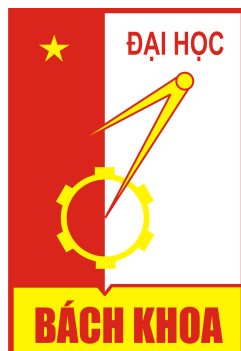


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



BÁO CÁO
AN TOÀN HỆ THỐNG THÔNG TIN

Chủ đề
TỔNG QUAN VỀ GIẤU TIN VÀ CÀI ĐẶT THUẬT
TOÁN LSB

Giảng viên hướng dẫn: Vũ Thành Nam

Ngô Thị Hiền

Sinh viên thực hiện: Phạm Thị Ngọc Anh

Đào Bảo Đại

Lớp: Hệ thống thông tin 02 – K65

HÀ NỘI – 2023

Mục lục

| | |
|---|-----------|
| Lời mở đầu | 2 |
| Chương 1 Tổng quan về giấu tin | 3 |
| 1.1 Khái niệm | 3 |
| 1.1.1 Mã hoá | 3 |
| 1.1.2 Giấu tin | 4 |
| 1.2 Phân loại các kỹ thuật giấu tin | 5 |
| 1.2.1 Thuỷ vân số (Watermarking) | 5 |
| 1.2.2 Giấu tin mật (Steganography) | 6 |
| 1.3 Mô hình kỹ thuật giấu tin và tách tin | 7 |
| 1.4 Ứng dụng | 8 |
| 1.5 Các tiêu chí đánh giá kỹ thuật giấu tin | 10 |
| Chương 2 Cài đặt thuật toán LSB với ảnh RGB | 12 |
| 2.1 Cơ sở lý thuyết | 12 |
| 2.1.1 Mô hình màu RGB | 12 |
| 2.1.2 Ảnh RGB | 13 |
| 2.2 Thuật toán LSB | 13 |
| 2.3 Cài đặt thuật toán | 15 |
| Tài liệu tham khảo | 21 |

Phân công công việc

| Họ và tên | MSSV | Công việc |
|-------------------|----------|--|
| Phạm Thị Ngọc Anh | 20206224 | Tìm tài liệu chung và tìm hiểu phần tổng quan về giấu tin. |
| Đào Bảo Đại | 20200126 | Tìm hiểu phần cài đặt thuật toán LSB với ảnh RGB và thiết kế chương trình mô phỏng thuật toán. |

Lời mở đầu

Giấu tin là một lĩnh vực quan trọng trong bảo mật thông tin, với ứng dụng rộng rãi trong nhiều lĩnh vực như an ninh mạng, pháp y, nghiên cứu dữ liệu và truyền thông. Việc khám phá và hiểu về giấu tin là một yếu tố quan trọng để đảm bảo an toàn và bảo mật thông tin trong thế giới kỹ thuật số ngày nay.

Bằng cách nghiên cứu đề tài này, chúng ta có thể hiểu được cách giấu thông tin trong các tập tin và hình ảnh thông qua việc thay đổi các bit không quan trọng. Ngoài ra, đề tài này cũng nhấn mạnh về thuật toán LSB (Least Significant Bit), một phương pháp giấu tin phổ biến được sử dụng rộng rãi. Việc cài đặt và hiểu thuật toán LSB sẽ giúp chúng ta áp dụng và ánh xạ các khái niệm và nguyên tắc của giấu tin vào thực tế.

Bên cạnh đó, việc nghiên cứu về giấu tin còn mang lại nhiều lợi ích và ứng dụng thực tiễn. Chẳng hạn, trong lĩnh vực an ninh mạng, giấu tin có thể được sử dụng để truyền tải thông tin một cách an toàn và bảo mật. Trong pháp y, giấu tin có thể hỗ trợ quá trình điều tra tội phạm. Ngoài ra, giấu tin còn có thể được áp dụng trong việc nghiên cứu và phân tích dữ liệu, đặt watermark và cả trong các trò chơi và mã hóa tin nhắn.

Vì những lý do trên, chúng em quyết định chọn đề tài “Tổng quan về giấu tin và cài đặt thuật toán LSB”. Chúng em mong rằng việc tìm hiểu và hiểu rõ hơn về giấu tin có thể giúp chúng ta nắm vững các nguyên tắc và phương pháp cơ bản để áp dụng vào các lĩnh vực khác nhau.

Trong quá trình thực hiện đề tài, chúng em đã nhận được rất nhiều sự chỉ bảo và góp ý chân thành của thầy Vũ Thành Nam và cô Ngô Thị Hiền. Chúng em xin cảm ơn thầy Nam, cô Hiền nói riêng và toàn thể các thầy cô trong viện Toán ứng dụng và Tin học nói chung đã giảng dạy cho em các kiến thức bổ ích là nền tảng để chúng em hoàn thiện đề tài.

Chương 1

Tổng quan về giấu tin

1.1 Khái niệm

1.1.1 Mã hoá

Mã hoá là quá trình chuyển đổi thông tin từ dạng "có thể hiểu được" thành dạng "không thể hiểu được". Mục đích chính của mã hoá là bảo vệ thông tin bằng cách biến đổi nó thành một dạng không thể đọc hoặc hiểu được cho những người không có quyền truy cập.

Giải mã là quá trình chuyển đổi thông tin ngược lại từ bản mã thành bản rõ.

Thuật toán mã hoá hoặc giải mã là một tập hợp các quy tắc, quy trình và phép toán được sử dụng để mã hoá hoặc giải mã.

Khoá mã hoá là một thông tin bí mật, một giá trị đặc biệt được sử dụng trong quá trình mã hoá và giải mã để bảo vệ và điều khiển quyền truy cập vào thông tin. Khóa được sử dụng để ánh xạ dữ liệu ban đầu sang dạng mã hoá và ngược lại từ mã hoá về dữ liệu ban đầu.

Hệ mã hoá là một hệ thống toàn diện bao gồm các thuật toán, quy tắc và quy trình để thực hiện việc mã hoá và giải mã thông tin. Có thể chia hệ mã hoá thành hai loại chính là hệ mã hoá khoá đối xứng và hệ mã hoá khoá bất đối xứng.

