

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

MẬT MÃ HỌC VÀ CHỮ KÝ ĐIỆN TỬ

Đề tài:

Xây Dựng Hệ Thống Mật Mã Steganography

Sinh viên: Đỗ Thị Linh
Phạm Minh Trung
Nhóm: 5
Lớp: K20L
Giáo viên: Trịnh Minh Đức

Thái Nguyên, năm 2024

Mục lục

PHẦN 1: TỔNG QUAN.....	4
1.1. Sơ Lược	4
1.2. Đặt Vấn Đề.....	7
1.3. Mục Tiêu Cụ Thể.....	8
PHẦN 2: KỸ THUẬT GIẤU TIN – STEGANOGRAPHY	9
2.1. Giới Thiệu Chung	9
2.1.1. Lịch Sử Hình Thành.....	9
2.1.2. Khái Niệm	10
2.1.3. Steganography và Cryptography	12
2.1.4. Đặc Điểm.....	13
2.1.5. Ưu và Nhược Điểm.....	14
2.1.6. Ứng Dụng Thực Tiễn.....	15
2.2. Các Loại Kỹ Thuật Giấu Tin	15
2.2.1. Giấu Tin Trong Văn Bản (Text Steganography).....	15
2.2.1.1. Phương Pháp Dựa Trên Định Dạng	17
2.2.1.1.1. Sử dụng khoảng trắng.....	17
2.2.1.1.2. Giấu tin vào cuối mỗi dòng.....	20
2.2.1.1.3. Dịch chuyển vị trí dòng	21
2.2.1.1.4. Dịch chuyển vị trí từ	22
2.2.1.2. Phương pháp sinh ngẫu nhiên và thống kê.....	23
2.2.1.2.1. Sử dụng văn phạm phi ngữ cảnh	23
2.2.1.2.2. Dựa trên tính phản xạ đối xứng của ký tự	28
2.2.1.3. Phương pháp sử dụng tính chất ngôn ngữ	31
2.2.1.3.1. Sử dụng cú pháp	31
2.2.1.3.1. Sử dụng ngữ nghĩa.....	32
2.2.2. Giấu Tin trong Mạng (Network Steganography)	34
2.2.3. Giấu Tin trong Hình Ảnh (Image Steganography)	35
2.2.3.1. LSB (Least Significant Bit) Steganography	36
2.2.3.2. DCT (Discrete Cosine Transform) Steganography	42
2.2.3.3. Palette-Based Steganography	48

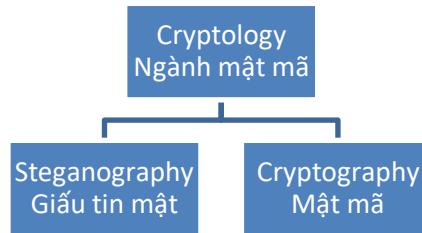
2.2.3.4. Spread Spectrum Steganography	52
2.2.3.5. DWT (Discrete Wavelet Transform) Steganography .	54
2.2.3.6. Các phương pháp khác	59
2.2.4. Giấu Tin Trong Video (Video Steganography)	59
2.2.4.1. Phương pháp phát hiện thay đổi khung cảnh	60
2.2.4.2. Phương pháp mặt phẳng bit.....	61
2.2.4.3. Phương pháp giấu trong miền video nén dựa trên sự khác biệt năng lượng.....	65
2.2.4.4. Phương pháp giấu trên miền nén của video chất lượng cao	68
2.2.5. Giấu Tin Trong Âm Thanh (Audio Steganography).....	74
2.2.5.1. Phương pháp LSB	75
2.2.5.2. Phương pháp mã hóa pha	76
2.2.5.3. Phương pháp trai phô	81
2.2.5.3.1. Trai phô nhảy tần (Frequency Hopping Spread Spectrum- FHSS).....	81
2.2.5.3.2. Trai phô dây trực tiếp (Direct Sequence Spread Spectrum - DSSS).....	85
2.2.5.4. Phương pháp Echo	89
PHẦN 3. THIẾT KẾ HỆ THỐNG.....	93
3.1. Tổng quan về hệ thống	93
3.2. Công nghệ và công cụ thiết kế	93
3.3. Mô Tả Và Thiết Kế Hệ Thống.....	94
3.4. Xây Dựng Và Triển Khai Hệ Thống	96
3.5. Kết Quả và Đánh Giá.....	100
3.6. Hướng Phát Triển Trong Tương Lai	101

PHẦN 1: TỔNG QUAN

1.1. Sơ Lược

Sự phát triển của công nghệ thông tin đã đem lại những thay đổi sâu sắc trong xã hội và trong cuộc sống của chúng ta. Những thuận lợi mà thông tin kỹ thuật số mang lại cũng sinh ra những thách thức và cơ hội cho quá trình đổi mới. Sự ra đời những phần mềm có tính năng rất mạnh, các thiết bị mới như máy ảnh kỹ thuật số, máy quét ảnh chất lượng cao, máy in, máy ghi âm kỹ thuật số, v.v..., đã được sáng tạo trên cơ sở thỏa mãn thế giới tiêu dùng rộng lớn, để xử lý và thưởng thức các dữ liệu đa phương tiện (multimedia data). Mạng Internet toàn cầu đã hình thành một xã hội ảo nơi diễn ra quá trình trao đổi thông tin trong mọi lĩnh vực chính trị, quân sự, quốc phòng, kinh tế thương mại,... Chính trong môi trường mở và tiện nghi như thế xuất hiện những vấn nạn, tiêu cực đang rất cần đến các giải pháp hữu hiệu cho vấn đề an toàn thông tin như nạn ăn cắp bản quyền, nạn xâm nhập thông tin, truy cập thông tin trái phép, ... Tìm giải pháp cho những vấn đề nêu trên không chỉ tạo điều kiện đi sâu vào lĩnh vực công nghệ phức tạp đang phát triển rất nhanh này mà còn dẫn đến những cơ hội phát triển kinh tế.

