TRƯỜNG ĐẠI HỌC LÂM NGHIỆP

KHOA CƠ ĐIỆN VÀ CÔNG TRÌNH

----------o0o----------

A logo with green and orange colors

AI-generated content may be incorrect.

BÁO CÁO TIỂU LUẬN

MÔN: AN TOÀN VÀ BẢO MẬT THÔNG TIN

ĐỀ TÀI: NHÚNG DỮ LIỆU VÀO ẢNH BMP

|  |  |
| --- | --- |
| *Giảng viên hướng dẫn*  *Họ tên*  *Lớp* | : ThS. Mai Hà An  : Đỗ Thành Công  : 2174801041054  : Đoàn Quang Khải  : 217480104  : K66-HTTT |

HÀ NỘI – 2025

MỤC LỤC

[I. Giới thiệu đề tài 2](#_Toc195471255)

[1.1. Lý do chọn đề tài 2](#_Toc195471256)

[1.2.Mục tiêu đề tài 3](#_Toc195471257)

[II. Cơ sở lý thuyết 4](#_Toc195471258)

[2.1. Steganography là gì? 4](#_Toc195471259)

[2.2. Kỹ thuật giấu dữ liệu trong ảnh (Least Significant Bit - LSB) 4](#_Toc195471260)

[III. Phân tích thiết kế 6](#_Toc195471261)

[3.1. Phân tích yêu cầu 6](#_Toc195471262)

[3.2. Thiết kế giao diện 6](#_Toc195471263)

[IV. Thuật toán và cài đặt 7](#_Toc195471264)

[4.1.Mô tả thuật toán 7](#_Toc195471265)

[4.1.1. Thuật toán mã hóa 7](#_Toc195471266)

[4.1.2. Thuật toán giải mã 8](#_Toc195471267)

[4.2. Cài đặt chương trình 9](#_Toc195471268)

[V. Kết quả và đánh giá 10](#_Toc195471269)

[5.1. Kết quả thực nghiệm 10](#_Toc195471270)

[5.2. Đánh giá – Nhận xét 10](#_Toc195471271)

[VI. Kết luận 14](#_Toc195471272)

# I. Giới thiệu đề tài

## 1.1. Lý do chọn đề tài

Trong thời đại số hóa, khi các giao dịch trực tuyến, trao đổi thông tin cá nhân và dữ liệu quan trọng diễn ra liên tục, yêu cầu về bảo mật thông tin ngày càng trở nên cấp thiết. Các phương pháp mã hóa truyền thống như AES hay RSA đã được sử dụng rộng rãi, nhưng vẫn tồn tại nguy cơ bị phát hiện hoặc tấn công nếu dữ liệu mã hóa được truyền trực tiếp qua các kênh không an toàn. Steganography (giấu tin) ra đời như một giải pháp bổ sung, cho phép ẩn dữ liệu nhạy cảm trong các tệp phương tiện như hình ảnh, âm thanh hoặc video mà không làm thay đổi đáng kể đặc điểm của tệp gốc.

Trong số các kỹ thuật giấu tin, phương pháp Least Significant Bit (LSB) nổi bật nhờ tính đơn giản và hiệu quả khi áp dụng trên ảnh kỹ thuật số. LSB cho phép nhúng dữ liệu vào các bit ít quan trọng nhất của pixel mà không gây ra sự thay đổi rõ rệt về mặt thị giác. Đặc biệt, định dạng ảnh BMP (Bitmap) không sử dụng nén, giúp bảo toàn dữ liệu nhúng và trở thành lựa chọn lý tưởng cho các ứng dụng steganography cơ bản.

Việc nghiên cứu và phát triển một phần mềm giấu tin sử dụng kỹ thuật LSB không chỉ giúp củng cố kiến thức về xử lý ảnh và lập trình mà còn mở ra cơ hội khám phá các ứng dụng thực tế trong bảo mật thông tin. Vì vậy, đề tài này được chọn nhằm xây dựng một công cụ đơn giản nhưng hiệu quả, phục vụ mục đích học tập và nghiên cứu về steganography.

