**GIẤU TIN TRONG ẢNH BITMAP 24 BIT VÀ ỨNG DỤNG**

Sinh viên thực hiện: Nguyễn Hồng Hải

hai117.c@gmail.com

Giáo Viên Hướng Dẫn: PGS.TS Nguyễn Văn Tảo

# MỤC LỤC

MỤC LỤC……………………………………………………………………………………………………………………………………… 2

MỞ ĐẦU………………………………………………………………………………………………………………………………………… 3

CHƯƠNG 1: TỔNG QUAN VỀ GIẤU TIN …………………………………………………………………… 4

1.1 Mô hình giấu tin……………………………………………………………………………… 4

1.2 Các kĩ thuật giấu tin……………………………………………………………………………. 5

1.3 Các ứng dụng chính của giấu tin……………………………………………………………… 5

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT……………………………………………………………………………………… 7

2.1 Cấu trúc ảnh bitmap…………………………………………………………………………… 7

2.2 Thuật toán LSB ( Least Significant Bit)……………………………………………………… 14

CHƯƠNG 3: THỰC HÀNH XÂY DỰNG ỨNG DỤNG……………………………………………………… 16

KẾT LUẬN…………………………………………………………………………………………………………………………………… 19

TÀI LIỆU THAM KHẢO………………………………………………………………………………………………………… 20

# MỞ ĐẦU

Công nghệ thộng tin ngày càng phát triển , điều đó đồng nghĩa với việc càng ngày thông tin được biểu diễn dưới dạng kỹ thuật số càng nhiều ,chúng được biểu diễn và truyền tải trong môi trường thế giới số do đó vấn đề bảo mật thông tin trong môi trường thế giới số ngày càng trở nên quan trọng. Đã có rất nhiểu hệ mật ra đời , nhưng sự phát triển của công nghệ thông tin cũng cho phép một số lượng lớn các phép tính được giải quyết rất nhanh điều này làm giảm tính an toàn của các hệ mật .

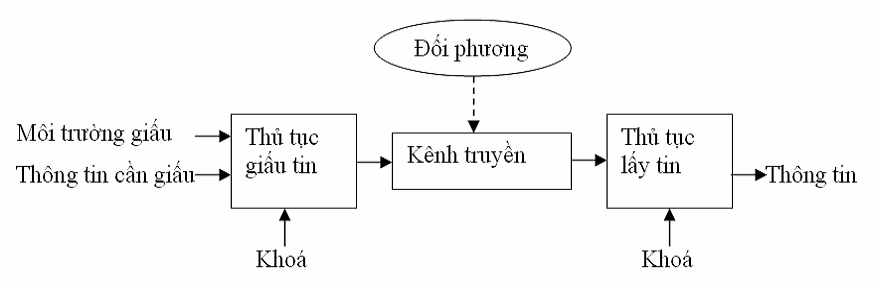
Một hướng tiếp cận mới trong bảo mật là giấu tin (information hiding),tức là những thông tin số cần bảo mật sẽ được giấu vào trong 1 đối tượng dữ liệu số khác(gọi là môi trường giấu tin) sao cho sự biến đổi của môi trường sau khi giấu tin là rất khó nhận biết,đồng thời có thể phục hồi lại được các thông tin đã giấu khi cần. Một ưu điểm của hướng tiếp cận giấu tin so với mã hóa là khi tiếp cận môi trường giấu tin,đối phương khó xác định được là có thông tin giấu trong đó hay không. Ngoài ra phương pháp giấu tin còn thể hiện được ưu thế rõ rệt trong nhiều ứng dụng như phân phối tài liệu số,bảo vệ bản quyền….