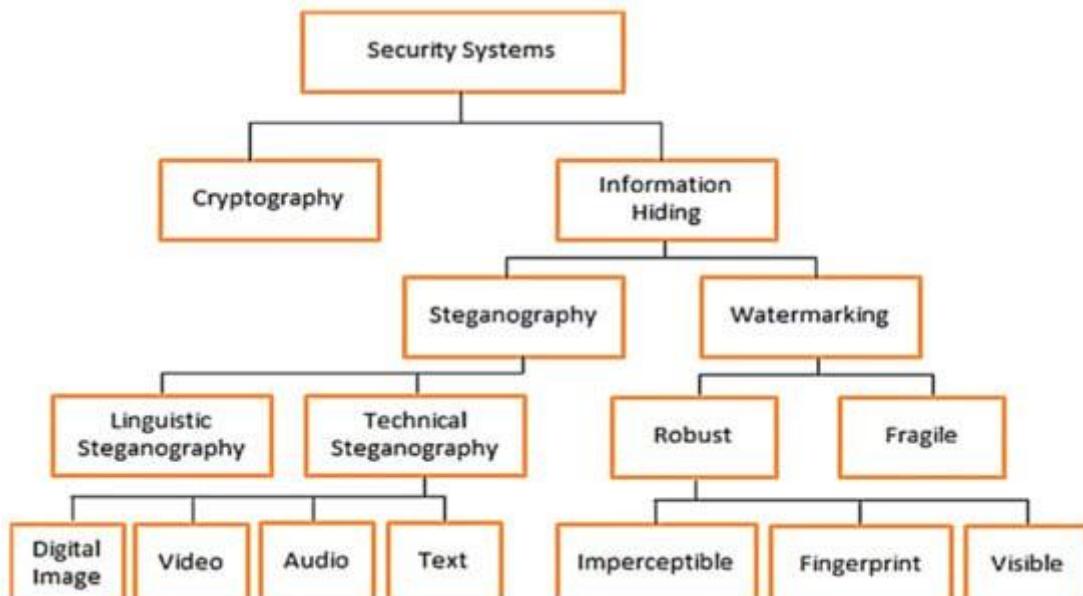
Quá trình phát triển lâu dài, có nhiều phương pháp bảo vệ thông tin đã được đưa ra, trong đó giải pháp dùng mật mã học là giải pháp được ứng dụng rộng rãi nhất. Các hệ mã đã được phát triển nhanh chóng và được ứng dụng rất phổ biến cho đến tận ngày nay. Thông tin ban đầu được mã hóa thành các ký hiệu vô nghĩa, sau đó sẽ được lấy lại thông qua việc giải mã nhờ vào khóa của hệ mã. Đã có rất nhiều những hệ mã phức tạp được sử dụng như DES, RSA,... 7 các phương pháp này trong thực tế tỏ ra rất hiệu quả và được ứng dụng phổ biến. Tuy nhiên trong báo cáo này không đi sâu vào nghiên cứu về các hệ mật mã mà chỉ tiếp cận với một phương pháp đã và đang được nghiên cứu, phát triển ở nhiều nước trên thế giới, đó là phương pháp che giấu thông tin



Hình 1.1: Phân cấp các lĩnh vực nghiên cứu của mật mã học

Ngày nay nó được ứng dụng rất nhiều trong công nghệ, đặc biệt trong ngành an ninh mạng bởi tính bảo mật của nó. Trong một số tình huống cần tính bảo mật cao thì việc mã hóa thôi là chưa đủ, một số mã hóa phổ biến hiện nay đang ngày càng thể hiện sự thiếu đi tính bí mật khi các tội phạm công nghệ cao ra đời cùng với sự thay đổi chóng mặt của các công nghệ máy tính hiện đại. Đó là lý do Steganography ra đời, nơi mà các thông tin thực sự cần ẩn giấu khỏi các “hacker”

Cây lược đồ sau chỉ rõ vấn đề, chia thành cách nhánh cây chi tiết. Có được cái nhìn tổng thể và trực quan nhất về hệ thống.



