BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN

AN TOÀN VÀ BẢO MẬT THÔNG TIN

Mã học phần: CT204

Đề tài

TÌM HIỂU VỀ ỨNG DỤNG GIẤU TIN TRONG ẢNH

Mã lớp học phần: CT204/01

Nhóm thực hiện: 01

Giảng viên hướng dẫn: Th.S Phan Bích Chung

Thành viên nhóm		
Họ và tên	**MSSV**	**Lớp**
Lã Thái Hòa	**B2113309**	**DI21Z6A1**
Nguyễn Quốc Thịnh	**B2106815**	**DI21Z6A1**
Lê Tuấn Đạt	**B2113328**	**DI21Z6A2**

Cần Thơ, 3/2025

NHẬN XÉT CỦA GIẢNG VIÊN

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Diễn giải
1	AI	Artificial Intelligence
2	CT204	Mã học phần an toàn bảo mật thông tin
3	LSB	Least Significant Bit
4	BMP	Bitmap (Windows bitmap)
5	PNG	Portable Network Graphics
6	JPEG	Joint Photographic Experts Group
7	DCT	Discrete Cosine Transform
8	DWT	Discrete Wavelet Transform
9	DFT	Discrete Fourier Transform
10	AES	Advanced Encryption Standard
11	RSA	Rivest–Shamir–Adleman (Mã hóa khóa công khai)
12	ECC	Elliptic Curve Cryptography
13	GDPR	General Data Protection Regulation
14	IoT	Internet of Things
15	DDoS	Distributed Denial of Service
16	Phishing	Lừa đảo trực tuyến
17	Zero Trust	Kiến trúc không tin tưởng

BẢNG PHÂN CÔNG VIỆC

Họ và tên	Công việc phụ trách			
	Chương 1	Chương 2	Chương 3	Khác
Lã Thái Hòa	Hỗ trợ chỉnh sửa	Tìm kiếm viết nội dung	Tìm kiếm viết nội dung	Làm báo cáo Hỗ trợ làm slide Code Viết phần hướng phát triển
Nguyễn Quốc Thịnh	Hỗ trợ chỉnh sửa	Tìm kiếm viết nội dung	Hỗ trợ chỉnh sửa	Hỗ trợ code Viết phần kết luận Hỗ trợ viết báo cáo, làm slide
Lê Tuấn Đạt	Tìm kiếm viết nội dung	Hỗ trợ chỉnh sửa	Hỗ trợ chỉnh sửa	Làm slide Hỗ trợ code Hỗ trợ viết báo cáo

CHƯƠNG 1

TỔNG QUAN AN TOÀN BẢO MẬT THÔNG TIN, MẬT MÃ

1. Giới thiệu tổng quan về môn học

Trong bối cảnh công nghệ số phát triển mạnh mẽ, dữ liệu đã trở thành "trái tim" của mọi hoạt động, từ giao dịch thương mại điện tử, quản lý thông tin doanh nghiệp, đến bảo vệ an ninh quốc gia. Sự phụ thuộc ngày càng lớn vào dữ liệu này khiến an toàn và bảo mật thông tin trở thành một lĩnh vực quan trọng hơn bao giờ hết, đặc biệt khi các nguy cơ như tấn công mạng, khai thác lỗ hổng người dùng, hay rò rỉ thông tin ngày càng gia tăng. Học phần An toàn bảo mật thông tin (CT204) được thiết kế để cung cấp những kiến thức và kỹ năng thực tiễn nhằm bảo vệ thông tin trong thế giới thực, đáp ứng nhu cầu cấp thiết của xã hội số hóa. Môn học xoay quanh việc phân tích các nguy cơ mất an toàn thông tin từ những sơ hở của người sử dụng như thiếu ý thức bảo mật đến các kỹ thuật tấn công mạng như botnet, phishing, malware, … Đồng thời, học phần cũng giới thiệu các giải pháp bảo mật như phương pháp tạo mật mã với các kĩ thuật khóa bí mật và khóa công khai, từ những phương pháp tạo mật mã cổ điển như Affine, Caesar hay Diffie-Hellman, đến các phương pháp bảo mật cao hơn như AES, RSA, hay hàm băm, các mô hình tin cậy (Trust Models) như chứng thư số từ bên thứ ba đáng tin cậy. Ngoài ra, người học còn được tiếp cận các chính sách cũng như giải pháp an toàn thông tin, từ đó phát triển khả năng đánh giá rủi ro, đề xuất giải pháp phù hợp, và thiết kế hệ thống trao đổi thông tin an toàn, chẳng hạn như triển khai chứng thư số dựa trên hệ thống phân phối khóa công khai.

2. Tổng quan về an toàn thông tin hiện nay

2.1. Tầm quan trọng của an toàn bảo mật thông tin trong thời đại dữ liệu lớn

Với thời đại dữ liệu lớn (Big Data), khi khối lượng thông tin khổng lồ từ giao dịch trực tuyến, hồ sơ cá nhân, đến dữ liệu doanh nghiệp được tạo ra và lưu trữ mỗi ngày, mật mã học đã trở thành "chìa khóa vàng" để đảm bảo an toàn thông tin. Dữ liệu không chỉ là tài sản quý giá mà còn là mục tiêu hàng đầu của các cuộc tấn công mạng, từ việc đánh cắp thông tin nhạy cảm đến phá hoại hệ thống. Chính vì vậy, an toàn thông tin, với cốt lõi là các phương pháp mật mã đóng vai trò sống còn trong việc bảo vệ tính bí mật, toàn vẹn và xác thực của dữ liệu. Sự bùng nổ của dữ liệu lớn càng làm nổi bật tầm quan trọng của mật mã học, khi các vụ rò rỉ dữ liệu quy mô lớn hay tấn công ransomware cho thấy hậu quả nghiêm trọng khi thiếu lớp bảo vệ này. Do đó, việc hiểu và ứng dụng mật mã không chỉ là yêu cầu kỹ thuật mà còn là yếu tố chiến lược để bảo vệ dữ liệu trong một thế giới số hóa đầy rủi ro.

2.2. Các mối đe dọa an toàn thông tin hiện nay

Trong bối cảnh mật mã học trở thành xương sống của an toàn thông tin, các mối đe dọa an toàn thông tin ngày càng trở nên tinh vi và đa dạng. Các cuộc tấn công mạng như DDoS (tấn công từ chối dịch vụ), phishing (lừa đảo trực tuyến), hay deepfake được hỗ trợ bởi trí tuệ nhân tạo (AI) không chỉ khai thác lổ hổng công nghệ mà còn nhắm vào những sơ hở trong triển khai mật mã, làm suy yếu tính bảo mật của thông tin. Các lỗ hổng trong thiết bị IoT, mạng 5G và hệ thống đám mây cũng bị khai thác mạnh mẽ, tạo điều kiện cho mã độc tự động hóa và ransomware nhắm vào dữ liệu nhạy cảm phát triển. Đặc biệt, sự xuất hiện của máy tính lượng tử đang đặt ra mối đe dọa chưa từng có đối với an toàn thông tin. Với khả năng tính toán vượt trội, máy tính lượng tử có khả năng phá vỡ hệ thống mật mã hiện đại như RSA hay ECC bằng cách giải nhanh bài toán phân tích số nguyên tố hoặc logarithm rời rạc. Những mối đe dọa này không chỉ đòi hỏi các giải pháp bảo mật tiên tiến mà còn thúc đẩy chuyển đổi sang các hệ thống mật mã mới để đối phó với kỷ nguyên lượng tử.

2.3. Xu hướng và giải pháp an toàn thông tin hiện đại

Để đối phó với các mối đe dọa an toàn thông tin hiện nay đang chứng kiến nhiều xu hướng mới. Trí tuệ nhân tạo được ứng dụng để phát hiện và ngăn chặn tấn công tự động, trong khi kiến trúc Zero Trust yêu cầu xác minh mọi kết nối. Mật mã hậu lượng tử đang được phát triển để chuẩn bị cho mối nguy từ máy tính lượng tử, bên cạnh các quy định pháp lý như GDPR hay luật bảo vệ dữ liệu tại nhiều quốc gia như Luật An ninh mạng năm 2018 ở Việt Nam. Đồng thời, việc nâng cao ý thức người dùng qua các biện pháp như xác thực hai yếu tố hay quản lý mật khẩu an toàn cũng đóng vai trò quan trọng trong việc bảo vệ dữ liệu.