Hệ mã hoá khoá đối xứng hay **Hệ mã khoá bí mật** là những hệ mật được sử dụng chung 1 khóa trong quá trình mã hóa và mã hóa. Do đó khóa phải được giữ bí mật tuyệt đối. Sự mã hoá và giải mã của hệ thống mã hoá khoá đối xứng biểu thị bởi:

$$E_k : P \rightarrow C \text{ và } D_k : C \rightarrow P.$$

Hệ mã hoá khoá phi đối xứng hay **Hệ mã khoá công khai** là hệ mã hoá dùng 1 cặp khoá là Khoá công khai (public key) và khoá riêng (private key).

1.1.2 Giấu tin

a) Khái niệm

- Môi trường giấu tin (cover multimedia) là đối tượng được dùng để giấu tin như văn bản, ảnh, audio, video,...

Giấu tin trong ảnh:

Thông tin sẽ được giấu vào dữ liệu ảnh nhưng chất lượng ảnh thay đổi ít và khó biết được đằng sau ảnh đó mang những thông tin có ý nghĩa gì. Trong ảnh, thông tin được giấu một cách vô hình. Nó là một cách truyền thông tin mật cho nhau mà người khác không thể biết được.

Giấu tin trong audio

Giấu tin trong audio phụ thuộc vào hệ thống thính giác. Giấu thông tin trong audio đòi hỏi yêu cầu rất cao về tính đồng bộ và tính an toàn của thông tin. Các phương pháp giấu thông tin trong audio đều lợi dụng điểm yếu trong hệ thống thính giác của con người.

Giấu tin trong video:

Cũng giống như giấu tin trong ảnh hay audio, giấu tin trong video cũng được quan tâm và phát triển mạnh mẽ trong nhiều ứng dụng như điều khiển truy cập thông tin, xác thực thông tin và bảo vệ quyền tác giả. Ý tưởng cơ bản của phương pháp là phân phối thông tin giấu dần trải theo tần số của dữ liệu gốc.

- Dữ liệu sẽ được giấu (information) là một lượng thông tin mang ý nghĩa nào đó, tùy thuộc vào mục đích của người sử dụng.
- Giấu thông tin là nhúng mẫu tin mật vào một vật mang tin khác, sao cho mắt thường khó phát hiện ra mẫu tin mật đó, mặt khác khó nhận biết được vật mang tin đã giấu một tin mật.

b) So sánh giấu tin và mã hoá

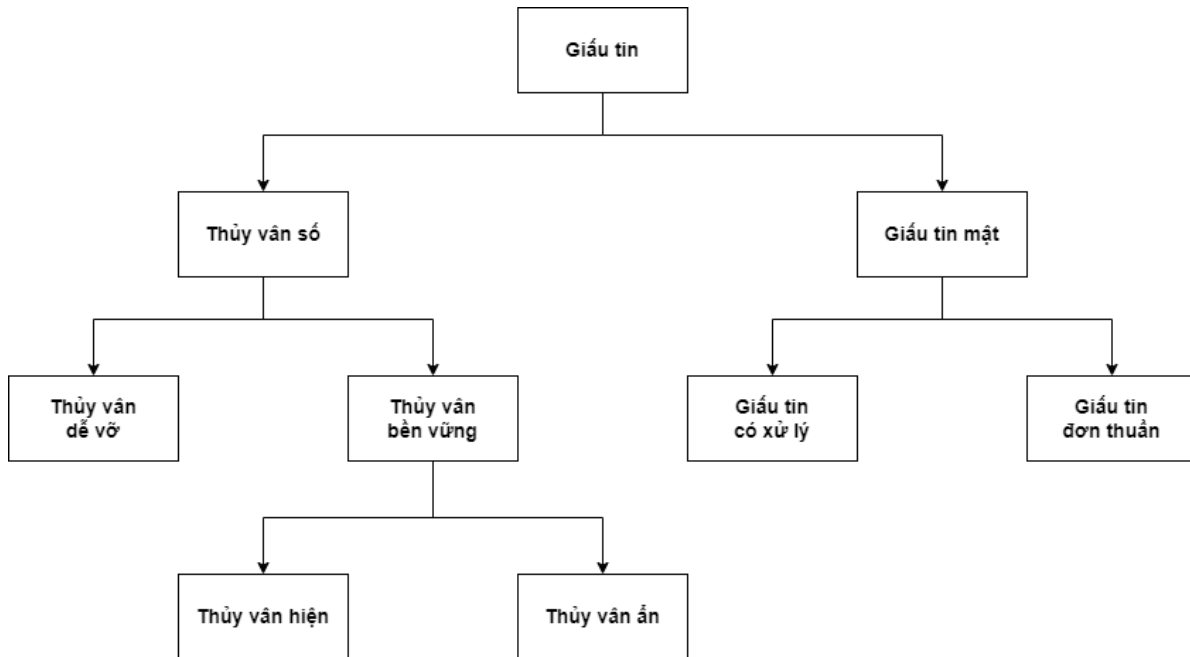
Giống nhau: Cùng mục đích là để người khác khó phát hiện ra tin cần giấu.

Khác nhau:

- "Mã hoá" là giấu đi "ý nghĩa" của thông tin.
- "Giấu tin" là giấu đi "sự hiện diện" của thông tin.

1.2 Phân loại các kỹ thuật giấu tin

Có thể chia kỹ thuật giấu tin ra làm 2 loại lớn đó là Watermarking và Steganography



Hình 1.1: Phân loại kỹ thuật giấu tin

1.2.1 Thủy văn số (Watermarking)

Giấu mẫu tin ngắn, nhưng đòi hỏi độ bền vững cao của thông tin cần giấu trước các biến đổi thông thường của tệp dữ liệu môi trường.

- **Thủy văn bền vững:** thường được ứng dụng trong bảo vệ bản quyền. Thủy văn được nhúng trong sản phẩm như một hình thức dán tem bản quyền. Trong trường hợp này, thủy văn phải tồn tại bền vững cùng với sản phẩm nhằm chống việc tẩy xóa, làm giả hay biến đổi phá hủy thủy văn.
 - **Thủy văn ẩn:** Thường nhúng thông tin vào dữ liệu gốc bằng cách thay đổi một số thuộc tính hoặc thống kê của dữ liệu, ví dụ như giá trị pixel trong hình ảnh hoặc các mẫu sóng trong âm thanh. Thông tin nhúng này có thể được mã hóa hoặc điều chỉnh theo các quy tắc nhất định để đảm bảo tính ẩn danh.
 - **Thủy văn hiện:** Là loại thủy văn hiện ngay trên sản phẩm và mọi người đều có thể nhìn thấy được.
- **Thủy văn dễ vỡ:** Là kỹ thuật nhúng thủy văn vào trong một đối tượng sao cho khi phân bố nó trong môi trường mở, nếu có bất kỳ biến đổi nào làm thay đổi sản phẩm gốc thì thủy văn đã được giấu trong đối tượng sẽ không còn nguyên vẹn như ban đầu.