Hình 1.2: Cây phân loại của hệ thống bảo mật dữ liệu chung

- *Hệ Thống Bảo Mật* (Security Systems):

- o *Information Hiding*: Là việc giấu thông tin, bao gồm Steganography và Watermarking, là hai chiến lược chính để che giấu thông tin trong các đối tượng khác.
 - *Steganography*: Được phân loại thành hai nhánh chính:
 - *Technical Steganography*: Sử dụng kỹ thuật và công nghệ để ẩn thông tin. Các phương pháp này thường liên quan đến việc sử dụng công nghệ số và thủ thuật kỹ thuật để giấu thông tin mà không bị phát hiện.
 - o *Digital Image*: Sử dụng hình ảnh kỹ thuật số để ẩn thông tin.
 - o *Video*: Ẩn thông tin trong video.
 - o *Audio*: Ẩn thông tin trong tệp âm thanh
 - o *Text*: Ẩn thông tin trong văn bản thông qua việc thay đổi định dạng hoặc sử dụng
 - *Linguistic Steganography*: Ẩn thông tin trong cấu trúc ngôn ngữ hoặc văn bản.
- *Watermarking*:
 - *Robust Watermarking*: Đây là loại watermark được thiết kế để chống lại sự thay đổi và xử lý, nhằm bảo vệ quyền sở hữu và xác minh nguồn gốc của đối tượng.
 - o *Imperceptible*: Watermark này không thể nhận biết được bằng mắt thường và không làm ảnh hưởng đến chất lượng của đối tượng chứa.
 - o *Fingerprint*: Dấu vân tay duy nhất cho mỗi bản sao của đối tượng, giúp theo dõi và xác định nguồn gốc của từng bản sao.
 - o *Visible*: Watermark rõ ràng có thể nhìn thấy, thường được sử dụng để gắn nhãn và bảo vệ hình ảnh.
 - *Fragile Watermarking*: Được thiết kế để dễ bị hỏng khi có sự thay đổi, như vậy có thể sử dụng để phát hiện sự can thiệp hoặc chỉnh sửa.

1.2. Đặt Vấn Đề

Trong bối cảnh kỹ thuật số ngày nay, dữ liệu và thông tin cá nhân trên internet ngày càng trở nên quan trọng và đồng thời cũng dễ bị tấn công hơn bao giờ hết. Sự gia tăng của các vụ việc an ninh mạng, từ việc đánh cắp thông tin cá nhân, xâm nhập hệ thống thông tin của doanh nghiệp cho đến các cuộc tấn công nhắm vào cơ sở hạ tầng quan trọng, đã làm dấy lên những lo ngại sâu sắc về khả năng bảo vệ thông tin trong thế giới kỹ thuật số. Các hacker không ngừng nâng cao kỹ năng và sử dụng công nghệ tiên tiến để tìm ra và khai thác các lỗ hổng bảo mật, khiến cho các tổ chức, doanh nghiệp và người dùng cá nhân gặp phải những rủi ro an ninh mạng ngày càng cao.

Tổn thất do các cuộc tấn công mang gây ra không chỉ là mất mát tài chính lớn cho các công ty, lên tới hàng tỷ đồng, mà còn bao gồm cả hậu quả nghiêm trọng về danh tiếng và niềm tin của người tiêu dùng. Việc lộ lọt thông tin cá nhân của hàng triệu người dùng không chỉ ảnh hưởng đến sự riêng tư của họ mà còn tạo điều kiện cho các hành vi lừa đảo, gian lận và tội phạm khác diễn ra. Do đó, việc tìm ra giải pháp bảo mật thông tin là vô cùng cấp thiết, nhằm ngăn chặn các hành vi xâm phạm dữ liệu và tăng cường khả năng phòng vệ trước các mối đe dọa từ môi trường mạng.

Để có thể bảo vệ thông tin của mình một cách hiệu quả hơn trước những mối đe dọa ngày càng tinh vi từ tội phạm mạng. Cần phải có sự chuyển mình trong cách tiếp cận vấn đề bảo mật, từ việc chủ động phòng ngừa, phát hiện sớm các cuộc tấn công, đến việc ứng phó linh hoạt và phục hồi nhanh chóng sau khi xảy ra sự cố. Ngoài ra, việc nâng cao nhận thức bảo mật cho người dùng cũng là một yếu tố không thể bỏ qua, giúp mọi người trở nên tinh táo hơn trong việc bảo vệ dữ liệu cá nhân của mình.

Với vấn nạn các hacker ngày nay đang cực kỳ tinh vi hơn với sự bùng nổ các lỗ hổng lớn được tìm thấy và khai thác gây tổn thất hàng tỷ đồng cho công ty cùng với hàng triệu tài khoản người dùng, cá nhân bị lộ danh tính đã đặt ra câu hỏi lớn: “ Vậy làm sao để phòng chống tội phạm mạng, làm thế nào để bảo mật thông tin được tốt hơn trên mạng internet ?”

1.3. Mục Tiêu Cụ Thể

Trong thời đại thông tin hiện nay, việc bảo vệ dữ liệu và thông tin cá nhân trở nên vô cùng quan trọng, đặc biệt là trong bối cảnh mà các mối đe dọa an ninh mạng ngày càng phức tạp và khó lường. Steganography, một chiến thuật bảo mật từ lâu đời, đã được tái khám phá và áp dụng với những tiến bộ công nghệ mới, trở thành một trong những kỹ thuật tiên tiến trong lĩnh vực bảo mật thông tin. Sự ra đời của những kỹ thuật steganography hiện đại có thể xem là một bước tiến đột phá, mở ra cánh cửa mới cho việc bảo vệ thông tin một cách khéo léo và hiệu quả.