CHƯƠNG 2

TÌM HIỂU VỀ ỨNG DỤNG GIẤU TIN TRONG ẢNH

1. Một số vấn đề của giấu tin trong ảnh

1.1. Khái niệm về giấu tin trong ảnh

Giấu tin trong ảnh là kỹ thuật giấu tin mà trong đó thông tin sẽ được giấu trong dữ liệu ảnh. Các kỹ thuật giấu tin trong ảnh được thực hiện sao cho chất lượng ảnh ít bị thay đổi nhất để bằng mắt thường con người không thể phát hiện ra sự thay đổi đó. Cụ thể, các thuật toán giấu tin sẽ tìm cách khai thác và lợi dụng sự hạn chế về cảm nhận hình ảnh của con người để giấu thông tin. Tùy theo từng ứng dụng mà các kỹ thuật giấu tin có những tính chất và yêu cầu khác nhau. Nhưng tựu chung lại, các kỹ thuật giấu tin trong ảnh không chỉ phải đảm bảo tất cả các tính chất của kỹ thuật giấu tin yêu cầu mà còn phải đảm bảo một số tính chất riêng đối với môi trường ảnh [1, 2, 3]. Ngày nay, kỹ thuật giấu tin trong ảnh thường được sử dụng để truyền thông tin mật giữa người dùng mà người khác không thể biết được. Chính từ những lợi ích mà các kỹ thuật giấu tin trong ảnh mang lại, nên hiện nay lĩnh vực giấu tin trong ảnh đang được phát triển nhanh chóng và mạnh mẽ. Ví dụ như đối với các nước phát triển, chữ kí tay đã được số hóa và lưu trữ sử dụng như là hồ sơ cá nhân của các dịch vụ ngân hàng và tài chính, nó được dùng để xác thực trong các thẻ tín dụng của người tiêu dùng. Ngoài ra, phần mềm Microsoft Word cũng cho phép người dùng lưu trữ chữ kí trong ảnh nhị phân rồi gắn vào vị trí nào đó trong file văn bản để đảm bảo tính toàn vẹn của thông tin.

1.2. Một số định dạng ảnh và công cụ xử lí ảnh

1.2.1. Một số định dạng ảnh

Hiện nay, có nhiều loại định dạng ảnh khác nhau có thể được lựa chọn để giấu tin. Mỗi định dạng ảnh sẽ có tiêu chuẩn và tính chất khác nhau. Do đó, để tối ưu hóa quá trình giấu tin thì trước khi tiến hành giấu tin người giấu tin cần phải xem xét và đánh giá các định dạng, tiêu chuẩn ảnh. Sau đay là mô tả về một số định dạng ảnh đang được sử dụng phổ biến hiện nay:

- Định dạng ảnh BMP [4]: BMP được biết đến với tên tiếng Anh khác là Windows bitmap, là một định dạng ảnh phổ biến. Định dạng ảnh BMP được sử dụng để lưu trữ hình ảnh kỹ thuật số bitmap, độc lập với thiết bị hiển thị và có khả năng lưu trữ hình ảnh kỹ thuật số hai chiều cả đơn màu, đa màu, ở các độ sâu màu khác nhau tùy vào dữ liệu nén, các kênh alpha và các cấu hình màu. Một tập tin Bitmap bao gồm các cấu trúc theo thứ tự như biểu trên Bảng 1.1.

Bảng 1.1. Cấu trúc tập tin Bitmap

Tên cấu trúc	Kích thước	Mục đích
Tiêu đề tệp Bitmap	14 byte	Lưu trữ thông tin tổng quát về tệp hình ảnh bitmap
Tiêu đề DIB	Tùy theo các phiên bản	Lưu trữ thông tin chi tiết về ảnh bitmap và xác định định dạng điểm ảnh
Mặt nạ thêm bit	12 hoặc 16 byte	Xác định định dạng điểm ảnh.
Bảng màu	Tùy theo các phiên bản	Xác định màu sắc được sử dụng bởi dữ liệu hình ảnh bitmap
Gap1	Tùy theo các phiên bản	Cân chỉnh cấu trúc
Mảng điểm ảnh	Tùy theo các phiên bản	Xác định giá trị các điểm ảnh
Gap2	Tùy theo các phiên bản	Cân chỉnh cấu trúc
Màu ICC	Tùy theo các phiên bản	Xác định cấu hình màu để quản lý màu sắc

Từ bảng 1.1 có thể thấy được định dạng BMP có cấu trúc tương đối đơn giản. Ngoài ra, khi ảnh BMP không nén thì các ảnh này chỉ là một ma trận các điểm ảnh. Trong đó, mỗi một phần tử của ma trận biểu diễn một điểm ảnh, bao gồm các thành phần đỏ (kí hiệu R), xanh lục (kí hiệu G), xanh lam (kí hiệu B), alpha (kí hiệu A), các thành phần bổ sung (kí hiệu X). Ngày nay các kỹ thuật giấu tin trong ảnh sử dụng ảnh theo chuẩn BMP không được sử dụng phổ biến. Bởi vì các ảnh này có cấu trúc đơn giản, do đó giấu được ít thông tin cũng như thông tin sau khi giấu dễ bị phát hiện.

- **Định dạng ảnh PNG**: PNG (Portable Network Graphics) là một dạng ảnh sử dụng phương pháp nén dữ liệu không làm mất đi dữ liệu gốc. PNG hỗ trợ các ảnh dựa trên bảng màu (với bảng màu RGB 24 bit hoặc RGBA 32 bit), ảnh xám (có hoặc không có các kỹ thuật giấu tin kênh alpha) và ảnh RGB/RGBA không có bảng màu đầy đủ. Các giá trị trong phần tiêu đề của định dạng ảnh PNG được liệt kê trong bảng 1.2.

Bảng 1.2. Các giá trị trong tiêu đề tập tin PNG

Giá trị	**Mục đích**
89	Có các bit cao thiết lập để phát hiện các hệ thống truyền dẫn không hỗ trợ dữ liệu 8 bit, giảm nguy cơ mà một tập tin văn bản bị hiểu nhầm là một tập PNG, hoặc ngược lại.
50 4E 47	Là chữ cái PNG trong bảng mã ASCII, cho phép xác định định dạng PNG.
0D 0A	Là một kiểu kết thúc của DOS giúp phát hiện dòng kết thúc chuyển đổi dữ liệu.
1A	Một byte thông báo dừng hiển thị của tập tin.

Ngoài các thành phần tiêu đề tập tin PNG được mô tả trong bảng 1.2 thì chuẩn PNG còn có các đoạn mã lưu trữ dữ liệu (chunk). Đoạn mã lưu trữ dữ liệu này là một đoạn thông tin được sử dụng trong nhiều định dạng đa phương tiện. Mỗi một đoạn mã lưu trữ dữ liệu truyền tải thông tin nhất định về hình ảnh. Có hai loại đoạn mã lưu trữ dữ liệu: một là đoạn mã chính, hai là đoạn mã phụ trợ. Một bộ giải mã có khả năng đọc các đoạn mã lưu trữ dữ liệu quan trọng và hiển thị tệp PNG. Các đoạn mã lưu trữ dữ liệu phụ trợ là các thuộc tính hình ảnh khác có thể được lưu trữ trong các tệp PNG bao gồm các giá trị gamma, màu nền… Các đoạn mã lưu trữ dữ liệu quan trọng bao gồm IHDR, PLTE, IDAT, IEND. Giá trị của các IHDR, PLTE, IDAT, IEND được mô tả trong tài liệu [7].

- **Định dạng ảnh JPEG**: JPEG (Joint Photographic Experts Group) một nhóm các nhà nghiên cứu đã phát minh ra định dạng này để hiển thị các hình ảnh đầy đủ màu hơn mà kích thước file lại nhỏ hơn. Chuẩn JPEG có thể hiển thị các hình ảnh với các màu chính xác lên đến 16 triệu màu. Cấu trúc ảnh JPEG bao gồm nhiều phân đoạn (segment), ở mỗi đoạn là 1 cờ (marker), mỗi cờ bắt đầu bằng byte 0xFF và theo sau đó là 1 byte chỉ ra mã của loại cờ. Một số cờ chỉ gồm 2 byte; sau 2 byte cờ là 2 byte chỉ ra độ dài của đoạn không tính 2 byte của cờ. Với những đoạn chứa dữ liệu nén (entropy-coded data), 2 byte xác định độ dài của đoạn không tính độ dài của dữ liệu nén. Ảnh JPEG không yêu cầu các đoạn phải nằm theo đúng thứ tự nhưng đoạn đầu tiên của ảnh phải là đoạn SOI; đoạn cuối cùng là đoạn EOI. Một số thuộc tính của những cờ thường gặp trong ảnh JPEG được mô tả trong bảng 1.3 [8].