# CHƯƠNG 1:

# TỔNG QUAN VỀ GIẤU TIN

## 1.1 Mô hình giấu tin

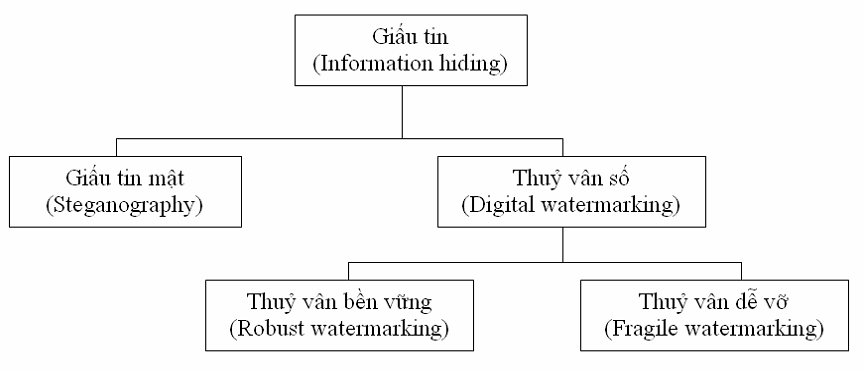
Để thực hiện giấu tin cần xây dựng được các thủ tục giấu tin. Các thủ tục này sẽ thực hiện nhúng thông tin cần giấu vào môi trường giấu tin. Các thủ tục giấu tin thường được thực hiện với một khóa giống như trong các hệ mật mã để tăng tính bảo mật. Sau khi giấu tin ta thu được chứa thông tin giấu và có thể phân phối đối tượng đó trên kênh thông tin. Để giải mã thông tin cần nhận được đối tượng có chứa thông tin đã giấu,sử dụng thủ tục giải mã cùng với khóa đã dùng trong quá trình giấu để lấy lại thông tin.



Hình 1: Mô hình giấu tin và lấy tin

## 1.2 Các kĩ thuật giấu tin

Các kĩ thuật giấu tin mới được chú ý phát triển mạnh trong khoảng 10 năm trở lại đây,nên việc phân loại còn chưa được thống nhất. Sơ đồ phân loại sau được đưa ra năm 1999 và được nhiều người chấp nhận:



Hình 2: Phân loại các mô hình giấu tin

Theo sơ đồ này, giấu tin được chia thành hai hướng chính là giấu tin mật và thủy vân số.Giấu tin mật quan tâm chủ yếu đến lượng tin có thể giấu,còn thủy vân số quan tâm đến tính bền vững của thông tin giấu. Trong từng hướng chính lại được chia ra các hướng nhỏ hơn,chẳng hạn với thủy vân số thì có thủy vân bền vững và thủy vân dễ vỡ. Thủy vân bền vững cần được bảo toàn được các thông tin thủy vân trước các tấn công như dịch chuyển ,cắt xén,xoay đối với ảnh. Ngược lại thủy vân dễ vỡ cần phải dễ bị phá hủy khi gặp các tấn công nói trên.

## 1.3 Các ứng dụng chính của giấu tin

Giấu tin mật (Steganography): các thông tin cần bảo mật được giấu trong các đối tượng vỏ và các đối tượng này có thể được truyền công khai tới người nhận mà không gây bất cứ sự chú ý nào của đối phương. Người nhận sẽ sử dụng thuật toán và khóa nào đó( đã thỏa thuận giữa 2 người) để khôi phục lại thông tin mật. Yêu cầu kĩ thuật là tỉ lệ giấu tin cần lớn nhưng bằng các giác quan không thể nhận thấy được sự khác biệt của đối tượng trước và sau khi giấu tin vào.

Bảo vệ bản quyền (Copyright Protection): đây là ứng dụng phổ biến nhất của thủy vân số. Một thông tin nào đó mang ý nghĩa quyền sở hữu tác giả(gọi là thủy vân) được nhúng vào trong các sản phẩm số. Yêu cầu kĩ thuật là việc nhúng thủy vân không ảnh hưởng đáng kể đến cảm nhận sản phẩm và phải bền vững trước các tấn công,tồn tại lâu dài cùng sản phẩm.

Xác thực thông tin (authentication): một thông tin được giấu trong đối tượng số để nhận biết xem đối tượng đó có bị thay đổi hay không. Yêu cầu kĩ thuật là việc nhúng thủy vân không ảnh hưởng đáng kể đến cảm nhận đối tượng và rất dễ bị phá hủy trước các tấn công. Đây là ứng dụng của thủy vân dễ vỡ.