## 1.2.Mục tiêu đề tài

Xây dựng phần mềm mã hóa và giải mã dữ liệu văn bản vào ảnh BMP: Phát triển một ứng dụng cho phép người dùng nhúng chuỗi văn bản vào ảnh BMP và trích xuất lại dữ liệu một cách chính xác.

Hiển thị trực quan: Cung cấp giao diện thân thiện, hiển thị ảnh trước và sau khi mã hóa để người dùng dễ dàng quan sát và so sánh.

Đảm bảo tính bảo mật cơ bản: Dữ liệu nhúng không làm thay đổi rõ rệt hình ảnh gốc và chỉ có thể được giải mã bởi những người biết phương pháp.

Khả năng mở rộng: Tạo nền tảng để có thể nâng cấp thêm các tính năng như hỗ trợ định dạng ảnh khác hoặc kết hợp với các thuật toán mã hóa nâng cao trong tương lai.

# II. Cơ sở lý thuyết

## 2.1. Steganography là gì?

Steganography là kỹ thuật giấu dữ liệu bí mật trong các tệp phương tiện (như hình ảnh, âm thanh, video) sao cho sự tồn tại của dữ liệu đó không bị phát hiện bởi mắt thường hoặc các phương pháp kiểm tra thông thường. Khác với mã hóa (cryptography) – tập trung vào việc làm cho dữ liệu trở nên không thể đọc được nếu không có khóa – steganography hướng đến việc che giấu sự tồn tại của chính dữ liệu.

Một hệ thống steganography cơ bản bao gồm các thành phần sau:

* Dữ liệu bí mật (message): Thông tin cần giấu, có thể là văn bản, hình ảnh, hoặc tệp khác.
* Tệp chứa (cover): Tệp phương tiện gốc, chẳng hạn như ảnh BMP, được sử dụng để ẩn dữ liệu.
* Tệp kết quả (stego): Tệp sau khi nhúng dữ liệu bí mật, trông giống hệt tệp gốc về mặt thị giác.
* Khóa (nếu có): Thông tin bổ sung để tăng tính bảo mật hoặc hỗ trợ giải mã.

- Ưu điểm của steganography là khả năng ẩn dữ liệu mà không thu hút sự chú ý, phù hợp cho các ứng dụng như truyền thông bí mật hoặc bảo vệ bản quyền.

## 2.2. Kỹ thuật giấu dữ liệu trong ảnh (Least Significant Bit - LSB)

Kỹ thuật LSB (Least Significant Bit) là một trong những phương pháp phổ biến nhất trong steganography áp dụng cho ảnh kỹ thuật số. Ý tưởng chính là thay thế các bit ít quan trọng nhất trong giá trị màu của pixel bằng các bit của dữ liệu bí mật.

Nguyên lý hoạt động:

* Mỗi pixel trong ảnh màu (RGB) được biểu diễn bằng ba kênh màu: Red (R), Green (G), Blue (B), mỗi kênh thường có giá trị từ 0 đến 255 (8 bit).
* Bit cuối cùng (LSB) của mỗi kênh có ảnh hưởng rất nhỏ đến giá trị màu tổng thể. Ví dụ, thay đổi giá trị R từ 255 (11111111) thành 254 (11111110) hầu như không thể nhận ra bằng mắt thường.
* LSB tận dụng đặc điểm này để nhúng dữ liệu bí mật bằng cách thay thế bit cuối cùng của một kênh màu (thường là R) bằng bit của dữ liệu cần giấu.

Ví dụ minh họa:

* Giả sử cần giấu chuỗi "A" (mã ASCII: 65, nhị phân: 01000001) vào ảnh.
* Mỗi bit của chuỗi nhị phân sẽ được nhúng vào LSB của giá trị R của 8 pixel liên tiếp.
* Nếu giá trị R của pixel là 10010110, nhúng bit 0 sẽ giữ nguyên giá trị, còn nhúng bit 1 sẽ đổi thành 10010111.

