**KỸ THUẬT GIẤU TIN TRONG ẢNH BẰNG THUẬT TOÁN LSB**

**Sinh viên thực hiện: Nguyễn Chiến Thắng**

**Lớp:D14CQAT01**

**MSSV:N14DCAT090**

# MỞ ĐẦU

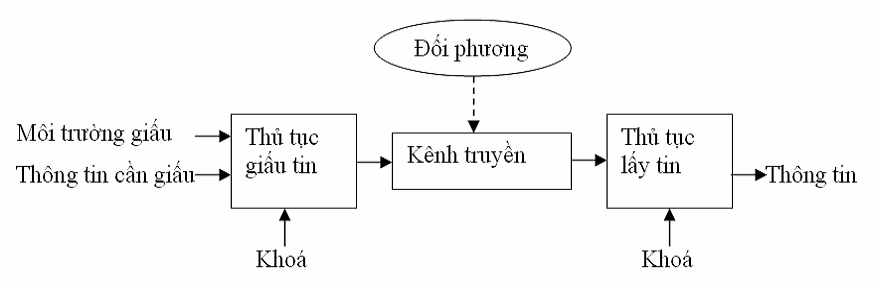
**Công nghệ thộng tin ngày càng phát triển , điều đó đồng nghĩa với việc càng ngày thông tin được biểu diễn dưới dạng kỹ thuật số càng nhiều ,chúng được biểu diễn và truyền tải trong môi trường thế giới số do đó vấn đề bảo mật thông tin trong môi trường thế giới số ngày càng trở nên quan trọng. Đã có 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….**

# TỔNG QUAN VỀ GIẤU TIN

## **Mô hình giấu tin**

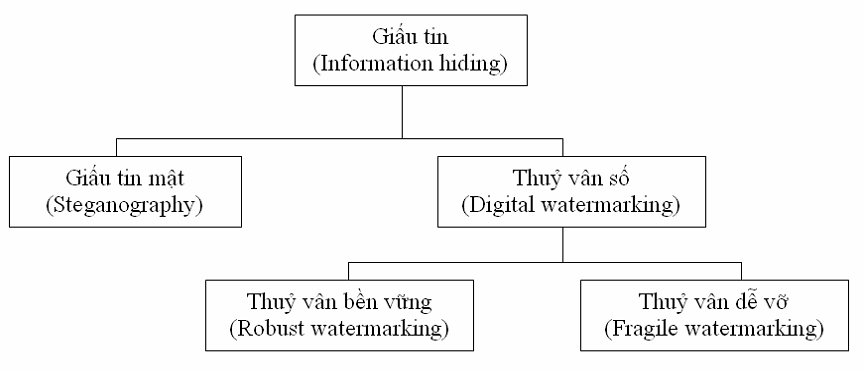
**Để thực hiện giấu tin cần xây dựng được các thủ tục giấu tin. Các thủ tục này sẽ thực hiện nhúng thông tin cần giấu vào môi trường giấu tin. Các thủ tục giấu tin thường được thực hiện với một khóa giống như trong các hệ mật mã để tăng tính bảo mật. Sau khi 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.**

# CƠ SỞ LÝ THUYẾT

## **2.1 Một số lý thuyết**

* **Một bức ảnh sẽ có chiều rộng là W, chiều cao là H nên số điểm ảnh (pixel) là W x H, chúng ta sẽ có một ma trận các pixel có H dòng và W cột (W x H)**
* **Mỗi pixel có một giá trị màu RGB (Red – Green – Blue), mỗi giá trị R – G – B có giá trị từ 0 -> 255 tương ứng với một Byte trong lưu trữ, suy ra có tất cả 2563tổ hợp màu khác nhau có thể được tạo ra.**

**Ví dụ: Màu đỏ R có giá trị RGB là (255, 0, 0); Xanh lá cây G (0,255,0); màu vàng Yello (255,255,0);….**

* **Mỗi Byte gồm 8 Bit, Bit có chỉ số nhỏ nhất gọi là LSB (Least Significant Bit).**