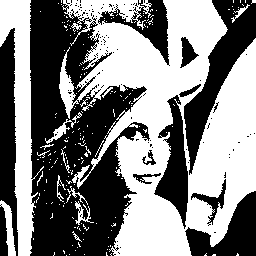
# CHƯƠNG 2:

# CƠ SỞ LÝ THUYẾT

## 2.1 Cấu trúc ảnh bitmap

Ảnh BITMAP là định dạng ảnh do microsoft đề xuất , có phần mở rộng là BMP , loại ảnh này truyền tải , sử dụng rộng rãi trên máy tính , và các thiết bị điện tử khác. Ảnh bitmap được chia thành ba dạng : ảnh nhị phân ( ảnh đen trăng ) , ảnh đa mức xám , ảnh màu.

· **Ảnh đen trắng** : là ảnh mà mỗi điểm ảnh chỉ thể hiện một trong hai trạng thái 0 và 1 để biểu diễn trạng thái điểm ảnh đen hay trắng.



Hình 3: ví dụ về ảnh đen trắng

· **Ảnh đa mức xám** : là ảnh mà mỗi điểm ảnh được biểu diễn bởi một giá trị và đó là cường độ sáng của điểm ảnh.



Hình 4: Ví dụ về ảnh đa cấp xám

· **Ảnh màu** : là ảnh mà mỗi điểm ảnh được biểu diễn bởi ba đại lượng R, G, B . Số lượng màu có thể của loại ảnh này lên tới 265^3 màu khác nhau.Nhưng số lượng màu trên thực tế của một ảnh nào đó thường khá nhỏ. Để tiết kiệm bộ nhớ với các ảnh có số lượng màu nhỏ hơn 256 thì màu các điểm ảnh được lưu trữ dưới dạng bảng màu.Với ảnh có số màu lớn thì các điểm ảnh không tổ chức dưới dạng bảng màu , khi đó giá trị của các điẻm ảnh chinh là giá trị của các thành phần màu R,G,B. Với ảnh có số lượng màu lớn ,tùy theo chất lượng ảnh mà quyết định số bit để biểu diễn cho mỗi màu thường là 24 bit, hoặc 32 bit . Với ảnh 24 bit mỗi thành phần màu được biểu diễn bởi một byte(8 bit).



Hình 5: ví dụ về ảnh màu

Ảnh bitmap đựợc lưu trữ dưới dạng nhị phân, một tệp dạng bitmap được chia thành các phần cơ bản như :

-Phần tiêu để tệp (Bitmap header)

-Thông tin về ảnh (Bitmap Infor)

-Bảng màu (Palette Table)

-Vùng dữ liệu(Data)

Thứ tự được lưu trữ trong bộ nhớ như sau: Bitmap Header->Bitmap Infor->Palette Table->Data

**Ý nghĩa của các phần trong tệp ảnh bitmap**.

-Bitmap Header: Mô tả thông tin chung về tệp định dạng bitmap, độ lớn của phần này cố định với mọi tệp bitmap.

-Bitmap Infor: Mô tả thông tin về ảnh được lưu trữ, độ lớn của phần này cố định.

-Pallete Table: Bảng màu của ảnh bitmap, độ lớn của phần này có thể bằng không ( không có bảng màu) đối với ảnh đen trắng và ảnh màu có số lượng màu lớn hơn 256 màu.

-Data: Thông tin về từng điểm ảnh , độ lớn của phần này phụ thuộc vào kich thước ảnh. Phần Data lưu trữ ảnh theo hướng từ dưới lên trên và từ trái qua phải.

Kích thước và giá trị các trường trong tệp ảnh

**Bitmap Header**

Phần này có độ lớn cố định là 14 bytes , phần này dùng để mô tả thông tin chung về tệp như :

Kiểu tệp, độ lớn và một số thông tin liên qua đến tệp.

|  |  |  |
| --- | --- | --- |
| **Offset(byte)** | **Giá Trị** | **Ý nghĩa** |
| 1 | ‘B’ | Định dạng kiểu tệp |
| 2 | ‘M’ | Định dạng kiểu tệp |
| 3->6 | Unsigned long | Kích thước tệp |
| 7->10 | Zero | Reserved |
| 11->14 | Unsigned long | Địa chỉ phần dữ liệu |