Bảng 1.3. Mô tả một số cờ thông dụng trong ảnh JPEG [9]

Tên rút gọn	Giá trị cờ	Mô tả tóm tắt
SOI	0xFF, 0xD8	Đánh dấu bắt đầu ảnh JPEG
SOFn	0xFF, 0xCn	Bắt đầu của khung, mô tả các thông số của ảnh: chiều cao, chiều rộng, số lượng thành phần màu, tỉ lệ số lượng thành phần màu.
DHT	0xFF, 0xC4	Xác định bảng Huffman. Trong ảnh JPEG có thể xuất hiện nhiều đoạn DHT.
DQT	0xFF, 0xDB	Xác định bảng lượng tử hóa. Trong ảnh JPEG có thể xuất hiện nhiều đoạn DQT.
SOS	0xFF, 0xDA	Đánh dấu bắt đầu quét ảnh từ trên xuống dưới.
APPn	0xFF, 0xEn	Dành riêng cho đoạn ứng dụng, đánh dấu bắt đầu của đoạn dữ liệu ứng dụng.
COM	0xFF, 0xEE	Cờ bắt đầu chứa lời bình (chú thích).
EOI	0xFF, 0xD9	Đánh dấu kết thúc ảnh.

Từ bảng 1.3 có thể thấy được ảnh JPEG có cấu trúc phức tạp bao gồm nhiều thành phần khác nhau. Dựa trên đặc điểm của các thành phần này, các phương pháp giấu tin trong ảnh sẽ khai thác để thực hiện giấu tin.

1.2.2. Một số công cụ xử lý ảnh phổ biến

Các công cụ xử lý ảnh hiện nay đóng vai trò quan trọng trong việc triển khai các phương pháp giấu tin trong ảnh và xử lý ảnh chuyên sâu. Sau đây là một số công cụ thường để xử lý ảnh phổ biến: Corel PaintShop Pro, GIMP, Adobe Photoshop Elements, Paint.NET, Photo Pos Pro, Zoner Photo Studio, PhotoScape, Xara Photo & Graphic Designer.

1.3. Phân loại kỹ thuật giấu tin trong ảnh

Các phương pháp, thuật toán giấu tin trong ảnh đang được chia thành 3 loại chính bao gồm [1, 2, 5, 6, 7]:

- **Giấu tin trên miền không gian ảnh:** Là kỹ thuật giấu tin mà các thông tin được giấu trực tiếp vào các điểm ảnh. Một số thuật toán và kỹ thuật thường được sử dụng để giấu tin trong miền không gian như [6, 7]: LSB (Least Significant Bit); Hoán vị giả ngẫu nhiên (Pseudo-random Permutation); Phương pháp giấu khối; Phương pháp Brundox; Phương pháp Darmstadter-Dellegle Quisquotter-McCa. Đặc điểm của các kỹ thuật giấu tin trong miền không gian là ảnh chứa tin sẽ không hoặc ít khi bị xử lý trước khi thực hiện giấu tin. Do đó, các kỹ thuật giấu tin trên miền không gian thường có hiệu quả thấp theo cả 2 tiêu chí: số lượng tin giấu và chất lượng hình ảnh sau khi giấu.

- **Giấu tin trong miền tần số ảnh:** Đây là kỹ thuật giấu tin mà trong đó các dữ liệu về điểm ảnh sẽ được biến đổi độc lập sang các dạng dữ liệu khác. Sau đó, thông tin sẽ được giấu vào các dữ liệu mới này. Như vậy, khác với kỹ thuật giấu tin trong miền không gian, các kỹ thuật giấu tin trong miền tần số thường tiến hành xử lý ảnh chứa tin rồi mới tiến hành giấu thông tin. Một số thuật toán và kỹ thuật thường được dụng để xử lý ảnh và giấu tin trong miền tần số ảnh như: Biến đổi cosine rời rạc (DCT - Discrete Cosine Transformations); Biến đổi Wavelet rời rạc (DWT - Discrete Wavelet Transform); Biến đổi Fourier rời rạc (DFT - Discrete Fourier Transform); Phương pháp Koch và Zhao; Phương pháp Bengam- Memon-Eo-Young; Phương pháp Hsu and Wu,... Các phương pháp giấu tin trong ảnh theo miền tần số mang lại hiệu quả tốt và giấu được nhiều thông tin và đảm bảo được tính bí mật. Hiện nay, hầu hết các ứng dụng đều sử dụng kỹ thuật giấu tin trên miền tần số.

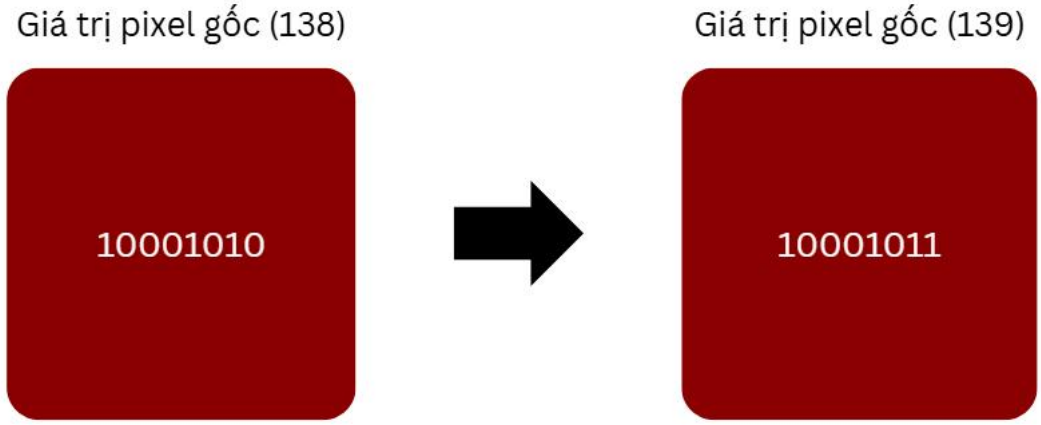
2. Kỹ thuật giấu tin trên K bit LSB

2.1. Nguyên lý của phương pháp LSB

Phương pháp LSB (Least Significant Bit)[8] là một trong những kỹ thuật cơ bản và hiệu quả nhất trong lĩnh vực giấu tin (steganography). Kỹ thuật này hoạt động dựa trên việc thay đổi có chọn lọc bit có trọng số thấp nhất trong biểu diễn nhị phân của các giá trị pixel trong ảnh kỹ thuật số. Mỗi pixel trong một ảnh màu RGB thường được biểu diễn bởi ba byte dữ liệu, tương ứng với cường độ của ba màu cơ bản: đỏ, xanh lá và xanh dương.

Bản chất của phương pháp này là khai thác một đặc tính tâm lý thị giác của con người: mắt người không thể phân biệt được những thay đổi rất nhỏ trong cường độ màu sắc. Cụ thể, khi thay đổi bit có trọng số thấp nhất (bit cuối cùng) trong biểu diễn nhị phân của một giá trị màu, sự thay đổi về cường độ màu sắc chỉ thay đổi 1 đơn vị trên thang 256 cấp độ (từ 0 đến 255). Sự thay đổi này quá nhỏ để mắt thường có thể nhận ra, tạo điều kiện lý tưởng cho việc giấu thông tin.