Nhận thức được tầm quan trọng này, nhóm chúng em đã quyết định đặt mục tiêu nghiên cứu và phát triển một hệ thống steganography, nhằm chế tác ra một phương pháp mới để ẩn thông tin mật trong các đối tượng như ảnh và âm thanh. Không chỉ dừng lại ở việc nghiên cứu lý thuyết, nhóm chúng em còn đặt ra mục tiêu cao hơn là hiện thực hóa nó thành một sản phẩm phần mềm có khả năng ứng dụng thực tiễn. Sản phẩm này không chỉ thể hiện được tính ứng dụng của steganography mà còn cung cấp một giải pháp bảo mật thông tin mới mẽ, giúp người dùng có thêm một lựa chọn trong việc bảo vệ dữ liệu của mình.

Báo cáo này sẽ đề cập đến quá trình phân tích, thiết kế và phát triển hệ thống steganography, từ việc tìm hiểu cơ sở lý thuyết đến việc triển khai và kiểm nghiệm sản phẩm. Chúng em sẽ đi sâu vào các phương pháp kỹ thuật, thuật toán liên quan, cũng như các thách thức và cách giải quyết trong quá trình phát triển. Mục tiêu cuối cùng là tạo ra một công cụ không chỉ mạnh mẽ về mặt kỹ thuật mà còn thân thiện với người dùng, dễ dàng tích hợp vào các hệ thống bảo mật hiện có và đủ linh hoạt để thích ứng với nhiều loại dữ liệu khác nhau.

Chúng em hy vọng rằng sản phẩm này không chỉ đáp ứng được nhu cầu bảo mật thông tin cá nhân và doanh nghiệp mà còn góp phần vào việc nâng cao nhận thức và kiến thức bảo mật cho cộng đồng. Kỳ vọng cao nhất là sản phẩm cuối cùng sẽ chứng minh được tính hiệu quả và độ tin cậy trong việc giấu tin, đồng thời mang lại giá trị thực tiễn cho người dùng trong việc chống lại các mối đe dọa an ninh mạng ngày càng tăng.

PHẦN 2: KỸ THUẬT GIẤU TIN – STEGANOGRAPHY

2.1. Giới Thiệu Chung

2.1.1. Lịch Sử Hình Thành

Việc sử dụng kỹ thuật giấu tin đầu tiên được ghi lại có thể bắt nguồn từ năm 440 trước Công nguyên ở Hy Lạp , khi Herodotus đề cập đến hai ví dụ trong Lịch sử của ông . Histiaeus đã gửi một thông điệp tới thuộc hạ của mình, Aristagoras , bằng cách cạo đầu người hầu thân tín nhất của ông, "đánh dấu" thông điệp trên da đầu của ông ta, sau đó tiễn ông ta lên đường khi tóc ông ta đã mọc lại, kèm theo lời chỉ dẫn, "Khi nào ngươi hãy đến Miletus, bảo Aristagoras cạo đầu và nhìn vào đó." Ngoài ra, Demaratus đã gửi cảnh báo về một cuộc tấn công sắp tới vào Hy Lạp bằng cách viết nó trực tiếp lên mặt sau **bằng gỗ** của một viên sáp trước khi bôi lên bè mặt sáp ong của nó. Các viên sáp sau đó được sử dụng phổ biến như các bè mặt viết có thể tái sử dụng, đôi khi được sử dụng để viết tốc ký .

Trong tác phẩm Polygraphiae của mình, Johannes Trithemius đã phát triển cái gọi là " Mật mã Ave-Maria " có thể che giấu thông tin bằng tiếng Latin ca ngợi Chúa. Ví dụ: " Auctor Sapientissimus Conseruans Angelica Deferat Nobis Charitas Potentissimi Creatoris " chứa từ ẩn VICIPEDIA .

Lần ghi chép đầu tiên của thuật ngữ này là vào năm 1499 do Johannes Trithemius trong cuốn sách Steganographia, một công trình nghiên cứu về mật mã học và kỹ thuật giấu tin, được ngụy trang dưới dạng một cuốn sách về ma thuật. Nói chung, thông điệp ẩn xuất hiện dưới dạng (hoặc là một phần của) một cái gì đó khác như hình ảnh, bài báo, danh sách mua hàng, văn bản bìa hoặc thông điệp bí mật có thể được viết bằng mực vô hình giữa các hàng của một lá thư riêng tư. Một số cách thực hiện kỹ thuật giấu tin thiếu bí mật chung là hình thức của bảo mật thông qua việc che giấu, và các lược đồ kỹ thuật giấu tin phụ thuộc vào khóa tuân thủ nguyên tắc của Kerckhoffs.