Ưu điểm của LSB:

* Đơn giản, dễ triển khai.
* Không làm thay đổi rõ rệt chất lượng ảnh.
* Phù hợp với các định dạng không nén như BMP.

Nhược điểm:

* Dễ bị phát hiện bởi các công cụ phân tích thống kê (như kiểm tra phân bố LSB).
* Dung lượng dữ liệu nhúng phụ thuộc vào kích thước ảnh.

# III. Phân tích thiết kế

## 3.1. Phân tích yêu cầu

Để xây dựng phần mềm giấu tin sử dụng kỹ thuật LSB, đề tài cần đáp ứng các yêu cầu chức năng và phi chức năng sau:

Chức năng chính:

* Chọn ảnh BMP: Cho phép người dùng tải ảnh định dạng BMP từ máy tính làm tệp chứa (cover image).
* Nhập văn bản cần mã hóa: Cung cấp trường nhập liệu để người dùng nhập chuỗi văn bản muốn giấu.
* Mã hóa dữ liệu: Nhúng chuỗi văn bản vào ảnh BMP bằng kỹ thuật LSB và lưu ảnh kết quả (stego image).
* Giải mã dữ liệu: Trích xuất chuỗi văn bản từ ảnh đã mã hóa và hiển thị lại nội dung gốc.
* Hiển thị ảnh: Hiển thị ảnh gốc và ảnh sau mã hóa trong giao diện để người dùng quan sát.

## 3.2. Thiết kế giao diện

Giao diện người dùng (UI) được thiết kế sử dụng Windows Forms, tập trung vào tính trực quan và dễ sử dụng, phù hợp cho cả người dùng không chuyên. Các thành phần chính bao gồm:

* RadioButton:
* Hai lựa chọn: Mã hóa (Encode) và Giải mã (Decode).
* Giúp người dùng chuyển đổi nhanh giữa hai chế độ.
* TextBox nhập dữ liệu:
* Trường nhập văn bản cần mã hóa (chế độ mã hóa).
* Hiển thị kết quả văn bản sau khi giải mã (chế độ giải mã).
* TextBox đường dẫn ảnh:
* Hiển thị đường dẫn của tệp ảnh BMP được chọn.
* Có thể nhập trực tiếp hoặc chọn qua nút “Chọn ảnh”.
* PictureBox:
* Hiển thị ảnh BMP đã chọn (gốc hoặc sau mã hóa).
* Cho phép người dùng quan sát trực quan chất lượng ảnh.
* Button “Chọn ảnh”: Mở hộp thoại để người dùng chọn tệp BMP từ máy tính.
* Button “Thực hiện”: Thực thi quá trình mã hóa hoặc giải mã tùy thuộc vào chế độ đã chọn.
* Label thông báo: Hiển thị trạng thái (thành công, lỗi) hoặc thông tin bổ sung (ví dụ: “Dữ liệu quá dài”).
* Bố cục giao diện:
* Các thành phần được sắp xếp gọn gàng, sử dụng bảng màu trung tính.
* PictureBox chiếm phần lớn không gian để hiển thị ảnh rõ ràng.
* Các nút điều khiển được đặt ở vị trí dễ tiếp cận, kèm chú thích rõ ràng.
* Mục tiêu thiết kế:
* Đơn giản hóa thao tác: Người dùng chỉ cần 2-3 bước để hoàn thành mã hóa/giải mã.
* Trực quan: Kết quả (ảnh, văn bản) được hiển thị ngay trên giao diện.
* Phù hợp với người dùng không có kiến thức chuyên sâu về lập trình hoặc steganography.

# IV. Thuật toán và cài đặt

## 4.1.Mô tả thuật toán

### 4.1.1. Thuật toán mã hóa

* Input:
* Ảnh BMP (cover image).
* Chuỗi văn bản cần giấu.
* Đường dẫn lưu ảnh kết quả (stego image).
* Output: Ảnh BMP chứa dữ liệu ẩn (stego image).