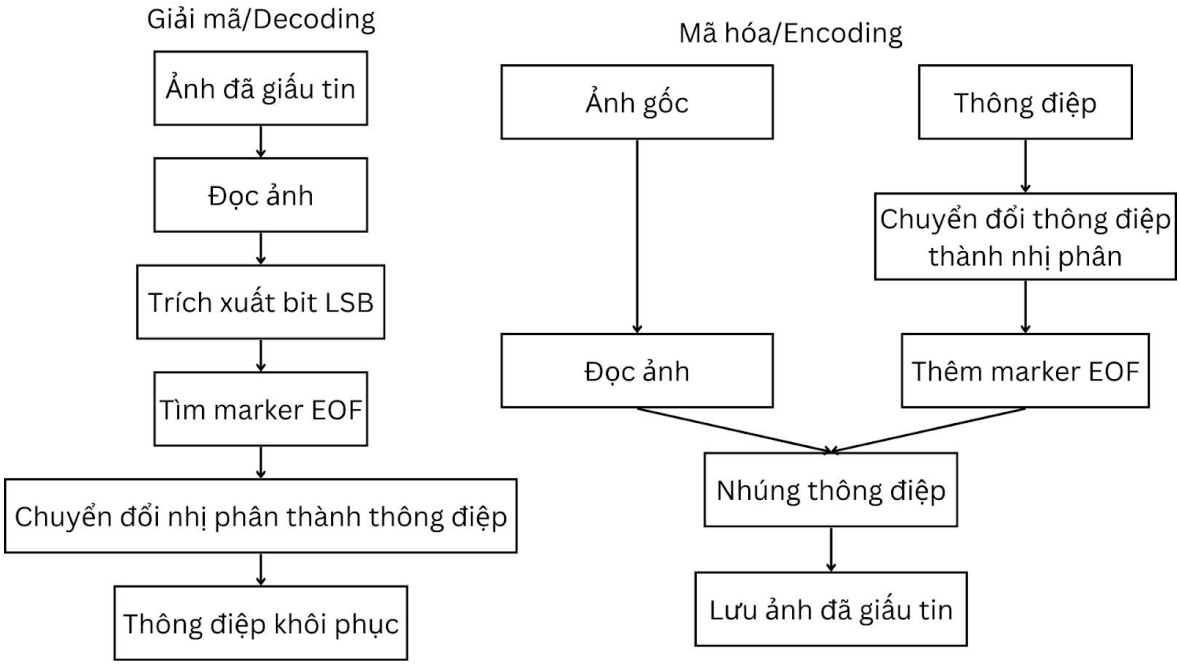
Ví dụ, nếu một pixel có giá trị màu đỏ là 138 (biểu diễn nhị phân là 10001010), việc thay đổi bit cuối cùng từ 0 thành 1 sẽ biến giá trị này thành 139 (10001011). Sự thay đổi này hầu như không gây ra sự khác biệt về mặt thị giác khi nhìn vào ảnh (Hình 1).



Hình 1 Sự thay đổi màu sắc khi thay đổi bít ít quan trọng

Luồng xử lý chính của hệ thống, chia thành hai phần chính (Hình 2):

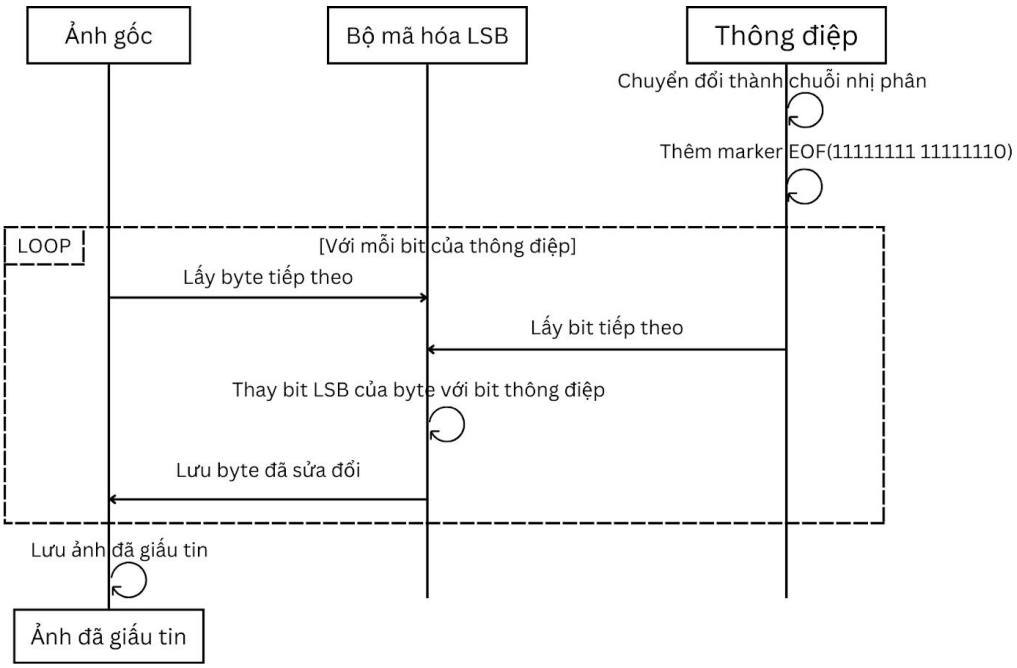
- **Quá trình mã hóa**: từ ảnh gốc và thông điệp đầu vào đến ảnh đã giấu tin
- **Quá trình giải mã**: từ ảnh đã giấu tin đến thông điệp được khôi phục



Hình 2 Kiến trúc của hệ thống LSB

2.2. Quá trình mã hóa thông tin

Quá trình mã hóa thông tin (Hình 2) bắt đầu bằng cách chọn một hình ảnh kỹ thuật số để ẩn thông tin. Sau đó, ảnh này sẽ được chuyển đổi sang định dạng pixel, trong đó mỗi pixel được biểu diễn bởi một giá trị màu. Tiếp theo, vị trí của bit ít quan trọng nhất (LSB) trong mỗi pixel sẽ được xác định. Bit này sẽ được thay thế bằng thông tin cần giấu.



Hình 3 Quy trình nhúng bit LSB

2.2.1. Chuyển đổi thông điệp thành dạng nhị phân

Bước đầu tiên trong quá trình mã hóa là chuyển đổi thông điệp cần giấu thành dạng nhị phân. Mỗi ký tự trong thông điệp văn bản được biểu diễn bằng một chuỗi 8 bit (1 byte) theo bảng mã như ASCII hoặc UTF-8. Ví dụ, ký tự 'A' trong bảng mã ASCII có giá trị thập phân là 65, tương ứng với biểu diễn nhị phân là 01000001.

Quá trình chuyển đổi này áp dụng cho toàn bộ thông điệp, kết quả là một chuỗi bit dài biểu diễn toàn bộ thông tin cần được giấu trong ảnh. Đối với các dạng dữ liệu phức tạp hơn như hình ảnh hoặc âm thanh, quá trình chuyển đổi có thể phức tạp hơn nhưng vẫn tuân theo nguyên tắc biểu diễn dữ liệu dưới dạng chuỗi các bit.

2.2.2. Thêm marker đánh dấu kết thúc thông điệp

Để có thể trích xuất chính xác thông điệp sau này, cần có cơ chế để xác định điểm kết thúc của thông điệp được giấu. Một phương pháp phổ biến là thêm một chuỗi bit đặc biệt, gọi là marker hoặc signature, vào cuối thông điệp. Marker này phải là một chuỗi bit đủ độc đáo để không xuất hiện một cách tự nhiên trong thông điệp.

Trong nhiều hệ thống, marker có thể là một chuỗi như '1111111111111110' (16 bit gồm 14 bit '1' liên tiếp và 2 bit '1' và '0' ở cuối). Độ dài và mẫu cụ thể của marker có thể được điều chỉnh tùy theo yêu cầu của hệ thống và đặc tính của dữ liệu cần giấu.

Việc thêm marker không chỉ giúp xác định điểm kết thúc của thông điệp mà còn là một biện pháp kiểm tra để đảm bảo tính toàn vẹn của thông điệp trong quá trình giải mã.

2.2.3. Thay thế bit LSB của từng byte trong ảnh

Sau khi có chuỗi bit biểu diễn thông điệp (bao gồm cả marker), bước tiếp theo là phân bố các bit này vào các pixel của ảnh. Quá trình này thực hiện bằng cách duyệt qua từng byte của ảnh (mỗi pixel RGB gồm 3 byte) và thay thế bit cuối cùng (LSB) của mỗi byte bằng một bit từ chuỗi thông điệp (Bảng 1).