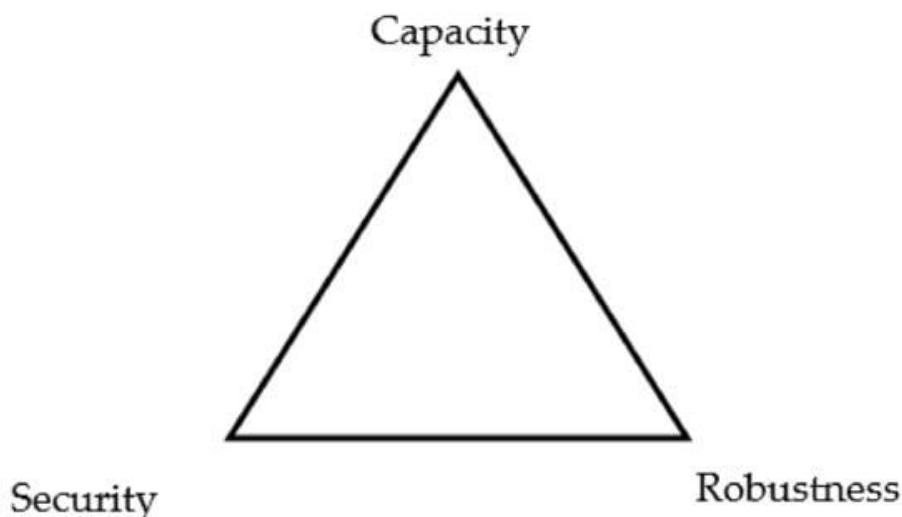
2.1.2. Khái Niệm

Kỹ thuật giấu tin hay kỹ thuật giấu thư, kỹ thuật ẩn mã (tiếng Anh: steganography) là nghệ thuật và khoa học về việc biểu diễn thông tin trong một thông điệp hoặc vật thể khác, sao cho sự hiện diện của thông tin đó không bị phát hiện bởi con người. Trong ngữ cảnh máy tính/điện tử, một tập tin, thông điệp, hình ảnh, hoặc video được ẩn trong một tập tin, thông điệp, hình ảnh, hoặc video khác. Từ steganography có gốc tiếng Hy Lạp có nghĩa là "ghi chép giấu kín", kết hợp từ hai từ steganos (<στεγανός>), nghĩa là "bị che đậy hoặc giấu kín", và -graphia (<γραφή>) nghĩa là "viết".

Theo Wikipedia

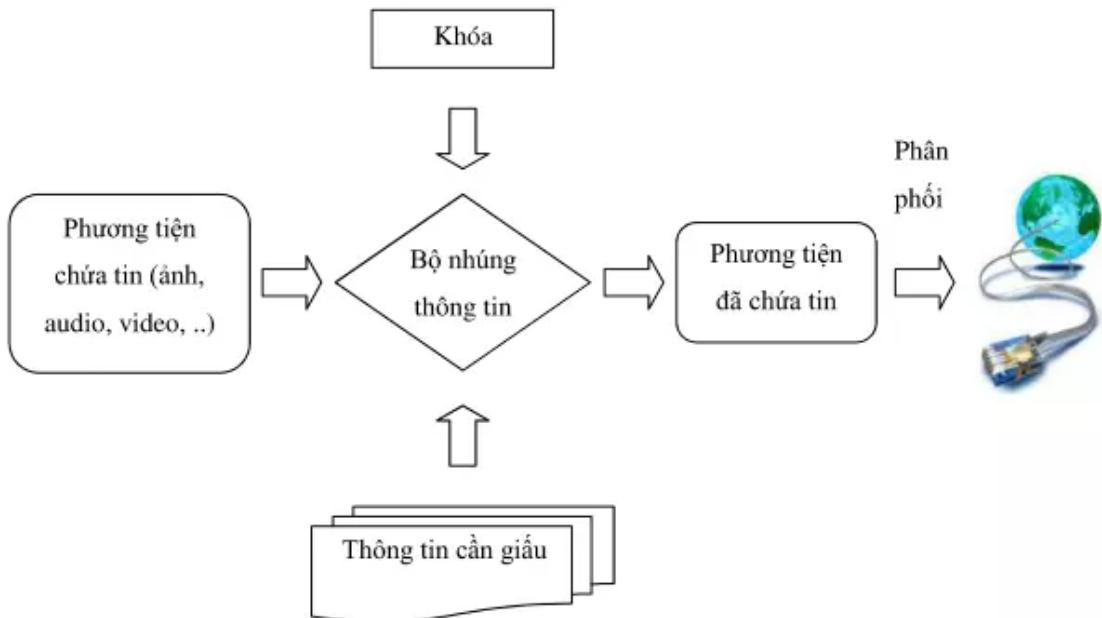
Hay nói theo cách khác steganography là một phương pháp của việc ẩn thông tin bằng cách giấu nó bên trong một đối tượng khác, nhằm mục đích không để lộ sự tồn tại của thông tin ẩn đó. Khác với mã hóa, nơi thông tin được chuyển đổi thành một dạng khó hiểu nhưng vẫn rõ ràng là có sự tồn tại của nó, steganography tập trung vào việc giấu đi sự hiện diện của thông tin.

Là một hình thức giao tiếp bí mật, kỹ thuật giấu tin đôi khi được so sánh với mật mã . Tuy nhiên, cả hai đều không giống nhau vì kỹ thuật giấu tin không liên quan đến việc xáo trộn dữ liệu khi gửi hoặc sử dụng khóa để giải mã khi nhận