1.2.2 Giấu tin mật (Steganography)

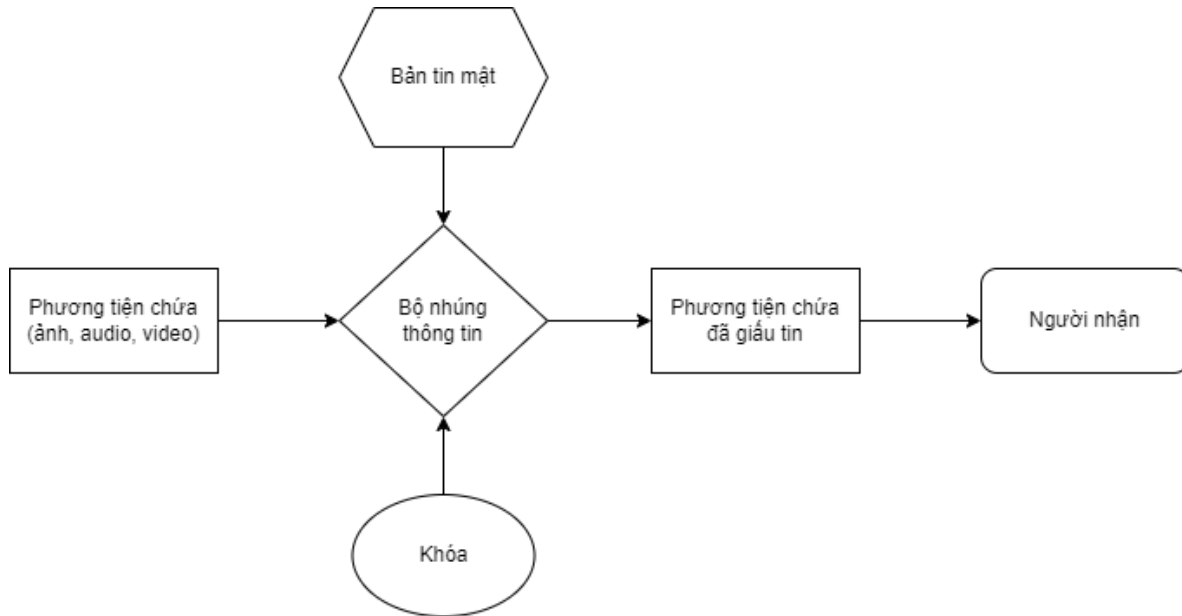
Che giấu bản tin (đòi hỏi độ mật cao và dung lượng càng lớn càng tốt) vào môi trường (đối tượng gốc).

| Watermarking | Steganography |
|---|---|
| <ul style="list-style-type: none">- Không cần giấu nhiều thông tin, chỉ cần lượng thông tin nhỏ đặc trưng cho bản quyền của người sở hữu.- Trong trường hợp thuỷ vân nhìn thấy thì thuỷ vân sẽ hiện ra.- Thuỷ vân phải bền vững với mọi tấn công có chủ đích hoặc không có chủ đích vào sản phẩm.- Thuỷ vân có đánh dấu vào chính đối tượng, nhằm khẳng định bản quyền sở hữu hay phát hiện xuyên tạc thông tin. | <ul style="list-style-type: none">- Tập trung vào việc giấu được càng nhiều tin càng tốt, ứng dụng trong truyền dữ liệu mật.- Cố gắng ít làm ảnh hưởng nhất tới chất lượng của đối tượng gốc để không bị chú ý đến dữ liệu đã được giấu trong đó.- Thay đổi đối tượng gốc cũng làm cho dữ liệu giấu bị sai lệch.- Bảo mật cho dữ liệu cần giấu. Khía cạnh này tập trung vào kỹ thuật giấu tin mật, tức là giấu tin sao cho ít người có thể phát hiện ra thông tin được giấu. |

1.3 Mô hình kỹ thuật giấu tin và tách tin

Mô hình giấu tin

Giấu thông tin vào phương tiện chứa và tách lấy thông tin là 2 quá trình trái ngược nhau. Ta có thể mô tả sơ đồ khối của hệ thống giấu tin như hình sau: Trong đó:



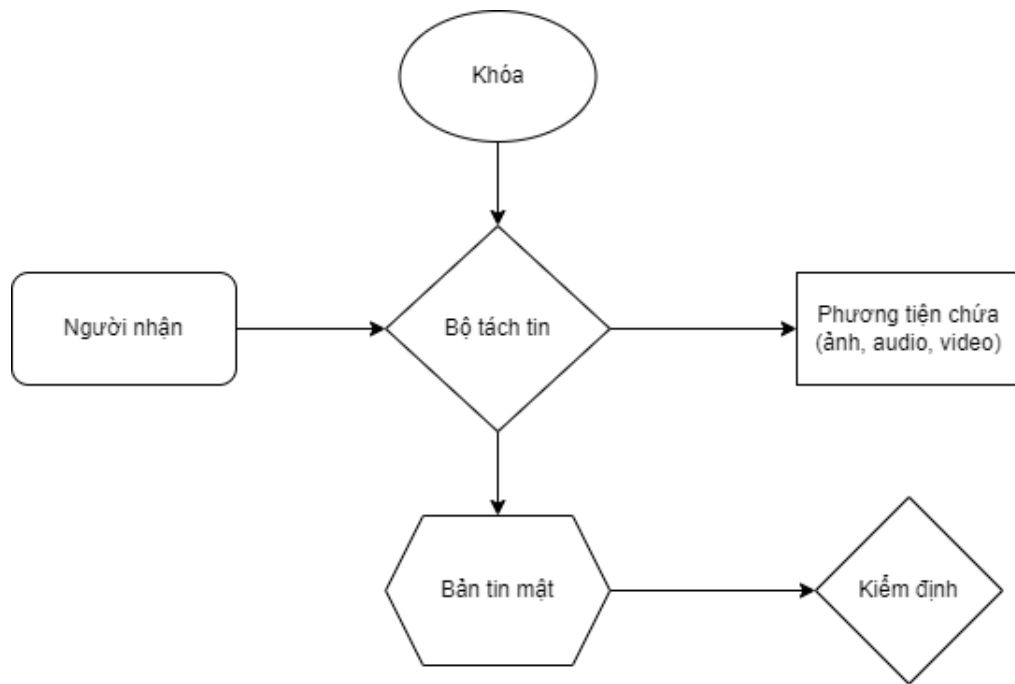
Hình 1.2: Mô hình giấu tin

- Bản tin mật tùy theo mục đích của người sử dụng, nó có thể là thông điệp (với các tin bí mật) hay các logo, hình ảnh bản quyền.
- Phương tiện chứa: các file ảnh, text, audio, ... là môi trường để nhúng tin.
- Bộ nhúng thông tin: là những chương trình thực hiện việc giấu tin.
- Đầu ra là các phương tiện chứa đã có tin giấu trong đó.

Mô hình tách tin

Tách thông tin từ các phương tiện chứa diễn ra theo quy trình ngược lại với đầu ra là các thông tin được giấu vào phương tiện chứa. Phương tiện chứa sau khi tách lấy thông tin có thể được sử dụng, quản lý theo những yêu cầu khác nhau.

Mô hình 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 một bộ giải mã tương ứng với bộ nhúng thông tin cùng với khóa của quá trình nhúng. Kết quả thu được bao gồm phương tiện chứa gốc và bản tin mật. Bước tiếp theo bản tin mật sẽ được xử lý kiểm định so sánh với thông tin ban đầu.



Hình 1.3: Mô hình tách tin

1.4 Ứng dụng

Bảo mật thông tin

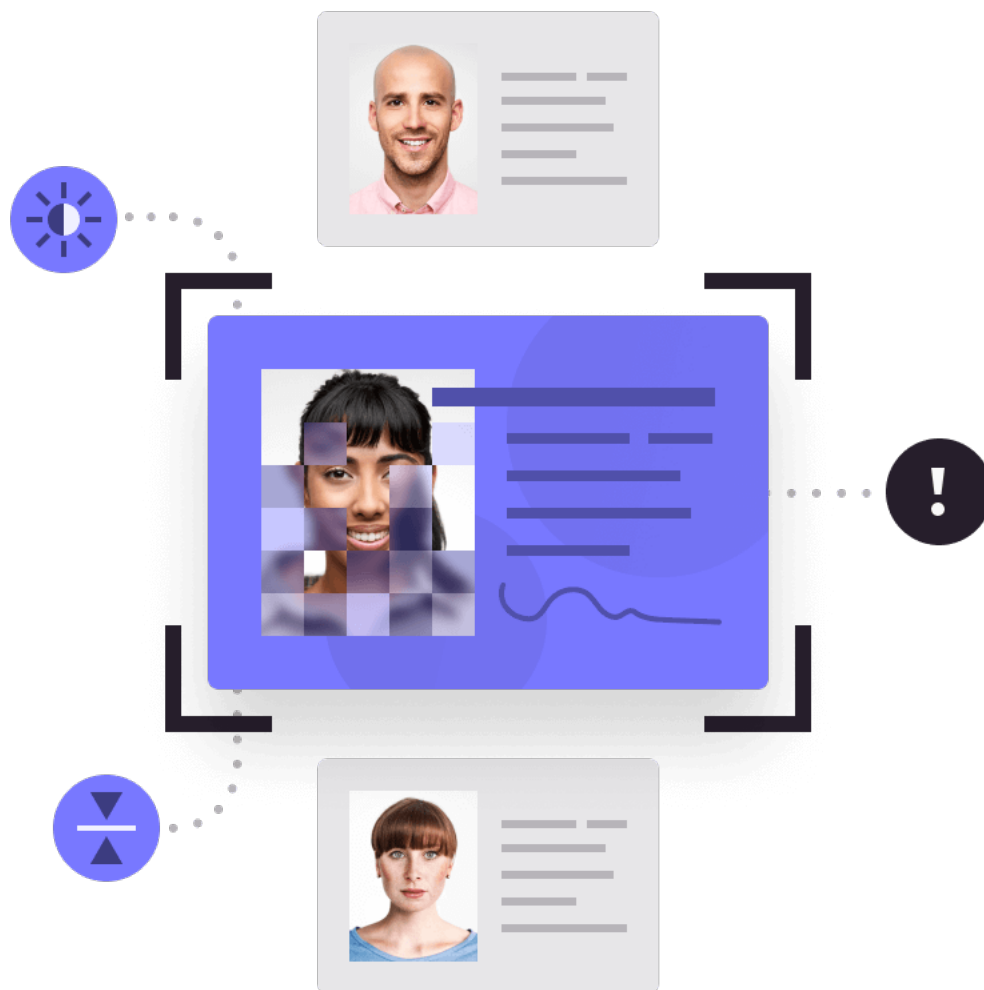
Kỹ thuật giấu tin được sử dụng để bảo vệ thông tin quan trọng bằng cách ẩn đi sự tồn tại của thông tin đó. Ví dụ, các tệp tin hoặc tin nhắn quan trọng có thể được giấu trong các tệp tin hoặc hình ảnh khác để tránh việc bị phát hiện hoặc đánh cắp.

Chứng thực và chống giả mạo

Kỹ thuật giấu tin có thể được sử dụng để nhúng các dấu vân tay số (digital watermark) vào các tài liệu hoặc hình ảnh, giúp xác định nguồn gốc và đảm bảo tính toàn vẹn của chúng. Điều này có thể được áp dụng trong việc chứng thực bản quyền, bảo vệ sự trung thực của tài liệu, hoặc ngăn chặn việc giả mạo.

Bảo vệ quyền tác giả

Đây là ứng dụng cơ bản nhất của kỹ thuật thủy văn số. Một thông tin nào đó mang ý nghĩa quyền sở hữu tác giả (watermark) sẽ được nhúng vào trong sản phẩm, chỉ có người chủ sở hữu hợp pháp sản phẩm đó có và được dùng làm minh chứng cho bản quyền sản phẩm. Yêu cầu kỹ thuật đối với ứng dụng này là thủy văn tồn tại bền vững với sản phẩm, muốn bỏ thủy văn mà không được phép của chủ sở hữu thì chỉ có cách phá hủy sản phẩm.