Các bước thực hiện:

* Chuyển đổi văn bản thành nhị phân:
* Mỗi ký tự trong chuỗi được chuyển thành mã ASCII (8 bit).
* Ví dụ: Chuỗi “Hi” → “H” (01001000), “i” (01101001).
* Kiểm tra kích thước ảnh:
* Tính số pixel cần thiết: số ký tự × 8 + 32 (32 bit để lưu độ dài chuỗi).
* Nếu số pixel khả dụng nhỏ hơn yêu cầu, thông báo lỗi.
* Nhúng dữ liệu vào pixel:
* Duyệt qua từng pixel của ảnh theo thứ tự (từ trái sang phải, trên xuống dưới).
* Với mỗi bit của chuỗi nhị phân:
* Lấy giá trị R (Red) của pixel.
* Thay bit LSB của R bằng bit dữ liệu (0 hoặc 1).
* Cập nhật giá trị R mới cho pixel.
* Lưu độ dài chuỗi:
* Chuyển độ dài chuỗi (số ký tự) thành nhị phân (32 bit).
* Nhúng 32 bit này vào các pixel cuối cùng của ảnh.
* Lưu ảnh kết quả: Ghi ảnh đã chỉnh sửa thành tệp BMP mới.
* Ví dụ:
* Văn bản: “A” (ASCII: 65, nhị phân: 01000001).
* Pixel 1: R = 10010110, bit cần nhúng = 0 → R = 10010110.
* Pixel 2: R = 11110000, bit cần nhúng = 1 → R = 11110001.

### 4.1.2. Thuật toán giải mã

* Input: Ảnh BMP chứa dữ liệu ẩn (stego image).
* Output: Chuỗi văn bản gốc.

Các bước thực hiện:

* Đọc độ dài chuỗi:
* Lấy 32 pixel cuối cùng của ảnh.
* Trích xuất bit LSB của kênh R để ghép thành số nhị phân (32 bit).
* Chuyển số này thành số nguyên, biểu thị số ký tự (N).
* Trích xuất dữ liệu:
* Tính số bit cần đọc: N × 8.
* Duyệt qua N × 8 pixel đầu tiên.
* Lấy bit LSB của kênh R, ghép thành chuỗi nhị phân.
* Chuyển đổi về văn bản:
* Cắt chuỗi nhị phân thành các đoạn 8 bit.
* Chuyển mỗi đoạn thành ký tự ASCII.
* Hiển thị kết quả: Trả về chuỗi văn bản gốc.
* Ví dụ:
* Pixel 1: R = 10010110 → LSB = 0.
* Pixel 2: R = 11110001 → LSB = 1.
* Ghép 8 bit: 01000001 → “A”.

## 4.2. Cài đặt chương trình

* Ngôn ngữ lập trình: C#
* Lý do: Hỗ trợ tốt cho xử lý ảnh, lập trình giao diện và các thư viện liên quan.
* Môi trường phát triển: Visual Studio 2022
* Cung cấp công cụ mạnh mẽ để thiết kế giao diện Windows Forms và debug.
* Công nghệ sử dụng: Windows Forms: Phù hợp cho ứng dụng giao diện đơn giản, nhẹ và dễ triển khai.
* Các thư viện liên quan:
* System.Drawing: Xử lý ảnh BMP (đọc, ghi pixel).
* System.IO: Quản lý tệp (đọc/ghi ảnh, đường dẫn).
* System.Text: Xử lý chuỗi và chuyển đổi nhị phân.
* Cấu trúc chương trình:
* Form chính: Chứa giao diện và logic điều khiển.
* Hàm mã hóa: Thực hiện thuật toán nhúng dữ liệu vào ảnh.
* Hàm giải mã: Trích xuất dữ liệu từ ảnh.