*Bảng : Ý nghĩa từng trường trong vùng Bitmap Header*

Trong phần Bitmap Header có mô tả thông tin về độ dài tệp , thông tin này thực sự cần thiết với mọi chương trình .Tuy nhiên qua thực nghiệm thấy rằng với một số tệp thông tin này không chính xác. Do đó trong báo cáo đưa cách tính kích thước tệp bitmap thông qua công thức:

Tệp\_Size=Sizeof(Bitmap Header ) + Sizeof(Bitmap Infor) + Sizeof(Pallete) + Sizeof(Data)

Địa chỉ vị (offset) của vùng dữ liệu có thể được xác định thông qua công thức :

Địa chỉ vùng data = 54+ Sizeof(Pallete)

Đối với ảnh đen trắng và ảnh màu có số lượng màu lớn hơn 256 thì giá trị địa chỉ vị cố định là 54

**Bitmap Infor**:

Phần bitmap infor dùng để mô tả thông tin về ảnh đang dùng được lưu trữ trong tệp kích thước của phần này cố định là 40 byte.

Ý nghĩa và giá trị của từng trường trong vùng Bitmap Infor được mô tả chi tiết trong bảng :

|  |  |  |
| --- | --- | --- |
| **Offset(byte)** | **Giá Trị** | **Ý nghĩa** |
| 1->4 | 40 | Số lượng byte của vùng bitmap info |
| 5->8 | Unsigned long | Độ rộng của Ảnh tính theo Pixel |
| 9->12 | Unsigned long | Độ cao của Ảnh tính theo Pixel |
| 13->14 | 1 | Number of Color Plans |
| 15->16 | Unsigned long | Số bit để biểu diễn 1 pixel |
| 17->20 | Unsigned long | Kiểu nén |
| 21->24 | Unsigned long | Độ lớn của ảnh (byte) |
| 25->28 | Unsigned long | Độ phân giải của ảnh theo chiều ngang |
| 29->32 | Unsigned long | Độ phân giải của ảnh theo chiều dọc |
| 33->36 | Unsigned long | Số lượng màu trong bảng màu |
| 37->40 | Unsigned long | Số màu quan trọng |

*Bảng : Ý nghĩa từng trường trong vùng BitmapInfor*

**Pallete Table :**

Bảng màu là tập các màu sử dụng trong ảnh , mỗi một màu trong ảnh được gọi là một entry và được lưu trữ bằng 4 byte, mỗi thành phần màu được lưu trữ một byte còn một byte để dự trữ ( chứa dùng) và thứ tự là B,G,R,Reserved. Như vậy kích thước của bảng màu có thể tinh theo công thức:

Sizeof(Pallete Table)=(Number color)\*4

**Data:**

Vùng dữ liệu ảnh là giá trị của điểm ảnh , kích thước của vùng dữ liệu ảnh phụ thuộc vào độ rộng , chiều cao và kiểu ảnh. . Với ảnh 8 bit màu thì ta có công thức

Sizeof(Data)=Width\*Height

Với ảnh có số màu lớn hơn 256 ta có công thức

Sizeof(Data)=Width\*Height\*Bit\_Number\_of\_Pixel

Vậy tại vùng Data là giá trị các thành phần màu cơ bản , số lượng bit dùng để biểu diễn giá trị cho từng thành phần màu có thể sẽ khác nhau phụ thuộc vào chất lượng ảnh. Ảnh 24 bit mỗi thành phần màu đựơc lưu trữ bởi 8 bit và thứ tự lưu trữ là B, G, R.

## 2.2 Thuật toán LSB ( Least Significant Bit)

Phần này sẽ trình bày một thuật toán giấu tin trong ảnh bitmap 24 bit,giải thuật này sẽ thay thế tuần tự từng bit của thông điệp cần ẩn bằng một bit ít có ý nghĩa nhất của 1 byte trong ảnh gốc. Ít có ý nghĩa nhất ở đây được hiểu là việc thay thế bit đó sẽ làm thay đổi giá trị của byte gốc là ít nhất,trong trường hợp này bit đó là bit đầu tiên của byte (có vị trí 0).