**Ví dụ: Giá trị 147 chuyển sang nhị phân là 10010011 sẽc có LSB là 1. Giá trị 200 chuyển sang nhị phân là 11001000 sẽ có LSB là 0**

**Ý tưởng của thuật toán này là sẽ thay thế các bit LSB của từng byte trong giá trị RGB của từng Pixel. Sỡ dĩ chúng ta chỉ thay thế các bit LSB là vì sau khi thay đổi các giá trị trong RBG chỉ dao động 1 đơn vị, dùng mắt thường sẽ khó phát hiện ra sự thay đổi của bức ảnh, giúp đạt được mục đích ẩn dấu.**

## **2.2 Thuật toán LSB ( Least Significant Bit)**

**Phần này sẽ trình bày một thuật toán giấu tin trong ảnh,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 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):**

* **Đầu tiên chúng ta chuyển đoạn thông điệp bí mật sang dạng nhị phân (Text bits).**
* **Duyệt các pixel trong bức ảnh (Original Image), đọc các giá trị R – G - B của từng pixel một.**
* **Thay LSB của từng giá trị R – G – B với một Bit trong chuỗi thông điệp bí mật. Lưu bức ảnh sau khi đã thay đổi lại thành một tập tin mới (Modified Image)**
* **Giả sử Pixel tại (100,100) có giá trị màu là (215,162,10) BufferedImage buff = ImageIO.read(new File(“input.png”)); int rgb = buff.getRGB(100,100); Ta phân tích biến rgb trả về như bên dưới**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte** | **Byte** | **Byte** | **Byte** |
| **Alpha** | **Red** | **Green** | **Blue** |
| **XXX** | **215** | **162** | **10** |
| **XXX** | **11010111** | **10100010** | **00001010** |

* **Dữ liệu cần ẩn: “Secret” <=> 101001111001011100011111001011001011110100  
  Ví dụ ẩn 3 bit đầu tiên là 101 vào ảnh:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **XXX** | **11010111** | **10100010** | **00001010** | **RGB của Pixel trên ảnh gốc** |
|  | **OR** | **AND** | **OR** | **Phép toán** |
| **XXX** | **00000001** | **11111110** | **00000001** | **BitMask để thay đổi LSB** |
| **XXX** | **0x01** | **0xFE** | **0x01** | **Dạng hexa** |
|  | **=** | **=** | **=** |  |
| **XXX** | **11010111** | **10100010** | **00001011** | **RGB đã thay đổi** |
| **XXX** | **215** | **162** | **11** |  |

* **Source Code:**
* **int rgb = buff.getRGB(x, y); //Giá trị RGB hiện tại**
* **//Chỉnh sửa giá trị R**
* **if(binHidden[i++] == '0')rgb = rgb & 0xFFFEFFFF);**
* **else rgb = (rgb | 0x00010000);**
* **//Chỉnh sửa giá trị G**
* **if(binHidden[i++] == '0')rgb = rgb & 0xFFFFFEFF);**
* **else rgb = (rgb | 0x00000100);**
* **//Chỉnh sửa giá trị B**
* **if(binHidden[i++] == '0')rgb = rgb & 0xFFFFFFFE);**
* **else rgb = (rgb | 0x00000001);**
* **buff.setRGB(x, y, rgb); //Cập nhật lại RGB cho pixel này**

**+Phục hồi tin (Decode):**

* **Duyệt từng pixel của bức ảnh nhận được (Modified Image).**
* **Đọc giá trị R – G – B của từng pixel, lấy các LSB của từng giá trị R – G – B, ta được một chuỗi nhị phân của thông điệp gốc.**
* **Chuyển chuỗi nhị phân sang mã ASCII đọc được.**