Hình 1.4: Chứng thực và chống giả mạo

Điều khiển truy cập

Các thiết bị phát hiện ra thủy vân thường được gắn sẵn vào trong hệ thống đọc ghi. Ví dụ như hệ thống quản lý sao chép DVD đã được ứng dụng ở Nhật. Ứng dụng này yêu cầu thủy vân phải được bảo đảm an toàn và sử dụng phương pháp phát hiện thủy vân đã giấu mà không cần thông tin gốc.

Giấu tin mật

Đây là ứng dụng giấu một lượng thông tin mật, quan trọng vào bên trong một đối tượng gốc nhằm che giấu, truyền thông bí mật điểm - điểm. Các thông tin giấu được càng nhiều càng tốt. Việc giải mã để nhận thông tin cũng không cần phương tiện gốc ban đầu.



Hình 1.5: Bảo vệ quyền tác giả

1.5 Các tiêu chí đánh giá kỹ thuật giấu tin

Tính vô hình

Như đã nêu, kỹ thuật giấu thông tin trong ảnh phụ thuộc rất nhiều vào hệ thống thị giác của con người. Tính vô hình (imperceptible) của mắt người thường giảm dần ở những vùng ảnh có màu xanh tím, thủy vân ẩn thường được chọn giấu trong vùng này.







Khả năng giấu thông tin

Khả năng giấu thông tin (Hiding capacity) hay lượng thông tin giấu được (dung lượng) trong một ảnh được tính bằng tỉ lệ giữa lượng thông tin giấu và kích thước của ảnh. Các thuật toán giấu tin đều cố gắng đạt được mục tiêu giấu được nhiều tin và gây nhiễu không đáng kể. Thực tế, người ta luôn phải cân nhắc giữa dung lượng tin cần giấu với các tiêu chí khác như chất lượng, tính bền vững của thông tin giấu.

Tính bền vững của thông tin được giấu

Tính bền vững thể hiện qua việc các thông tin giấu không bị thay đổi khi ảnh mang tin phải chịu tác động của các phép xử lý ảnh như nén, lọc, biến đổi, tỉ lệ,...

Change View "Conditions: Condition Types": Details

New Entries      

Condit. type Freight % Access seq. Vendor

Control data 1

| | | | |
|---------------|--|------------|---|
| Cond. class | <input type="text" value="A"/> Discount or surcharge | Plus/minus | <input type="text" value="A"/> Positive |
| Calculat.type | <input type="text" value="A"/> Percentage | | |
| Cond.category | <input type="text" value="B"/> Delivery costs | | |
| Rounding rule | <input type="text" value="Commercial"/> | | |
| StrucCond. | <input type="text"/> | | |

Hình 1.6: Điều khiển truy cập

Thuật toán và độ phức tạp tính toán

Cần nắm được một số kiến thức cơ bản về cấu trúc của ảnh để chọn ra thuật toán tìm miền ảnh thích hợp cho việc giấu tin. Độ phức tạp của thuật toán mã hoá và giải mã là yếu tố quan trọng để đánh giá các phương pháp giấu tin trong ảnh. Yêu cầu về độ phức tạp tính toán phụ thuộc vào từng ứng dụng. Những ứng dụng theo hướng Watermarking thường có thuật toán phức tạp hơn hướng Steganography.

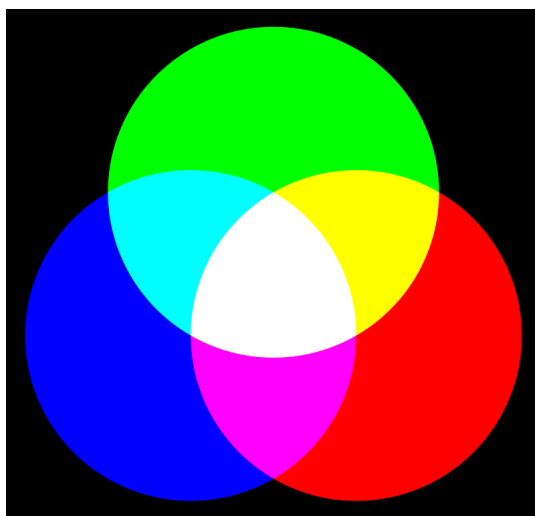
Chương 2

Cài đặt thuật toán LSB với ảnh RGB

2.1 Cơ sở lý thuyết

2.1.1 Mô hình màu RGB

Mô hình màu RGB sử dụng mô hình bổ sung trong đó ánh sáng đỏ, xanh lục và xanh lam được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu khác. Từ viết tắt RGB trong tiếng Anh có nghĩa là đỏ (red), xanh lục (green) và xanh lam (blue), là ba màu cơ bản trong các mô hình ánh sáng bổ sung.



Hình 2.1: Mô hình màu RGB

Cũng lưu ý rằng mô hình màu RGB tự bản thân nó không định nghĩa thế nào là "đỏ", "xanh lục" và "xanh lam" một cách chính xác, vì thế với cùng các giá trị như nhau của RGB có thể mô tả các màu tương đối khác nhau trên các thiết bị khác nhau có cùng một mô hình màu. Trong khi chúng cùng chia sẻ một mô hình màu chung, không gian màu thực sự của chúng là dao động một cách đáng kể.

2.1.2 Ảnh RGB

Một bức ảnh có chiều rộng là W , chiều cao là H nên số điểm ảnh (pixel) là $W \times H$, chúng ta sẽ có một ma trận các pixel có H dòng và W cột ($W \times H$). Mỗi pixel có một giá trị màu RGB (Red – Green – Blue), mỗi giá trị $R - G - B$ có giá trị từ $0 \rightarrow 255$ tương ứng với một Byte trong lưu trữ, suy ra có tất cả 256^3 tổ hợp màu khác nhau có thể được tạo ra.

