng cách lượng hoá sự chênh lệch giữa hai điểm ảnh kề nhau xi và xi+1.

***Δi =Q(xi+1 - xi )***

Khi đó lỗi sẽ sinh ra khi lượng hoá. Tại đầu nhận Δi được tính và cộng vào với giá trị mẫu gần nhất để tính xấp xỉ xi.

Khoá mật cho thuật toán này là bảng các bít ứng với Δi.

Ví dụ:

Δi

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |

Để cất bit thứ i của mẩu tin vào ảnh phủ, trước hết tính Δi. Nếu bit ứng với Δi không giống với bit bí mật, thì Δi sẽ được thay với Δk gần với Δi nhất và có bít gán với nó trùng với bít bí mật. Sau đó chuỗi Δi được mã hoá entrôpi và gửi đi. Tại đầu nhận người ta tính lại Δi, sau đó dùng khoá bí mật (chính là bảng ánh xạ trên) để lấy lại chuỗi 0, 1.

Dung lượng tin giấu trong phương pháp này là rất ít, nên thích hợp hơn với việc thuỷ ấn.

**8. Kỹ thuật giấu tin dựa trên mặt nạ và lọc:**

Các kỹ thuật mặt nạ và lọc giấu tin bằng cánh đánh dấu một ảnh theo cách tương tự như kỹ thuật thuỷ ấn. Vì các kỹ thuật thuỷ ấn thiên về tích hợp thông tin vào ảnh nên chúng không bị mất khi nén mất dữ liệu. Bằng cách phủ hoặc che mặt nạ các tín hiệu yếu (nhưng vẫn nhận biết) với các tín hiệu khác để làm tín hiệu yếu trở nên vô hình, chúng ta khai thác điểm yếu của hệ thống thị giác của con người là không phát hiện ra các thay đổi nhỏ trên các vùng thời gian của ảnh. Về mặt kỹ thuật mà nói thì thuỷ ấn không phải là một dạng của giấu tin. Nếu nói một cách chặt chẽ thì giấu tin giấu dữ liệu vào ảnh trong khi thuỷ ấn mở rộng thông tin ảnh biến thông tin trở thành các thuộc tính của ảnh phủ, phục vụ mục đích xác thực và bản quyền. Các kỹ thuật mặt nạ phù hợp với sử dụng trên ác ảnh nén mất dữ liệu như JPEG hơn kỹ thuật LSB vì nó tương đối miễn dịch đối với các phép xử lý ảnh như nén, hay cắt xén.

Kỹ thuật giấu tin này gần tương tự như kỹ thuật thuỷ ấn nên dung lượng giấu tin là không lớn.

….

**9. Nhận xét đánh giá chung về các kỹ thuật giấu tin:**

Lĩnh vực giấu tin cũng giống như mật mã học, còn nhiều hướng nghiên cứu tìm ra các phương pháp mới và kỹ thuật mới. Ở đây hai lĩnh vực có sự tương đồng ở chỗ những cái mới không nằm chính trong lĩnh vực đó mà phụ thuộc vào kết quả nghiên cứu của các lĩnh vực khác. Nếu một lúc nào đó trong toán học người ta có thể nghĩ ra cách giải các bài toán khó hoặc tìm ra một hàm cửa lật chẳng hạn thì có thể đó sẽ là cơ sở cho một hệ mật mã mới. Trong khi một phương pháp nén ảnh mới, ví dụ Fractal hay biến đổi Wavelet có thể là xuất phát điểm cho nhiều hướng nghiên cứu giấu tin mới .

Các hướng nghiên cứu phát triển trong lĩnh vực giấu tin trong ảnh chịu ảnh hưởng của một số lĩnh vực khác, trong số đó có:

* Nghiên cứu hệ thống thị giác con người và sự cảm nhận mầu sắc
* Các phương pháp xử lý ảnh
* Các cách nén ảnh
* Các phép biến đổi ảnh

Tập trung nghiên cứu khai thác các "kẽ hở" của các lĩnh vực trên là định hướng chính của công tác nghiên cứu giấu tin trong ảnh.

**CHƯƠNG III : MỘT SỐ CHƯƠNG TRÌNH ỨNG DỤNG GIẤU TIN TRONG ẢNH**

**I. Hide And Seek v4.1**

Ứng dụng này gồm một số các chương trình chạy dưới DOS do Colin Maroney viết để giấu tin vào các ảnh GIF. Tư tưởng chủ đạo là giấu ngẫu nhiên, vậy nên nếu lượng thông tin cần giấu nhỏ thì nó sẽ được dải đều khắp ảnh. Tuy nhiên nếu lượng thông tin nhiều thì các vùng thay đổi dầy hơn, dễ bị phát hiện. Hạn chế của chương trình là chỉ dấu được tối đa 19,000 byte vì màn hình hiển thị chỉ sử Một số kỹ thuật giấu tin trong ảnh mầu dụng 320 x 480 điểm ảnh. Nếu ảnh sử dụng bé hơn kích thước này sẽ xuất hiện vùng đen quanh ảnh.

**II. StegoDos**

Cũng là một số các chương trình chạy dưới DOS, sử dụng ảnh 320 x 200 điểm ảnh, 256 mầu để giấu tin.

**III. White Noise Storm**

Do Ray (Arsen) Arachelian viết, dễ dùng hơn và nhúng được nhiều tin hơn các chương trình trước. Ảnh mang không cần có kích thước cố định, tính vô hình cao.

**IV.S-Tools for Windows**

Có lẽ đây là chương trình giấu ảnh tốt nhất. Có thể giấu tin trong ảnh BMP, GIF, tệp âm thanh WAV và thậm chí lên các vùng chưa dùng đến của đĩa mềm. Giao diện đồ hoạ kéo thả. Để giấu tin chỉ cần kéo biểu tượng tệp tin cần giấu và thả lên ảnh. Chương trình sử dụng một số hệ mật mã là DES, DES ba đoạn, IDEA và MDC.

Tài liệu tham khảo:

* + 1. ***Kỹ thuật đồ hoạ máy tính*** *- Đặng Văn Đức - Viện Công Nghệ Thông Tin – Trung tâm khoa học và công nghệ quốc gia.*
    2. *Luận văn cao học các khoá 8,9 – Khoa Công Nghệ - Đại học Quốc gia Hà Nội.*
    3. *Tài liệu bài giảng (file \*.pdf) :* ***Kỹ thuật đồ hoạ máy tính*** *– Giáo viên TS.Đặng Văn Đức.*
    4. *Các Website có liên quan.*