Ví dụ: Giả sử ta muốn giấu chữ A (mã ASCII là 65 hay 01000001) vào trong 8 byte của file gốc (môi trường) ta làm như sau:

|  |  |  |
| --- | --- | --- |
| 8 byte ban đầu | Byte cần giấu (A) | 8 byte sau khi giấu |
| 01001001 | 0 | 01001000 |
| 11010111 | 1 | 11010111 |
| 11001100 | 0 | 11001100 |
| 10110101 | 0 | 10110100 |
| 00100100 | 0 | 00100100 |
| 00100101 | 0 | 00100100 |
| 00100000 | 0 | 00100000 |
| 00001010 | 1 | 00001011 |

Như phần trên đã trình bày,1 ảnh bitmap là 1 ma trận các pixel, mỗi pixel bao gồm 3 thành phần màu cơ bản là Red,Green và Blue. Mỗi thành phần này được biểu diễn bởi 1 byte (có giá trị từ 0->255),và đối với mỗi byte này ta sẽ sử dụng bit đầu tiên để thay thế bằng 1 bit của thông điệp cần ẩn.

Các bước thực hiện như sau:

+**Giấu tin (Encode):**

* Biến đổi thông điệp cần ẩn thành dãy các bit, giả sử có dạng: b1b2........bn
* Đọc lần lượt từng byte từ file đầu vào giả sử byte này có dạng:a7….. a1a0, thay thế bit đang xét ,giả sử là bk (với k=1..n) vào vị trí của a0, được byte mới có dạng a7….. a1bk.
* Viết byte vừa thu được vào file ảnh đầu ra

**+Phục hồi tin (Decode):**

* Đọc từng byte từ file ảnh cần giải mã giả sử byte này có dạng a7….. a1a0,tách lấy bit a0, đưa vào mảng bit B
* Từ mảng B biến đổi thành thông điệp ban đầu.

# CHƯƠNG 3:

# THỰC HÀNH XÂY DỰNG ỨNG DỤNG

**Để xây dựng chương trình,ta cần thiết kế các module như sau:**

* Xây dựng 1 class có tên là CryptoHelper.cs chứa 2 phương thức

1. public static byte[] Encrypt(byte[] message,string password)

Phương thức này nhận vào 2 tham số gồm: thông điệp mà ta muốn mã hóa và mật khẩu để trao đổi tin giữa người gửi và người nhận. Phương thức này sẽ biến đối mật khẩu thành 1 khóa có độ dài 128 byte thông qua lớp PasswordDeriveBytes của Microsoft.NET nhằm tăng tính bảo mật. Sau đó kết hợp trộn giữa thông điệp và khóa này bằng toán tử XOR để tạo ra mảng byte là cái ta thực sự đem giấu vào trong bức ảnh.

1. public static byte[] Decrypt(byte[] message, string password)

Phương thức này là quá trình giải mã ngược của phương thức trên,tham số thứ nhất là mảng byte sau khi đã trích ra được từ file ảnh cần giải mã,tham số thứ 2 là mật khẩu để lấy thông tin do người nhận nhập vào,với mật khẩu này,ta cũng dùng lớp PasswordDeriveBytes của Microsoft.NET để tạo ra 1 khóa 128byte (lưu ý rằng nếu người dùng nhập đúng mật khẩu,thì nó cũng sẽ sinh ra cùng 1 khóa như lúc ta mã hóa). Dùng phép XOR giữa mảng byte và khóa này ta sẽ nhận được thông điệp gốc ban đầu.

* Xây dựng 1 class có tên là LSBHelper.CS chứa 2 phương thức

1. public static void Encode(FileStream inStream,byte[] Message,FileStream outStream)