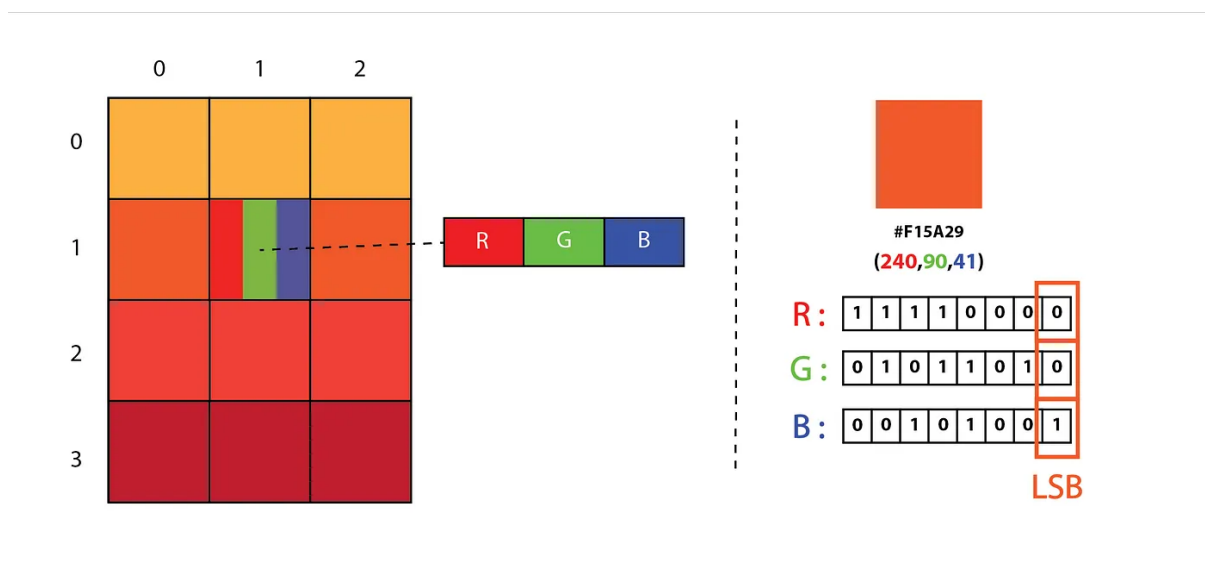
Ví dụ, một pixel với giá trị màu $(R, G, B) = (255, 0, 0)$ sẽ cho chúng ta màu đỏ đậm, trong khi $(0, 255, 0)$ sẽ tạo ra màu xanh lá cây đậm và $(0, 0, 255)$ sẽ tạo ra màu xanh dương đậm. Bằng cách kết hợp các giá trị R, G và B với các giá trị khác nhau, chúng ta có thể tạo ra mọi màu sắc từ màu trắng đến màu đen và tất cả các màu sắc khác nhau trên quy mô màu.

2.2 Thuật toán LSB

Thuật toán LSB (Least Significant Bit) là một kỹ thuật Steganography được sử dụng để giấu thông điệp vào trong dữ liệu không đáng kể của hình ảnh, âm thanh hoặc video. Tư tưởng chính của thuật toán LSB là sử dụng các bit ít quan trọng nhất của dữ liệu để lưu trữ thông điệp.

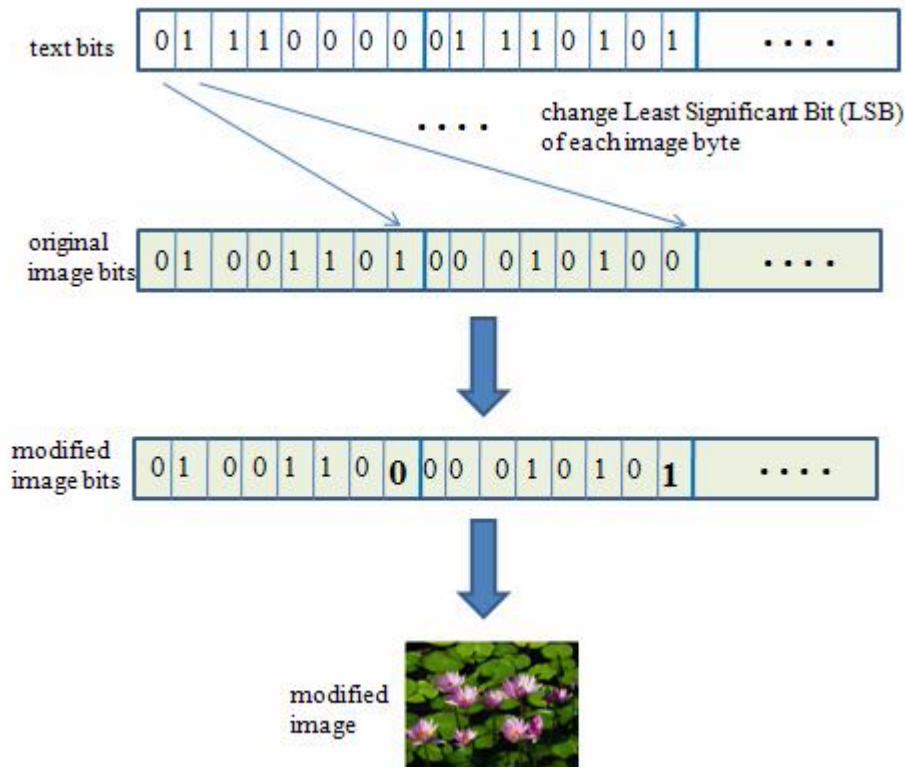
Ví dụ:

- Giá trị 147 chuyển sang nhị phân là 10010011 sẽ có LSB là 1.
- Giá trị 200 chuyển sang nhị phân là 11001000 sẽ có LSB là 0.



Hình 2.2: Least Significant Bit

Trong trường hợp của ảnh, thuật toán LSB sẽ sử dụng các bit cuối cùng của các giá trị màu RGB của từng pixel để chứa thông điệp. Vì các giá trị màu thay đổi rất ít, việc thay đổi bit cuối cùng sẽ không làm thay đổi đáng kể về màu sắc của ảnh, và thông điệp sẽ được giấu đi.



Hình 2.3: Mô tả thuật toán LSB

Quá trình Encode để giấu thông điệp vào bức ảnh:

- 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).

Quá trình Decode để đọc được thông điệp tại phía người nhận:

- 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.