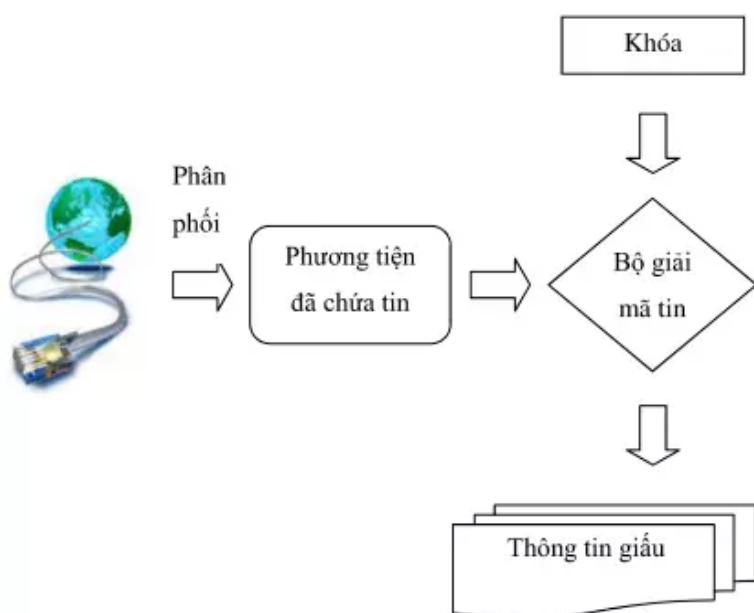


Hình 2.1: Các yêu cầu chính đối với hệ thống steganography.

Steganography có liên quan đến an ninh mạng vì các nhóm ransomware và các tác nhân đe dọa khác thường che giấu thông tin khi tấn công mục tiêu. Ví dụ: chúng có thể ẩn dữ liệu, che giấu công cụ độc hại hoặc gửi hướng dẫn cho máy chủ ra lệnh và kiểm soát. Họ có thể đặt tất cả thông tin này vào các tệp hình ảnh, video, âm thanh hoặc văn bản có vẻ vô hại.



Hình 2.1: Lược đồ thể hiện quá trình giấu tin



Hình 2.2: Lược đồ thể hiện quá trình giải mã thông tin

2.1.3. Steganography và Cryptography

Mặc dù 2 phương pháp này đều phương pháp quan trọng trong lĩnh vực bảo mật thông tin, nhưng chúng có những đặc điểm và mục đích sử dụng khác nhau:

Steganography

- Mục Tiêu: Steganography tập trung vào việc ẩn sự tồn tại của thông tin. Mục tiêu chính là đảm bảo rằng thông tin không bị phát hiện.
- Phương Pháp: Thông tin được giấu kín bên trong một đối tượng khác - như hình ảnh, âm thanh, video hoặc tài liệu văn bản - mà không làm thay đổi đáng kể đối tượng đó.
- Phát Hiện: Khi thực hiện đúng, thông tin ẩn bởi steganography rất khó phát hiện. Ngay cả khi đối tượng chứa thông tin bị phát hiện, việc xác định thông tin ẩn cụ thể có thể rất khó khăn.
- Ứng Dụng: Thường được sử dụng trong các hoạt động tình báo, bảo mật thông tin cá nhân và doanh nghiệp, nơi mà việc ẩn thông tin là quan trọng.

Cryptography

- Mục Tiêu: Cryptography nhằm mục đích bảo vệ nội dung của thông tin. Nó chuyển đổi thông tin gốc (plaintext) thành một dạng không thể đọc được (ciphertext).
- Phương Pháp: Sử dụng các thuật toán và khóa mã hóa để biến đổi thông tin, đảm bảo rằng chỉ những người có khóa giải mã phù hợp mới có thể truy cập thông tin.
- Phát Hiện: Trong khi thông tin mã hóa rõ ràng là có vẻ bất thường hoặc quan trọng, việc xác định nội dung cụ thể của thông tin đó mà không có khóa là rất khó, nếu không muốn nói là không thể.
- Ứng Dụng: Rộng rãi trong nhiều lĩnh vực từ giao dịch trực tuyến đến truyền thông an toàn giữa các tổ chức và cá nhân.

Sự Khác Biệt Chính

- **Ẩn Thông Tin vs Bảo Vệ Nội Dung:** Steganography giấu đi sự tồn tại của thông tin trong khi cryptography giữ thông tin an toàn bằng cách làm cho nó không thể đọc được mà không có gắng ẩn sự tồn tại của nó.

- Phương Pháp Tiếp Cận: Steganography sử dụng các phương tiện như ảnh hoặc âm thanh để che giấu thông tin. Cryptography, mặt khác, chuyển đổi thông tin thành một dạng mã hóa.
- Phát Hiện và An Toàn: Thông tin ẩn bằng steganography khó bị phát hiện nhưng không nhất thiết phải được bảo vệ khỏi việc đọc nếu nó được phát hiện. Cryptography bảo vệ thông tin từ việc đọc trái phép nhưng không giấu đi sự tồn tại của nó.