**Để việc lấy ra các bit LSB đã Encode trước đó một cách đơn giản ta có thể sử dụng các thư viện hỗ trợ sau:**

* **Lớp java.awt.Color hỗ trợ làm việc với giá trị màu dễ dàng hơn từ một giá trị RGB kiểu integer lấy được từ phương thức getRGB(x, y)**
* **Phương thức toBinaryString(int value) của lớp Integer để lấy chuỗi nhị phân của biến value**

**Source Code:**

**int rgb = buff.getRGB(100,100); //Lấy một RGB(215, 162, 11)**

**Color c = new Color(rgb); //Tạo đối tượng Color từ một giá trị RGB**

**int R = c.getRed(); // Trả về 215**

**int G = c.getGreen(); // Trả về 162**

**int B = c.getBlue(); // Trả về 11**

**String rString = Integer.toBinaryString(R); // Trả về 11010111**

**String gString = Integer.toBinaryString(G); // Trả về 10100010**

**String bString = Integer.toBinaryString(B); // Trả về 00001011**

**Từ đây ta sẽ lấy được giá trị ẩn ban đầu là “101”**

# THỰC HÀNH XÂY DỰNG ỨNG DỤNG

**Để xây dựng chương trình,ta cài đặt code bằng Java trên NetBean. Một số thư viện, lớp và hàm hỗ trợ của Java ta sẽ sử dụng:**

**+Lớp javax.imageio.ImageIO: Hỗ trợ các phương phức đọc ghi các tập tin hình ảnh.  
+Lớp java.awt.image.BufferedImage: Dữ liệu từ ảnh sẽ được đưa vào một đối tượng kiểu BufferedImage, mọi thay đổi về dữ liệu của tập tin hình ảnh sẽ được thực hiện trên đối tượng này.  
+Hàm getRGB(x, y) của đối tượng BufferedImage giúp lấy giá trị RBG của một pixel tại tọa độ (x,y). Hàm này trả về một số nguyên 4 byte  
+Hàm setRGB(x, y, value) của đối tượng BufferedImage giúp đặt lại giá trị RBG cho một pixel tại tọa độ (x,y).**

**Trong Source Code có 2 Class:**

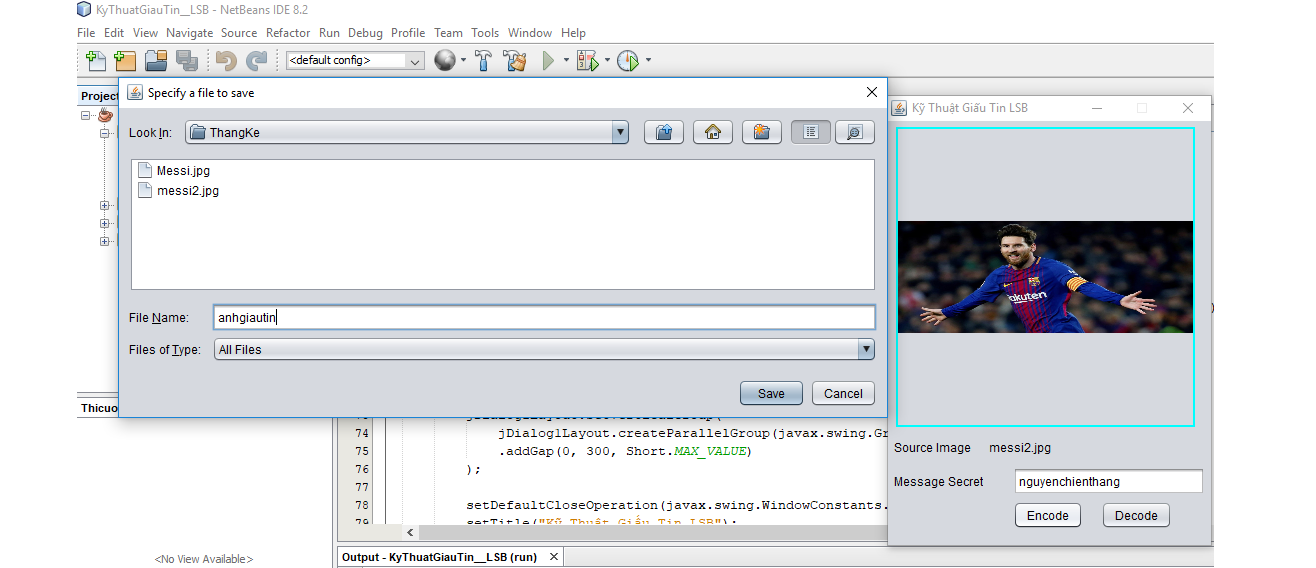
* **GiauTin.java chứa code cung cấp các phương thức Encode/Decode và các phương thức hỗ trợ**
* **GiaoDien.java: Cài đặt giao diện đồ họa cho chương trình**