Bảng 1. Ví dụ giấu chữ A (mã ASCII là 65 hay 01000001) vào trong 8 byte của file gốc

8 byte ban đầu	Byte cần giấy (A)	8 byte sau khi dấu
1001001**1**	**0**	1001001**0**
0110101**1**	**1**	0110101**1**
0101010**0**	**0**	0101010**0**
1010101**0**	**0**	1010101**0**

00011011	**0**	00011010
11010111	**0**	11010110
10101011	**0**	10101010
10011000	**1**	10011001

Quá trình thay thế tuân theo một quy tắc nhất định:

1. Nếu bit thông điệp là 0 và LSB của byte ảnh là 1, thì LSB sẽ được đổi thành 0
2. Nếu bit thông điệp là 1 và LSB của byte ảnh là 0, thì LSB sẽ được đổi thành 1
3. Nếu bit thông điệp trùng với LSB của byte ảnh, không cần thay đổi

Quá trình này tiếp tục cho đến khi toàn bộ thông điệp (bao gồm marker) đã được giấu vào ảnh. Trong trường hợp ảnh có kích thước lớn hơn nhiều so với lượng thông tin cần giấu, có thể chỉ một phần nhỏ của ảnh được sử dụng để giấu thông tin.

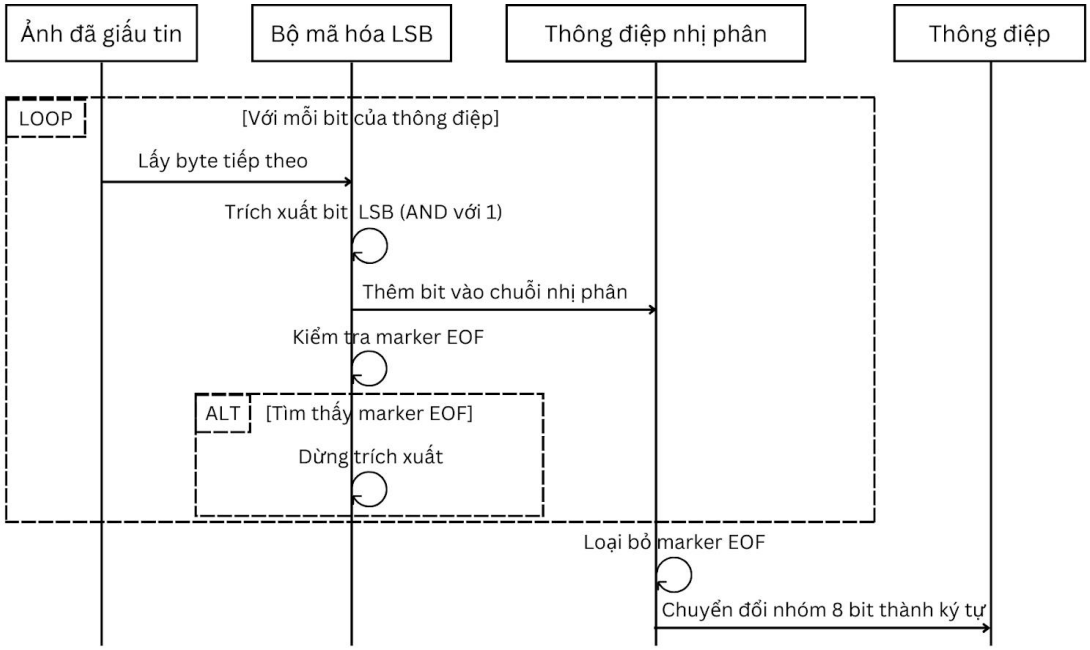
2.2.4. Lưu ảnh đã chứa thông điệp

Bước cuối cùng trong quá trình mã hóa là lưu ảnh đã được nhúng thông điệp vào một file mới. Điều quan trọng là phải sử dụng định dạng lưu trữ không nén hoặc nén không mất dữ liệu (lossless compression) như PNG hoặc BMP để đảm bảo các bit LSB đã được sửa đổi không bị thay đổi.

Nếu sử dụng định dạng nén có mất dữ liệu (lossy compression) như JPEG, quá trình nén có thể làm thay đổi các giá trị pixel, dẫn đến mất mát thông tin đã được giấu. Đây là một hạn chế quan trọng cần lưu ý khi triển khai hệ thống LSB steganography.

2.3. Quá trình giải mã thông tin

Quá trình giải mã thông tin của LSB (Hình 3) bắt đầu bằng chuyển hình ảnh đã được mã hóa bằng kỹ thuật LSB sang định dạng pixel, trong đó mỗi pixel được biểu diễn bởi một giá trị màu. Tiếp theo, vị trí của bit ít quan trọng nhất (LSB) trong mỗi pixel sẽ được xác định. Bit này chứa thông tin mà chúng ta đã giấu. Để lấy lại thông tin đã giấu, chúng ta sẽ đọc giá trị của LSB trong mỗi pixel và chuyển đổi nó thành dạng nhị phân. Sau đó, chúng ta sẽ kết hợp các giá trị nhị phân này lại với nhau theo nhóm 8 bit để tạo thành thông tin ban đầu đã giấu. Cuối cùng, thông tin đã giấu sẽ được chuyển đổi sang dạng văn bản hoặc hình ảnh để chúng ta có thể đọc hoặc xem được.



Hình 4 Quá trình giải mã thông tin

2.3.1. Đọc từng bit LSB của ảnh

Quá trình giải mã bắt đầu với việc đọc ảnh đã chứa thông điệp và trích xuất bit LSB từ mỗi byte của ảnh. Quá trình này phải tuân theo đúng thứ tự đã áp dụng trong quá trình mã hóa để đảm bảo các bit được trích xuất theo đúng trình tự của thông điệp gốc.

Việc trích xuất bit LSB tương đối đơn giản về mặt kỹ thuật: đối với mỗi byte của ảnh, bit LSB được xác định bằng cách thực hiện phép AND bit với giá trị 1 (operand AND 1). Kết quả sẽ là bit LSB của byte đó.

Quá trình này tiếp tục cho đến khi phát hiện ra marker kết thúc hoặc đã duyệt qua toàn bộ ảnh. Các bit LSB được trích xuất sẽ được ghép lại thành một chuỗi bit liên tục biểu diễn thông điệp được giấu.

2.3.2. Tìm marker kết thúc

Sau khi trích xuất chuỗi bit LSB, bước tiếp theo là xác định điểm kết thúc của thông điệp bằng cách tìm marker đã được thêm vào cuối thông điệp trong quá trình mã hóa. Quá trình này thường được thực hiện bằng cách duyệt qua chuỗi bit đã trích xuất và kiểm tra sự xuất hiện của mẫu bit đặc trưng của marker.

Khi phát hiện marker, hệ thống sẽ loại bỏ marker này và chỉ giữ lại phần bit biểu diễn thông điệp thực sự. Nếu không tìm thấy marker, có thể là do ảnh không chứa thông điệp hoặc thông điệp đã bị hỏng do ảnh bị chỉnh sửa sau khi giấu thông tin.

2.3.3. Chuyển đổi dãy bit thành văn bản

Bước cuối cùng trong quá trình giải mã là chuyển đổi chuỗi bit đã trích xuất (không bao gồm marker) thành thông điệp ban đầu. Quá trình này thực hiện bằng cách chia chuỗi bit thành các nhóm 8 bit (1 byte) và chuyển đổi mỗi nhóm thành ký tự tương ứng dựa trên bảng mã đã sử dụng trong quá trình mã hóa (thường là ASCII hoặc UTF-8).

Ví dụ, nếu một nhóm 8 bit là 01000001, giá trị thập phân tương ứng là 65, đại diện cho ký tự 'A' trong bảng mã ASCII. Quá trình này tiếp tục cho đến khi toàn bộ chuỗi bit đã được chuyển đổi thành văn bản.

Kết quả cuối cùng là thông điệp gốc đã được khôi phục hoàn toàn từ ảnh, hoàn thành quá trình giải mã.

3. Đặc điểm của hệ thống LSB Steganography

3.1. Độ an toàn

Hệ thống steganography sử dụng phương pháp LSB có một ưu điểm lớn về mặt an toàn là thông tin được giấu không thể nhìn thấy bằng mắt thường. Khi so sánh ảnh gốc và ảnh đã chứa thông điệp, hầu như không thể phát hiện sự khác biệt về mặt thị giác, ngay cả khi biết có thông tin được giấu trong đó.