2.1.4. Đặc Điểm

- *Ẩn Sự Tồn Tại của Thông Tin:* Mục đích chính của steganography không chỉ là bảo vệ nội dung thông tin mà còn là ẩn đi sự tồn tại của chính thông tin đó.
- *Bảo Toàn Đối Tượng Gốc:* Trong steganography, thông tin được ẩn một cách kỹ lưỡng sao cho không làm thay đổi đáng kể hoặc làm hỏng đối tượng chứa thông tin.
- *Sử Dụng Các Phương Tiện Khác Nhau:* Thông tin có thể được ẩn trong nhiều loại đối tượng khác nhau như hình ảnh, tệp âm thanh, video, hoặc thậm chí trong văn bản.
- *Khó Phát Hiện và Phân Tích:* Một khi thông tin đã được ẩn một cách thành công, việc phát hiện và phân tích nó mà không có thông tin chi tiết hoặc công cụ cần thiết là rất khó khăn.
- *Đa Dạng Hóa Đối Tượng Chứa:* Steganography không giới hạn trong bất kỳ loại phương tiện cụ thể nào. Nó có thể được áp dụng trên nhiều loại đối tượng khác nhau, từ hình ảnh kỹ thuật số và âm thanh đến các trang web và tài liệu văn bản.
- *Khả Năng Tích Hợp Công Nghệ Mới:* Với sự phát triển của công nghệ, steganography cũng đã được tích hợp các công nghệ mới như trí tuệ nhân tạo (AI) và học máy để cải thiện khả năng ẩn và phát hiện thông tin một cách hiệu quả hơn.
- *Độc Lập với Mã Hóa:* Mặc dù cả steganography và mã hóa đều là các công cụ bảo mật thông tin, nhưng steganography không phụ thuộc vào việc mã hóa. Nó tập trung vào việc giấu thông tin, không nhất thiết phải biến thông tin thành dạng không thể đọc được.

- *Ứng Dụng Rộng Rãi:* Từ việc bảo vệ thông tin cá nhân, tới ứng dụng trong lĩnh vực quân sự và tình báo, steganography có nhiều ứng dụng trong các lĩnh vực khác nhau.

2.1.5. Ưu và Nhược Điểm

- **Ưu Điểm:**
 - + Giàu Sự Tồn Tại của Thông Tin: Điểm mạnh lớn nhất của steganography là khả năng ẩn thông tin, không chỉ là bảo vệ nội dung mà còn giấu đi sự tồn tại của thông tin đó.
 - + Khó Phát Hiện: Khi thực hiện đúng, thông tin ẩn thông qua steganography rất khó bị phát hiện, ngay cả bởi những người có kinh nghiệm trong lĩnh vực bảo mật.
 - + Đa Dạng Phương Tiện: Steganography có thể được áp dụng trong nhiều loại phương tiện khác nhau, bao gồm hình ảnh, âm thanh, video, và văn bản.
 - + Bảo Mật Thông Tin Cá Nhân và Doanh Nghiệp: Steganography hữu ích trong việc bảo vệ thông tin cá nhân và bí mật thương mại, đặc biệt trong môi trường có giám sát nghiêm ngặt.
 - + Ứng Dụng trong Tình Báo và Quân Sự: Trong các hoạt động tình báo và quân sự, steganography cung cấp một phương tiện để truyền thông tin mật một cách an toàn.
- **Nhược điểm:**
 - + Dung Lượng Thông Tin Giới Hạn: Lượng thông tin mà bạn có thể ẩn bằng steganography thường bị giới hạn bởi kích thước của đối tượng chứa thông tin.
 - + Phức Tạp trong Thực Hiện: Việc tạo và phát hiện thông tin ẩn qua steganography đòi hỏi kiến thức chuyên sâu và công cụ phức tạp.
 - + Dễ Bị Hỗn Hoặc Mất Mát: Nếu đối tượng chứa thông tin ẩn bị hỏng hoặc chỉnh sửa, thông tin ẩn có thể bị mất hoặc không thể truy cập.
 - + Cần Phương Tiện Truyền và Nhận Đặc Biệt: Cả người gửi và người nhận cần phải có phần mềm hoặc công cụ đúng đắn để tạo và giải mã thông tin ẩn.
 - + Khả Năng Bị Phát Hiện và Phân Tích: Mặc dù khó, nhưng không phải là không thể, để phát hiện và phân tích thông tin ẩn nếu kẻ tấn công biết chính xác phương pháp được sử dụng.

2.1.6. Ứng Dụng Thực Tiễn

- Bảo vệ quyền tác giả (copyright protection)
- Nhận thực thông tin hay phát hiện xuyên tạc thông tin (authentication and tamperdection)
- Giấu vân tay hay dán nhãn (fingerprinting and labeling)
- Điều khiển truy cập (copy control)
- Giấu tin mật (Steganography)

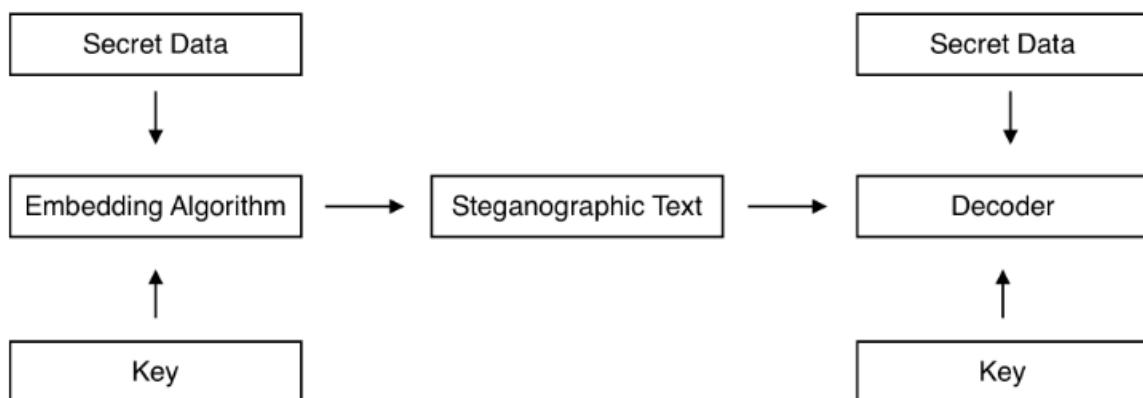
2.2. Các Loại Kỹ Thuật Giấu Tin

Các kỹ thuật giấu tin trong steganography rất đa dạng và phức tạp, tùy thuộc vào loại đối tượng chứa thông tin và mục đích sử dụng.