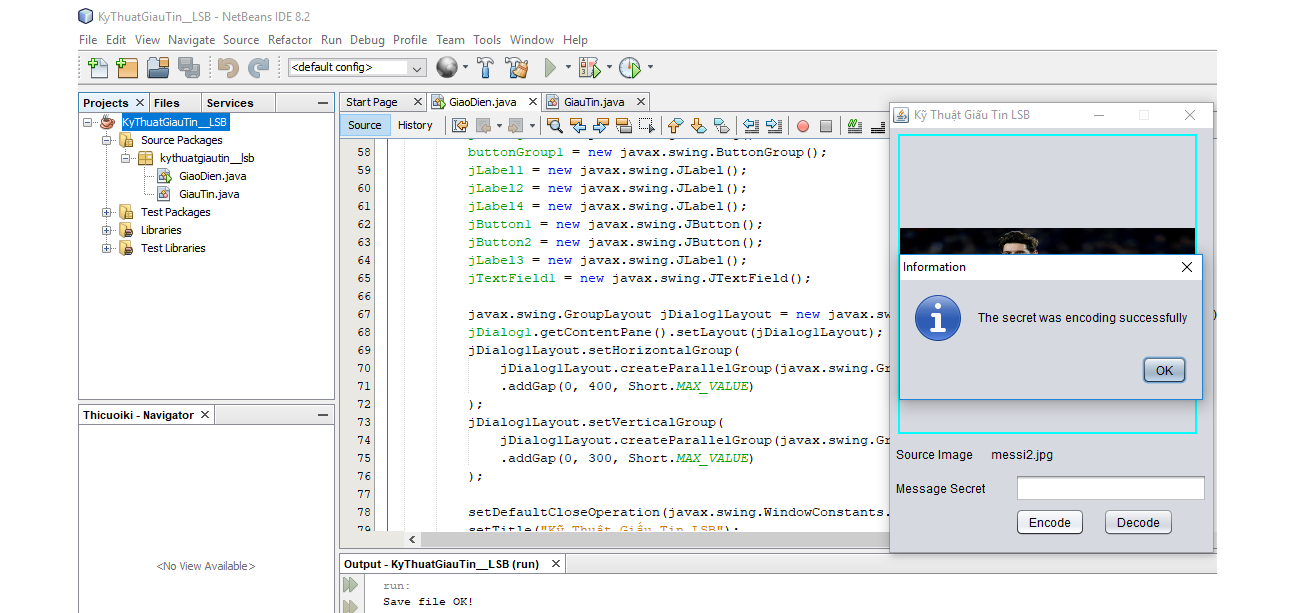
**Kết quả thực nghiệm:**

**Chạy chương trình và click chọn 1 ảnh đầu vào (hình bên trên),và điền thông điệp cần giấu là “nguyenchienthang”.**

**Ta click Encode để mã hóa thông điệp cần giấu và Save ảnh chứa thông điệp với 1 Modified Image mới có tên là “anhgiautin”.**