Đây chính là bản chất của steganography - khả năng giấu thông tin mà không gây chú ý. Không giống như các phương pháp mã hóa (cryptography) truyền thống chỉ làm cho thông điệp không thể đọc được nhưng vẫn để lộ sự tồn tại của thông điệp, steganography còn giấu luôn cả sự tồn tại của thông điệp đó.

Tuy nhiên, cần lưu ý rằng phương pháp LSB cơ bản không cung cấp khả năng bảo mật thực sự cho nội dung thông điệp. Nếu một bên thứ ba biết về việc sử dụng LSB steganography, họ có thể dễ dàng trích xuất các bit LSB và khôi phục thông điệp. Vì vậy, để tăng cường bảo mật, thường kết hợp mã hóa thông điệp trước khi giấu bằng LSB.

Các kỹ thuật phân tích thống kê cũng có thể phát hiện sự hiện diện của thông tin giấu trong ảnh bằng cách phân tích phân bố của các bit LSB. Trong một ảnh tự nhiên, bit LSB thường có phân bố ngẫu nhiên, trong khi ảnh đã giấu thông tin có thể có phân bố bit LSB khác biệt, đặc biệt nếu thông điệp đã được mã hóa.

3.2. Dung lượng

Một trong những ưu điểm của LSB steganography là khả năng lưu trữ lượng thông tin tương đối lớn trong một ảnh. Công thức tính dung lượng tối đa có thể giấu trong một ảnh RGB là:

Dung lượng (bytes) = (Chiều cao × Chiều rộng × 3) ÷ 8

Trong đó:

- Chiều cao và chiều rộng là kích thước của ảnh tính theo pixel
- Hệ số 3 đại diện cho ba kênh màu (đỏ, xanh lá, xanh dương) của mỗi pixel

- Phép chia cho 8 chuyển đổi từ bit sang byte

Ví dụ, một ảnh có kích thước 1024×768 pixel có thể giấu được tối đa (1024 × 768 × 3) ÷ 8 = 294,912 bytes, tương đương khoảng 288 KB thông tin. Đây là một dung lượng đáng kể, đủ để lưu trữ các văn bản dài hoặc thậm chí các file nhỏ.

Tuy nhiên, trong thực tế, để đảm bảo tính không nhận biết được của phương pháp, thường chỉ nên sử dụng một phần dung lượng tối đa này. Việc sử dụng quá nhiều bit LSB có thể dẫn đến những thay đổi có thể phát hiện được về mặt thị giác hoặc thống kê.

Một số phương pháp cải tiến có thể tăng dung lượng bằng cách sử dụng nhiều hơn một bit (ví dụ 2 hoặc 3 bit có trọng số thấp nhất) của mỗi byte, nhưng điều này cũng làm tăng khả năng phát hiện của phương pháp.

3.3. Tính bền vững

Mặc dù có nhiều ưu điểm, LSB steganography có một nhược điểm đáng kể là tính bền vững thấp. Thông tin được giấu bằng phương pháp này rất dễ bị mất hoặc hỏng nếu ảnh bị chỉnh sửa sau quá trình giấu tin.

Ngay cả những thay đổi nhỏ như điều chỉnh độ sáng, độ tương phản, cắt xén hoặc thay đổi kích thước ảnh cũng có thể làm thay đổi giá trị của các bit LSB, dẫn đến mất thông tin. Đặc biệt, việc lưu ảnh dưới định dạng nén có mất dữ liệu như JPEG gần như chắc chắn sẽ phá hủy thông tin đã giấu.

Đây là một hạn chế quan trọng cần cân nhắc khi sử dụng LSB steganography trong các ứng dụng thực tế. Trong nhiều trường hợp, có thể cần kết hợp với các kỹ thuật sửa lỗi (error correction) để tăng khả năng khôi phục thông tin khi một phần thông tin bị mất.

CHƯƠNG 3
ỨNG DỤNG BÀI TOÁN CỤ THỂ

1. Môi trường cài đặt

Ngôn ngữ cài đặt: **Python**
Hệ điều hành: **window 11**
Môi trường soạn thảo: **VS Code**
Thư viện sử dụng:
- **Tkinter**: Thư viện chính để xây dựng giao diện người dùng đồ họa (GUI).
- **PIL (Pillow)**: Thư viện để xử lý ảnh, giúp mở và lưu ảnh, chuyển đổi ảnh thành mảng NumPy.
- **NumPy**: Dùng để xử lý ảnh dưới dạng mảng và thao tác trên dữ liệu ảnh.
- **Messagebox**: Để hiển thị các thông báo lỗi hoặc thành công cho người dùng.

2. Thiết kế hệ thống

Hệ thống giấu tin trong ảnh có cấu trúc như sau:

2.1. Giao diện người dùng (GUI):

- Giao diện cho phép người dùng chọn thực hiện giữa việc mã hóa và giải mã thông điệp từ ảnh đã được mã hóa.
- Cửa sổ mã hóa giúp người dùng nhập thông điệp cần mã hóa và chọn ảnh đầu vào và địa chỉ lưu ảnh đầu ra để mã hóa.
- Của sổ giải mã giúp người dùng chọn ãnh đã mã hóa và hiển thị ra thông điệp đã được mã hóa trong ảnh sau khi giải mã.

2.2. Chức năng chính của hệ thống

- **Mã hóa thông điệp:** chuyển thông điệp của người dùng nhập vào sang dạng nhị phân, sau đó tiến hành giấu vào ảnh dưới dạng các bit ít quan trọng (LSB).
- **Giải mã thông điệp:** trích xuất, giải mã thông điệp đã giấu từ trước trong ảnh bằng cách đọc các bit ít quan trọng.

2.3. Các thành phần chính:

- **Chương trình mã hóa:** đọc ảnh đầu vào, chuyển đổi thông điệp sang dạng nhị phân rồi tiến hành nhúng vào ảnh.
- **Chương trình giải mã:** đọc ảnh đầu vào đã nhúng, giấu thông điệp và trích xuất thông điệp từ các bit quan trọng.
- **Giao diện người dùng:** tạo ra cách của sổ để người dùng tương tác chọn ảnh đầu vào, nhập thông điệp, chọn địa chỉ lưu ảnh đầu ra, xem được ảnh đầu vào đã chọn, xem được ảnh sau khi được mã hóa, giấu tin,…

3. Cài đặt giải thuật

Giải thuật giấu tin trong ảnh sử dụng phương pháp Least Significant Bit (LSB) để giấu tin vào ảnh. Dưới đây là các bước cụ thể trong giải thuật và các bước thực hiện chúng:

B1: chuyển đổi thông điệp thành dạng nhị phân

- Đầu tiên, thông điệp nhập vào từ người dùng sẽ được mã hóa thành chuỗi nhị phân với mã hóa UTF-8.
- Mỗi kí tự trong thông điệp sẽ được chuyển thành một byte (8 bit), sau đó tất cả các byte sẽ được nối lại với nhau thành 1 chuỗi nhị phân duy nhất.

B2: thêm marker vào thông điệp

- Một marker (mã đánh giấu) được thêm vào cuối thông điệp để đánh giấu điểm kết thúc của thông điệp, giúp trong việc giải mã.

B3: mã hóa thông điệp vào ảnh

- Thông điệp được mã hóa vào ảnh bằng cách thay đổi bit ít quan trọng nhất (LSB) của các pixel ảnh. Ảnh sẽ được mở và chuyển thành mảng NumPy để thao tác với dữ liệu pixel.

B4: giải mã thông điệp từ ảnh

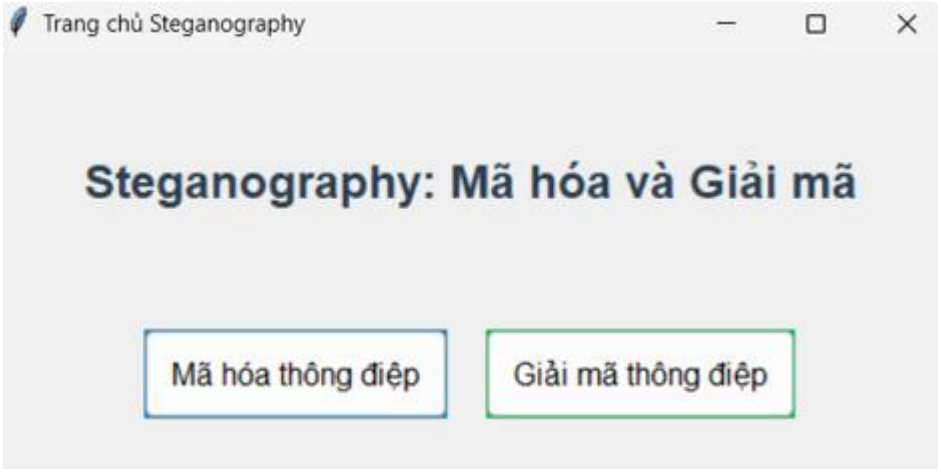
- Để giải mã thông điệp, hệ thống sẽ đọc lại các pixel của ảnh, trích xuất các bit ít quan trọng, và ghép lại cho chúng thành chuỗi nhị phân.
- Khi marker được tìm thấy, giải mã sẽ dừng lại tại vị trí đó.