2.3 Cài đặt thuật toán

Chương trình cài đặt thuật toán sẽ có 3 mục chính:

Hàm mã hóa (Encode): Dùng để giấu thông tin vào trong một bức ảnh cho trước.

```
def Encode(src, message, dest,password):

    img = Image.open(src, 'r')
    width, height = img.size
    array = np.array(list(img.getdata()))

    if img.mode == 'RGB':
        n = 3
    elif img.mode == 'RGBA':
        n = 4

    total_pixels = array.size//n

    message += password
    b_message = ''.join([format(ord(i), "08b") for i in message])
    req_pixels = len(b_message)

    if req_pixels > (total_pixels * 3):
        print("ERROR: Need larger file size")

    else:
        index=0
        for p in range(total_pixels):
            for q in range(0, 3):
                if index < req_pixels:
                    array[p][q] = int(bin(array[p][q])[2:9] + b_message[index], 2)
                    index += 1

        array=array.reshape(height, width, n)
        enc_img = Image.fromarray(array.astype('uint8'), img.mode)
        enc_img.save(dest)
        print("Image Encoded Successfully")
```

Hàm giải mã (Decode): Dùng để giải mã thông tin được giấu trong ảnh cho trước.

```
def Decode(src, password):

    img = Image.open(src, 'r')
    array = np.array(list(img.getdata()))
    if img.mode == 'RGB':
        n = 3
    elif img.mode == 'RGBA':
        n = 4
    total_pixels = array.size//n

    hidden_bits = ""
    for p in range(total_pixels):
        for q in range(0, 3):
            hidden_bits += (bin(array[p][q])[2:][-1])

    hidden_bits = [hidden_bits[i:i+8] for i in range(0, len(hidden_bits), 8)]

    message = ""
    hiddenmessage = ""
    for i in range(len(hidden_bits)):
        x = len(password)
        if message[-x:] == password:
            break
        else:
            message += chr(int(hidden_bits[i], 2))
            message = f'{message}'
            hiddenmessage = message
    #verifying the password
    if password in message:
        print("Hidden Message:", hiddenmessage[:-x])
    else:
        print("You entered the wrong password: Please Try Again")
```

Hàm main: Thực hiện các tác vụ của chương trình.

```
def Stego():
    print("--Welcome to $t3g0--")
    print("1: Encode")
    print("2: Decode")

    func = input()

    if func == '1':
        print("Enter Source Image Path")
        src = input()
        print("Enter Message to Hide")
        message = input()
        print("Enter Destination Image Path")
        dest = input()
        print("Enter password")
        password = input()
        print("Encoding...")
        Encode(src, message, dest,password)

    elif func == '2':
        print("Enter Source Image Path")
        src = input()
        print("Enter Password")
        password = input()
        print("Decoding...")
        Decode(src,password)

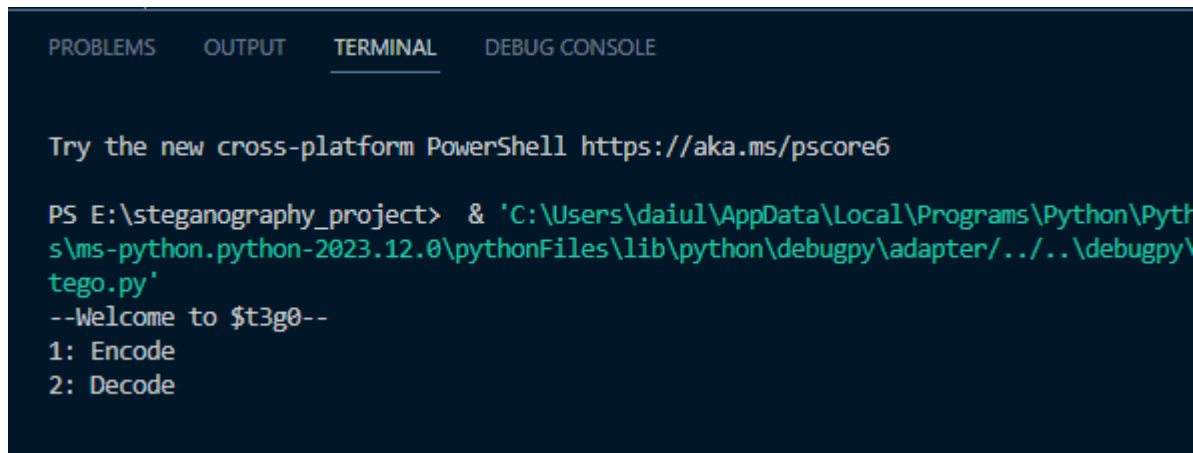
    else:
        print("ERROR: Invalid option chosen")

Stego()
```

Mô tả các bước thực hiện chương trình

Mã hóa

B1: Chạy chương trình và chọn 1 từ màn hình hiển thị để sử dụng chức năng mã hóa thông tin.