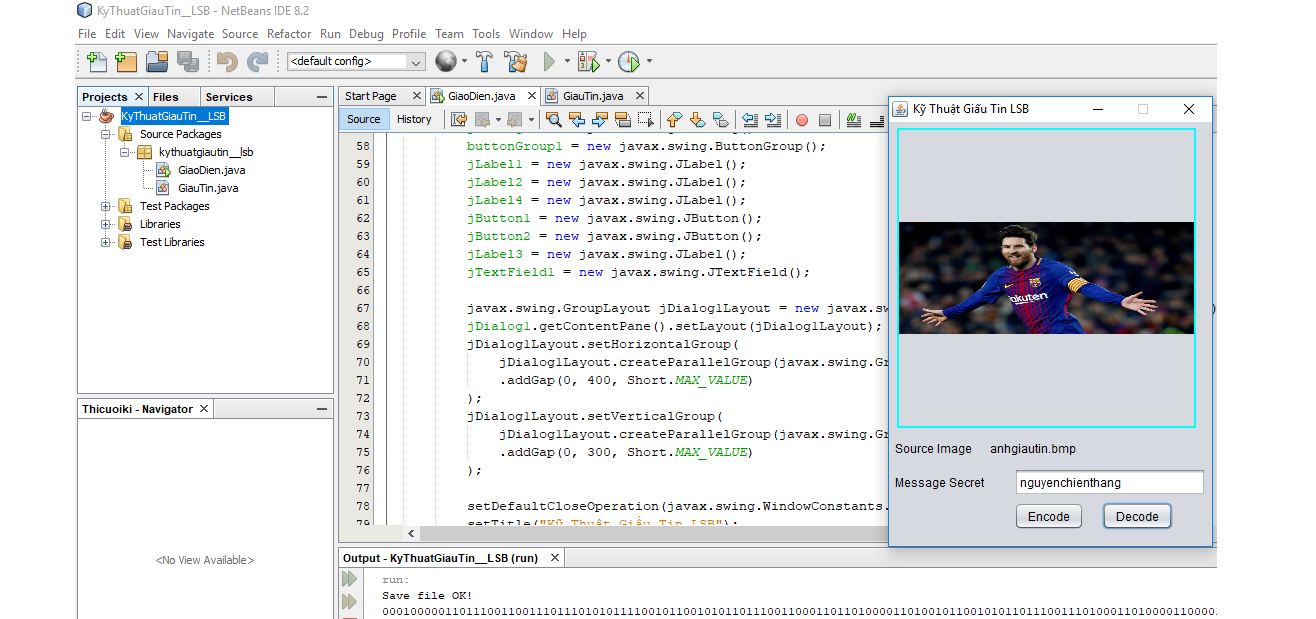
**Sau khi ta Save tên ảnh được thông báo như (hình bên dưới).**

****

****

***Hình 6:Quá trình giấu tin***

**Thực hiện trích xuất thông điệp từ bức ảnh trên bằng cách click chọn ảnh đã Save với tên “anhgiautin” rồi click Decode 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,cùng cài đặt và demo chương trình. Tuy nhiên chương trình vẫn còn nhiều hạn chế như mật độ giấu tin chưa cao, việc kiểm soát dung lượng tập tin hình ảnh sau khi thực hiện Encode: Các tập tin hình ảnh nguồn có các định dạng khác nhau (BMP, PNG, JPG,...), việc đọc dữ liệu vào một Buffered Image sẽ gỡ bỏ các chuẩn nén từ trước của hình ảnh, đặt biệt là JPG/JPEG nên sau khi Encode, tập tin tạo ra sẽ gặp một số vấn đề về dung lượng, nếu muốn tập tin không thay đổi dung lượng và không bị mất mát dữ liệu ta phải tìm hiểu thật kỹ về các chuẩn PNG, JPEG,... 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…..cũng như các thuật toán mã hóa khác sao cho tỉ lệ giấu tin đạt hiệu quả cao hơn.**