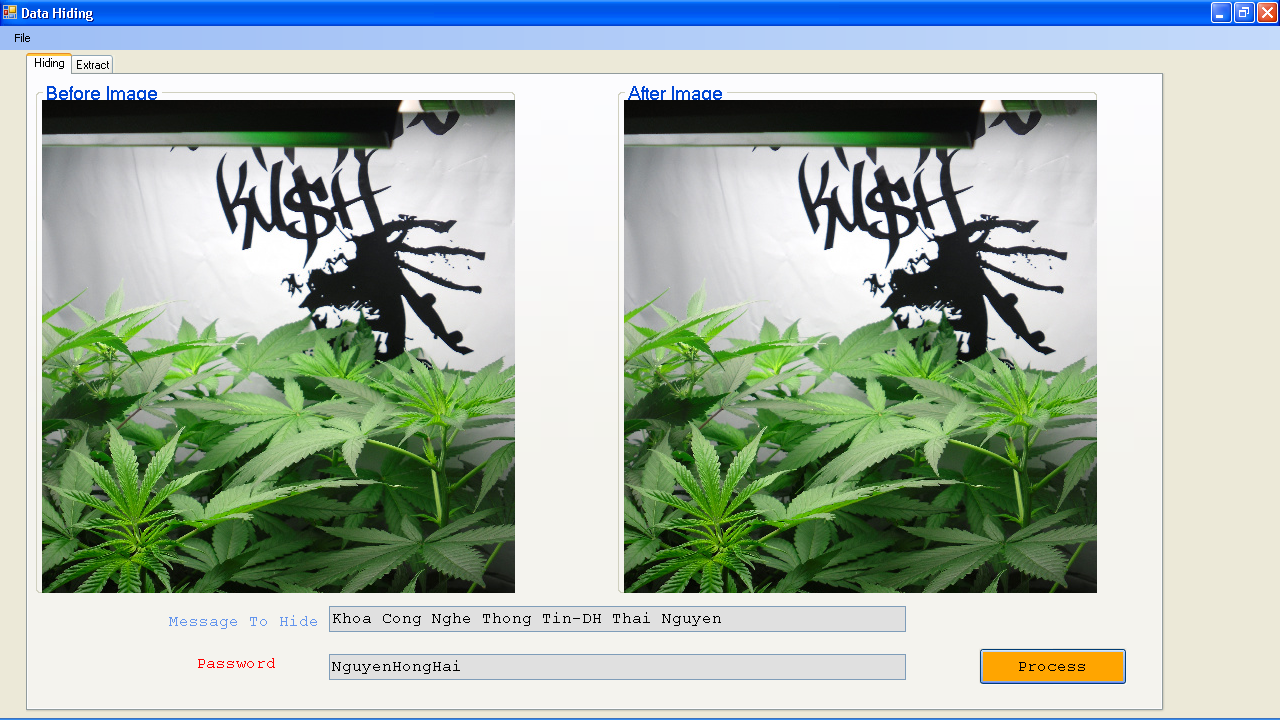
Phương thức này nhận vào 3 tham số: file ảnh đầu vào,mảng các byte cần đưa vào trong ảnh,và ảnh đầu ra.

1. public static byte[] Decode(FileStream inStream,int length)

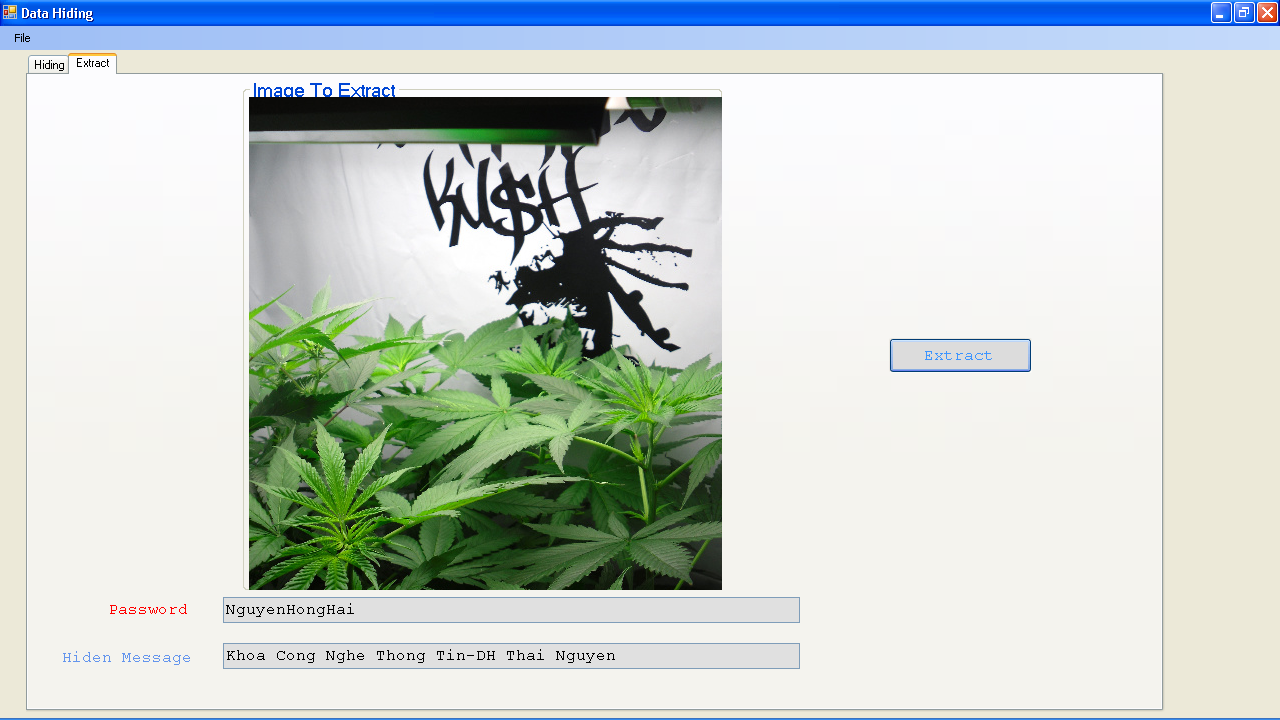
Phương thức này nhận vào 1 file ảnh có chứa tin giấu, độ dài của thông điệp chứa trong đó. Kết quả trả về là 1 mảng các byte mà ta đã giấu trong thủ tục Encode ở trên.