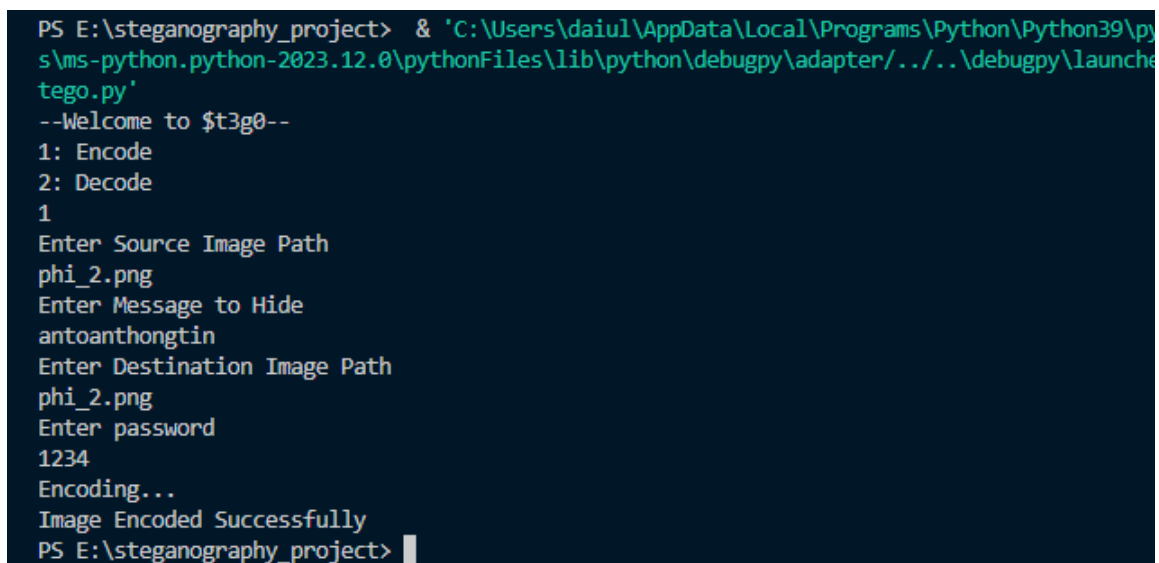


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\steganography_project> & 'C:\Users\daiul\AppData\Local\Programs\Python\Python39\python.exe' 'C:\Users\daiul\AppData\Local\Programs\Python\Python39\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' 'tego.py'
--Welcome to $t3g0--
1: Encode
2: Decode
```

B2: Chương trình sẽ yêu cầu ta thực hiện nhập vào các dữ liệu đầu vào để mã hóa bao gồm: ảnh gốc, thông tin cần giấu, nơi lưu trữ ảnh sau khi ảnh đã được nhúng tin mật và mật khẩu dùng để giải mã. Nếu chương trình in ra **"Image Encoded Successfully"** thì nghĩa là tin đã được giấu thành công.



```
PS E:\steganography_project> & 'C:\Users\daiul\AppData\Local\Programs\Python\Python39\python.exe' 'C:\Users\daiul\AppData\Local\Programs\Python\Python39\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' 'tego.py'
--Welcome to $t3g0--
1: Encode
2: Decode
1
Enter Source Image Path
phi_2.png
Enter Message to Hide
antoanthongtin
Enter Destination Image Path
phi_2.png
Enter password
1234
Encoding...
Image Encoded Successfully
PS E:\steganography_project> █
```

Trong trường hợp kích thước ảnh không đủ lớn so với kích thước của thông tin cần giấu thì chương trình sẽ báo lỗi: **"ERROR: Need larger file size"**.

Giải mã

B1: Để lấy được thông tin từ ảnh đã được giấu thì ta chạy chương trình và chọn 2 để có thể giải mã thông tin.

```

Enter password
1234
Encoding...
Image Encoded Successfully
PS E:\steganography_project> e;; cd 'e:\steganography_project'; & 'C:\Users\daiul
' 'c:\Users\daiul\.vscode\extensions\ms-python.python-2023.12.0\pythonFiles\lib\py
7' '--' 'E:\steganography_project\stego.py'
--Welcome to $t3g0--
1: Encode
2: Decode
2
Enter Source Image Path

```

B2: Chương trình sẽ yêu cầu ta thực hiện nhập vào các dữ liệu đầu vào để giải mã bao gồm: ảnh chứa thông tin cần giải mã và mật khẩu xác thực. Nếu chương trình in ra đúng thông tin như đã giấu ở phần mã hóa thì ta đã giải mã thành công.

```

PS E:\steganography_project> e;; cd 'e:\steganography_project'; & 'C:\Users\daiul
' 'c:\Users\daiul\.vscode\extensions\ms-python.python-2023.12.0\pythonFiles\lib\
7' '--' 'E:\steganography_project\stego.py'
--Welcome to $t3g0--
1: Encode
2: Decode
2
Enter Source Image Path
phi_2.png
Enter Password
1234
Decoding...
Hidden Message: antoanthongtin
PS E:\steganography_project>

```

Trong trường hợp nhập sai mật khẩu xác thực thì chương trình sẽ in ra "You entered the wrong password: Please Try Again".

```

--Welcome to $t3g0--
1: Encode
2: Decode
2
Enter Source Image Path
phi_2.png
Enter Password
asdfg
Decoding...
You entered the wrong password: Please Try Again
PS E:\steganography_project>

```

Kết luận

Chúng em đã tiến hành nghiên cứu về thuật toán LSB (Least Significant Bit) và áp dụng nó vào việc giấu tin trong ảnh RGB. Thuật toán này được xem là phương pháp đơn giản và phổ biến trong việc nhúng thông tin bổ sung vào ảnh. Tuy nhiên, nhận thấy rằng thuật toán LSB mang lại những ưu điểm và nhược điểm đặc thù.

Một số ưu điểm của thuật toán LSB là tính dễ hiện thực và hiệu quả, khả năng nhúng một lượng lớn thông tin vào ảnh mà không gây ảnh hưởng đáng kể đến chất lượng hình ảnh ban đầu. Việc trích xuất thông tin được ẩn đi từ ảnh cũng trở nên đơn giản. Tuy nhiên, cần lưu ý rằng thuật toán LSB cũng mang theo một số nhược điểm quan trọng.

Trước hết, thuật toán LSB dễ bị phát hiện và tấn công. Nhờ phân tích và so sánh các bit ít quan trọng, người khác có thể phát hiện thông tin ẩn một cách dễ dàng. Thứ hai, việc nhúng dữ liệu vào các bit ít quan trọng có thể dẫn đến mất mát dữ liệu nếu ảnh chứa nhiễu hoặc trải qua quá trình nén hay xử lý hình ảnh khác. Ngoài ra, ảnh hưởng đáng kể đến chất lượng hình ảnh cũng là một vấn đề cần quan tâm, đặc biệt khi nhúng dữ liệu lớn. Cuối cùng, thuật toán LSB cũng có hạn chế trong việc áp dụng cho ảnh có độ phân giải cao.

Do đó, để sử dụng thuật toán LSB một cách hiệu quả trong việc giấu tin vào ảnh RGB, cần xem xét kỹ lưỡng các yếu điểm và ưu điểm của thuật toán này. Đồng thời, cần đảm bảo lựa chọn thuật toán phù hợp dựa trên mục đích cụ thể và yêu cầu của ứng dụng. Sự cân nhắc kỹ lưỡng và kiểm soát chất lượng là rất quan trọng để đảm bảo hiệu quả và bảo mật trong quá trình giấu tin vào ảnh RGB.

Tài liệu tham khảo

- [1] GS. Phan Đình Diệu, *Lý thuyết mật mã và An toàn thông tin*, Nhà xuất bản Đại học quốc gia Hà Nội, 2002.
- [2] Wikipedia Contributor (2023), *Mô hình màu RGB*, https://vi.wikipedia.org/wiki/M%C3%B4_h%C3%ACnh_m%C3%A0u_RGB, 18/7/2023.
- [3] David Megias, Wojciech Mazurczyk, Minoru Kuribayashi. *Data Hiding and Its Applications: Digital Watermarking and Steganography*, Mdpi AG (January 21, 2022).