4. Giao diện hệ thống

Giao diện hệ thống được xây dựng bằng thư viện Tkinter với các cửa sổ chính như sau:

4.1. Cửa sổ chính

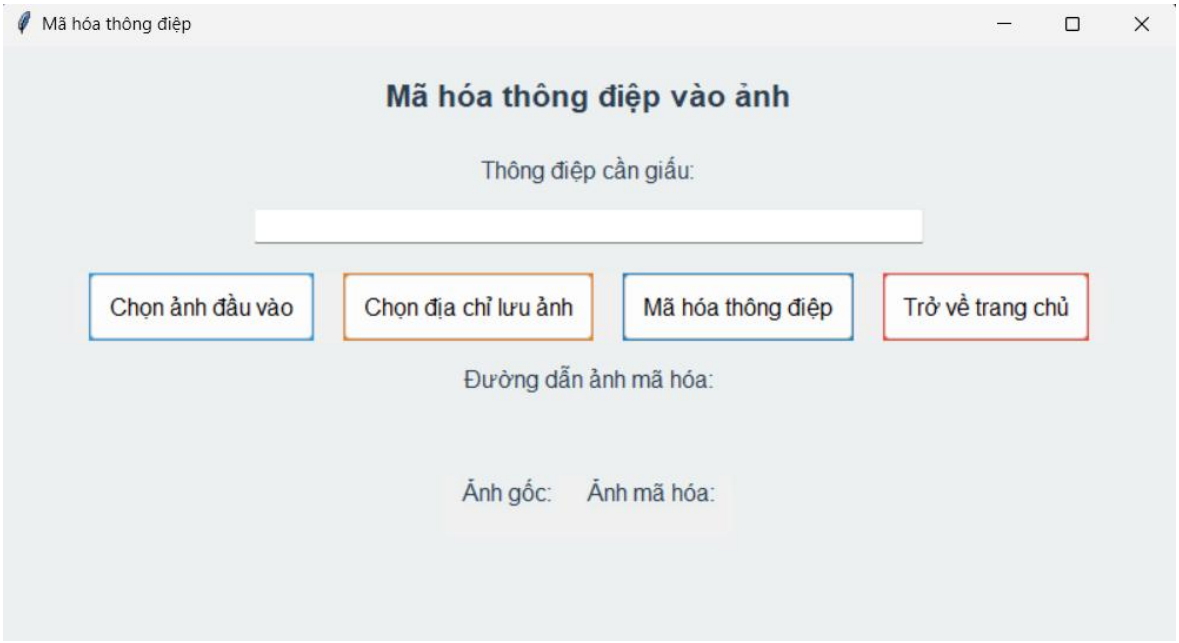
Cung cấp các nút để người dùng chọn thực hiện mã hóa hoặc giải mã thông điệp.



Hình 5 Giao diện trang chủ

4.2. Cửa sổ mã hóa thông điệp

Người dùng có thể chọn ảnh đầu vào, nhập thông điệp cần mã hóa, và chọn địa chỉ để lưu ảnh mã hóa, xem được ảnh đầu vào đã chọn cũng như xem được địa chỉ ảnh mã hóa sẽ được lưu, ảnh sau khi được mã hóa. Có nút để người dùng có thể trở về trang chính.



Hình 6 Giao diện trang mã hóa thông điệp

4.3. Cửa sổ giải mã thông điệp

Người dùng có thể chọn ảnh đầu vão để tiến hành giải mã thông điệp, người dùng có thể xem được ảnh đã được chọn, sau khi giải mã thành công thông điệp sẽ được hiện thị ra màn hình.

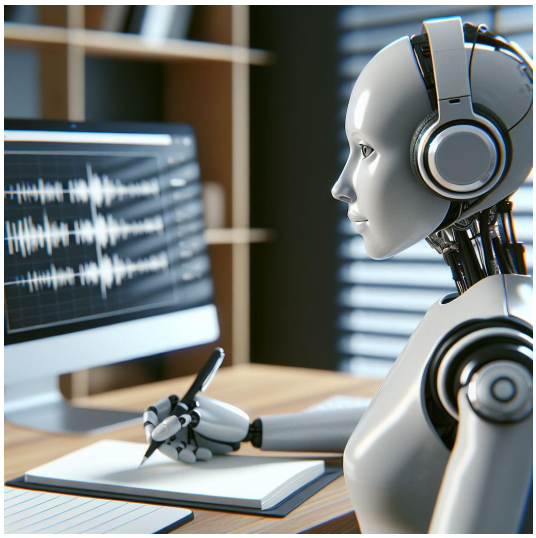
Hình 7 Giao diện trang giải mã thông điệp

5. Kiểm thừ và đánh giá

5.1. Kiểm tra quá trình mã hóa

Đầu vào là 1 ảnh bất kì.

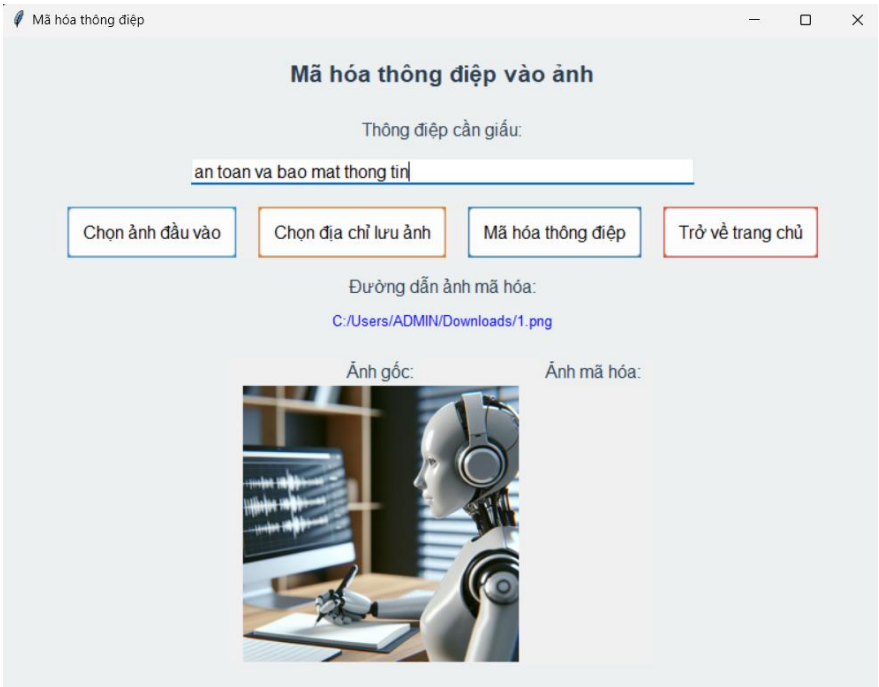
VD:



Hình 8 Ảnh đầu vào mã hóa

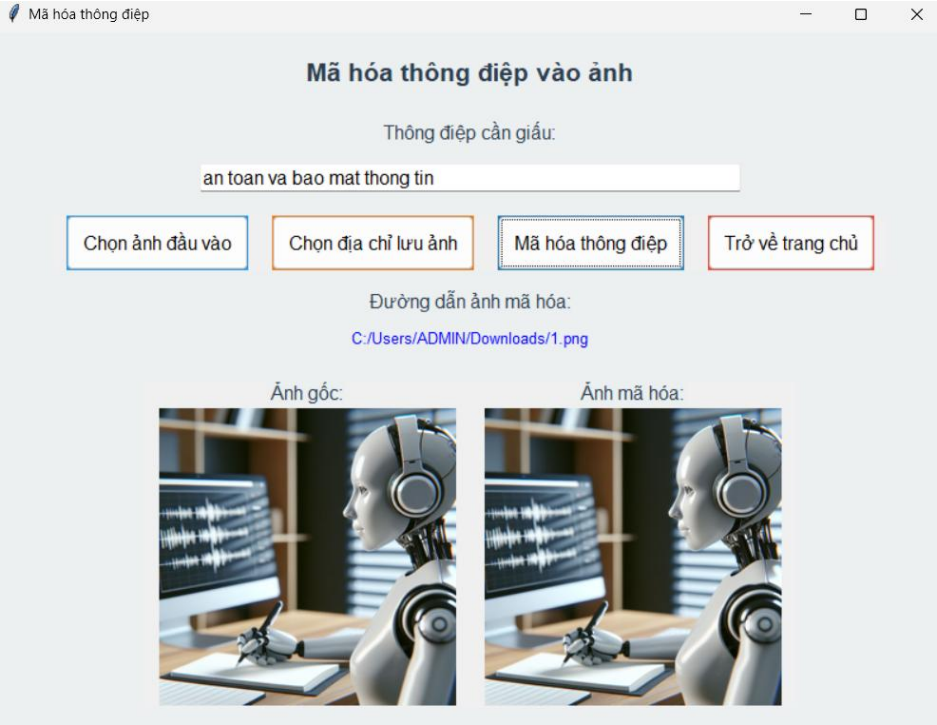
Thông điệp: An toan va bao mat thong tin

Địa chỉ lưu: C:/Users/ADMIN/Downloads/1.png



Hình 9 Giao diện mã hóa ảnh đã chọn

Tiến hành mã hóa

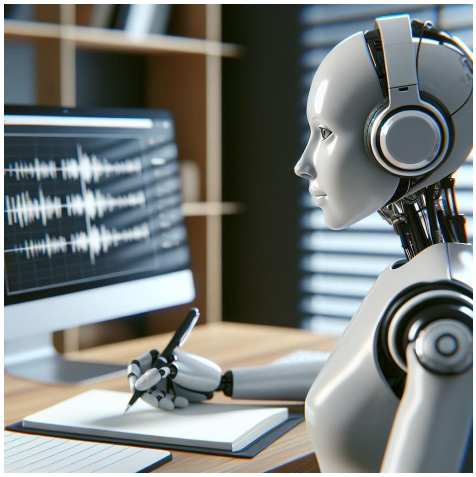


Hình 10 Giao diện khi mã hóa xong ảnh đầu vào

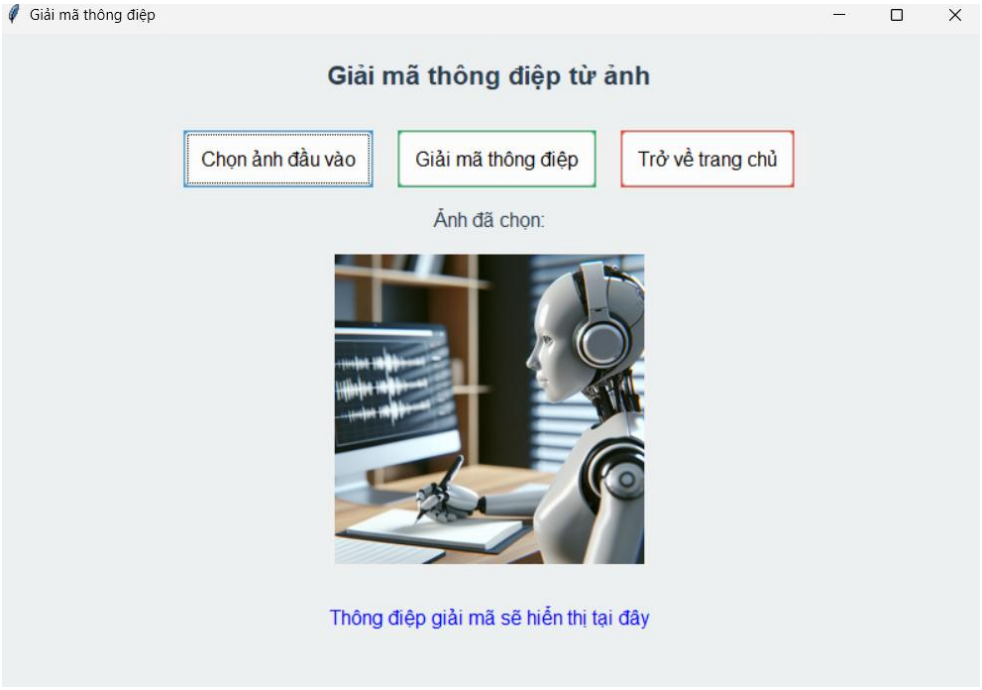
5.2. Kiểm tra quá trình giải mã

Ảnh đầu vào: là ảnh vừa được mã hóa ở quá mã hóa trên.

VD:



Hình 11 Ảnh đầu vào giải mã



Hình 12 Giao diện giải mã ảnh đã chọn

Tiến hành giải mã

Hình 13 Giao diện sau khi giải mã ảnh đã chọn

Kết luận: Cả 2 quá trình mã hóa và giải mã đều hoạt động bình thường và chính xác, hai ảnh mã hóa và ảnh ban đầu không sự khác biệt quá rõ rệt về chất lượng ảnh.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Ứng dụng đã đáp ứng các yêu cầu đặt ra ban đầu, giúp người dùng giấu và trích xuất thông tin một cách đơn giản và hiệu quả:

- Mã hóa và giải mã thông điệp:
  - Chương trình có khả năng chuyển đổi thông điệp thành dạng nhị phân và nhúng vào ảnh bằng cách thay đổi bit ít quan trọng nhất (LSB) của từng pixel.
  - Hỗ trợ nhiều định dạng ảnh như PNG, BMP, JPG với độ chính xác cao.
  - Khi giải mã, chương trình có thể tách thông điệp nhúng từ ảnh và hiển thị chính xác nội dung gốc.
- Giao diện đồ họa trực quan:
  - Xây dựng giao diện với Tkinter, giúp người dùng dễ dàng thao tác mã hóa và giải mã thông điệp.
  - Tích hợp các chức năng chọn ảnh, nhập thông điệp, xem ảnh gốc và ảnh mã hóa, nâng cao trải nghiệm người dùng.

2. Hướng phát triển:

- ✓ **Tăng cường tính bền vững**: Áp dụng các phương pháp giấu tin trong miền tần số (DCT, DWT) để chống nén và biến đổi hình học.
- ✓ **Kết hợp mã hóa mạnh hơn**: Sử dụng mã hóa AES kết hợp với giấu tin để nâng cao bảo mật.
- ✓ **Mở rộng ứng dụng**: Tích hợp giấu tin vào các phương tiện khác như âm thanh và video.
- ✓ **Cải thiện giao diện**: Hỗ trợ nhiều định dạng ảnh và tính năng kéo thả file.
- ✓ **Thủy vân số**: Thêm tính năng nhận dạng và bảo vệ bản quyền ảnh.
- ✓ **Tích hợp với hệ thống bảo mật**: Áp dụng cho các nền tảng truyền thông và giao dịch trực tuyến.

TÀI LIỆU THAM KHẢO

[1] Konakhovich Georgiy Filimonovich; Puzyrenko Alexander Yurievich. Computer Steganography. Theory and practice. "MKPress", 2019; 288p.

[2] Fabien A. Petitcolas, Stefan Katzenbeisser. Information Hiding Techniques for Steganography and Digital Watermarking. Boston, London: Artech House, 2000.

[3] Nguyễn Xuân Huy, Trần Quốc Dũng, Giáo trình Giấu tin và thủy vân ảnh, Trung tâm Thông tin tư liệu, TTKHTN – CN, 2003.

[4] Microsoft Windows Bitmap File Format Summary.

[5] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, Ton Kalker, Digital Watermarking and Steganography. Second Edition. Morgan Kaufmann Publishers is an imprint of Elsevier. 2008

[6] Shih, Frank Y. Digital Watermarking and Steganography: Fundamentals and Techniques (Second Edition). Taylor & Francis, CRC Press, 2017.

[s7] Chun-Shien Lu, Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property, Idea Group Publishing, 2005.

[8] Đỗ Xuân Chợ, Giáo trình các kỹ thuật giấu tin, Học viện công nghệ bưu chính viễn thông, 2023