**Kết quả thực nghiệm:**

Chạy chương trình với 1 ảnh đầu vào cỡ 479\*639 (hình bên trái),và thông điệp cần giấu là “Khoa Cong Nghe Thong Tin-DH Thai Nguyen” ta được 1 ảnh đầu ra như hình bên phải.

*****Hình 6:Quá trình giấu tin*

Thực hiện trích xuất thông điệp từ bức ảnh trên ta được thông điệp như ban đầu.

****

*Hình 7:Quá trình giải mã*

# KẾT LUẬN

Giấu tin trong ảnh là 1 cách tiếp cận khá mới mẻ trong an toàn và bảo mật thông tin,trong đề tài này em đã đưa ra một thuật toán áp dụng trong ảnh bitmap 24 bit,cùng cài đặt và demo chương trình. Tuy nhiên chương trình vẫn còn nhiều hạn chế như mật độ giấu tin chưa cao,khi độ dài của thông điệp quá lớn so với dung lượng của bức ảnh thì sẽ không thực hiện được. Song đây cũng là cơ sở để tiếp tục phát triển chương trình cho phép thực hiện với các định dạng ảnh khác như GIF,JPEG…..cũng như các thuật toán mã hóa khác sao cho tỉ lệ giấu tin đạt hiệu quả cao hơn.

Em xin được gửi lời biết ơn chân thành tới PGS. TS. Nguyễn Văn Tảo, người đã định hướng nghiên cứu và hướng dẫn em thực hiện đề tài này!

Thái Nguyên,ngày 18,tháng 09,năm 2010

Sinh Viên: Nguyễn Hồng Hải

# TÀI LIỆU THAM KHẢO

# Tiếng Việt:

# [1]: Giấu tin trong ảnh nhị phân và ứng dụng, PGS.TS Phạm Văn Ất,ThS Nguyễn Hữu Cường, Khoa công nghệ thông tin, Trường ĐH giao thông vận tải.

# [2]: Một số thuật toán giấu tin và áp dụng giấu tin mật trong ảnh. PGS.TS Nguyễn Văn Tảo, Đỗ Trung Tuấn, Bùi Thế Hồng báo cáo tại hội thảo RDA 8.

# Tiếng Anh:

# [1]: Fridrich, J.: A New Steganographic Method for Palette-Based Images. Proc. of the IS&T PICS conference, April 1998, Savannah, Georgia (1998) 285−289.

# [2]: Digital Steganography: *Hiding data within Data*. Donovan Artz

# [3]: Information Hiding: Steganography & Digital Watermarking *http://www.jjtc.com/Steganography/*