# V. Kết quả và đánh giá

## 5.1. Kết quả thực nghiệm

* Môi trường thử nghiệm:
* Máy tính: Windows 10, RAM 8GB, CPU Intel Core i5.
* Ảnh thử nghiệm: BMP 24-bit, kích thước từ 256×256 đến 1920×1080.
* Dữ liệu: Chuỗi văn bản từ 10 đến 10,000 ký tự.
* Kết quả:
* Giao diện:
* Hoạt động mượt mà, các nút chức năng phản hồi nhanh.
* PictureBox hiển thị rõ ràng ảnh gốc và ảnh sau mã hóa.
* Mã hóa:
* Thời gian xử lý: <1 giây cho ảnh 512×512 với 100 ký tự.
* Ảnh kết quả không có sự khác biệt rõ rệt bằng mắt thường.
* Giải mã:
* Dữ liệu trích xuất chính xác 100% nếu ảnh không bị chỉnh sửa.
* Thời gian giải mã tương đương mã hóa.

## 5.2. Đánh giá – Nhận xét

* Ưu điểm:

+ Đơn giản và hiệu quả:

* Thuật toán LSB dễ hiểu, dễ triển khai.
* Phù hợp cho mục đích học tập và ứng dụng cơ bản.
* Giao diện thân thiện:
* Người dùng không cần kiến thức chuyên sâu để sử dụng.
* Hiển thị trực quan giúp kiểm tra kết quả dễ dàng.
* Bảo mật cơ bản:
* Dữ liệu ẩn khó phát hiện nếu không biết phương pháp.
* Có thể nâng cấp thêm lớp mã hóa để tăng bảo mật.
* Hạn chế:
* Giới hạn định dạng: Chỉ hỗ trợ BMP, không phù hợp với các định dạng phổ biến như PNG, JPG.
* Bảo mật chưa cao:
* Kỹ thuật LSB dễ bị phát hiện bởi các công cụ phân tích thống kê.
* Không chống lại được các cuộc tấn công cố ý (như chỉnh sửa ảnh).
* Dung lượng dữ liệu: Phụ thuộc vào kích thước ảnh, không phù hợp cho dữ liệu lớn.
* Hướng phát triển:
* Hỗ trợ thêm định dạng: Mở rộng cho PNG, JPG bằng cách xử lý nén và tái tạo dữ liệu.
* Tăng cường bảo mật:
* Kết hợp với mã hóa AES trước khi nhúng dữ liệu.
* Sử dụng các bit ngẫu nhiên thay vì chỉ LSB.
* Mở rộng dữ liệu ẩn: Cho phép ẩn file (PDF, ZIP) hoặc hình ảnh thay vì chỉ văn bản.
* Cải thiện giao diện:
* Thêm tính năng xem trước dung lượng nhúng tối đa.
* Hỗ trợ kéo-thả tệp ảnh.

A screenshot of a computer

AI-generated content may be incorrect.

Giao diện ban đầu

Screens screenshot of a chat

AI-generated content may be incorrect.

Khi thực hiện mã hóa

A screenshot of a computer

AI-generated content may be incorrect.

Khi mã hóa sẽ hiện chỗ lưu ảnh

A screenshot of a computer

AI-generated content may be incorrect.

Nhúng xong sẽ hiển thị nhúng dữ liệu thành công

A screenshot of a computer

AI-generated content may be incorrect.

Giải mã ảnh

# VI. Kết luận

* Đề tài đã đạt được các mục tiêu đề ra:
* Xây dựng thành công phần mềm mã hóa và giải mã dữ liệu văn bản vào ảnh BMP sử dụng kỹ thuật LSB.
* Cung cấp giao diện trực quan, dễ sử dụng, hiển thị rõ ràng kết quả.
* Đảm bảo dữ liệu nhúng không làm thay đổi đáng kể chất lượng ảnh và được trích xuất chính xác.
* Qua quá trình thực hiện, nhóm nghiên cứu đã củng cố kiến thức về:
* Xử lý ảnh: Làm việc với pixel, kênh màu và định dạng BMP.
* Lập trình giao diện: Sử dụng Windows Forms để tạo ứng dụng thân thiện.
* Bảo mật thông tin: Hiểu rõ hơn về steganography và các ứng dụng thực tế.