Trong khuôn khổ báo cáo này chỉ đề cập đến các kỹ thuật giấu tin trong công nghệ (không bao gồm hóa học và vật lý) được chia thành các mục sau:

2.2.1. Giấu Tin Trong Văn Bản (Text Steganography)

Giấu tin trong tài liệu văn bản (Text Steganography) là một phương pháp trong lĩnh vực steganography, nơi thông tin được ẩn trong các tài liệu văn bản để tránh sự phát hiện. Mục đích của text steganography là để gửi thông tin bí mật một cách kín đáo, đảm bảo rằng chỉ người nhận mong muốn mới có thể phát hiện và giải mã thông tin đó.

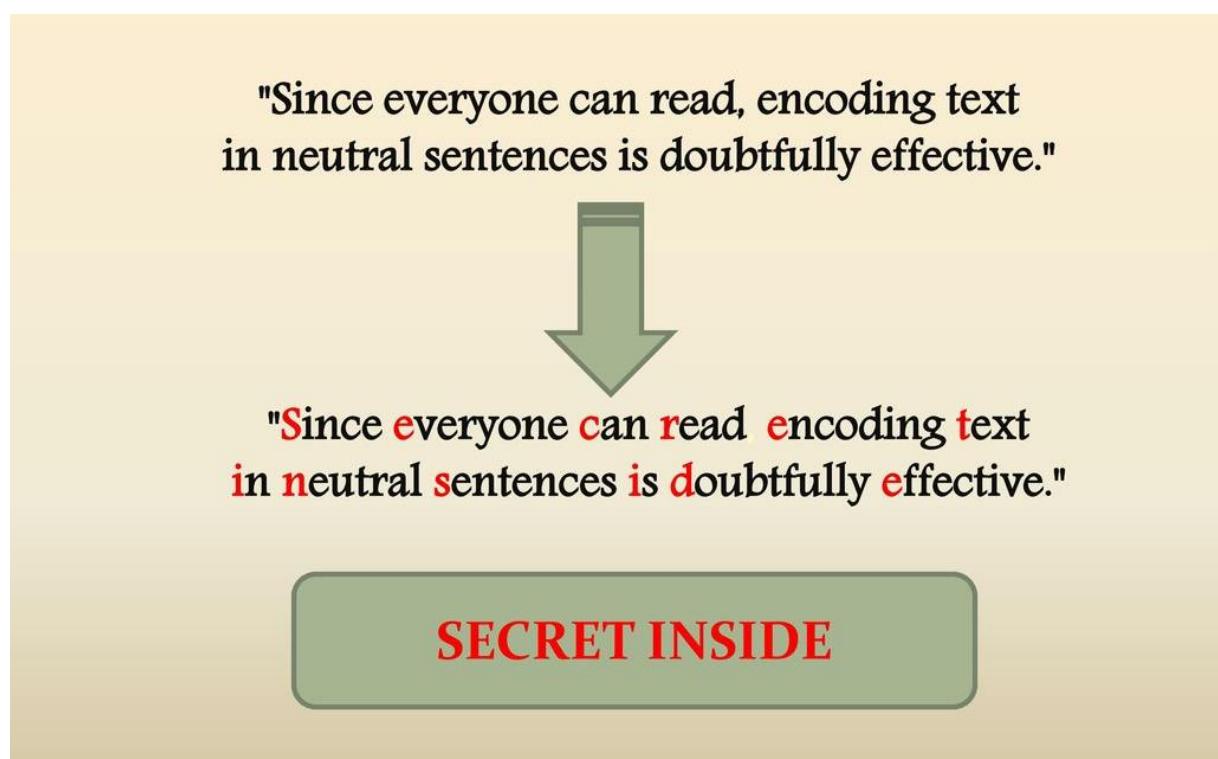


Hình 2.3: Sơ đồ giấu tin trong ảnh chung

Có nhiều phương pháp được sử dụng trong text steganography, bao gồm:

- *Sử dụng Khoảng Trắng và Định Dạng Văn Bản*: Thông tin bí mật có thể được ẩn thông qua việc thay đổi các khoảng trắng không nhìn thấy được, như thêm các khoảng trắng thừa, hoặc sử dụng các kỹ thuật định dạng văn bản đặc biệt.
- *Sử dụng Ký Tự Đặc Biệt*: Sử dụng các ký tự không in được như zero-width space (khoảng trống không rộng) giữa các từ để mã hóa thông tin. Khi xem tài liệu này trên một trình xử lý văn bản hỗ trợ, những ký tự này sẽ không hiển thị.
- *Sử dụng Cách Địệu Ngôn Ngữ*: Thông tin có thể được mã hóa thông qua việc sử dụng các cấu trúc ngôn ngữ đặc biệt, như sử dụng chơi chữ, ngôn ngữ lập trình, hoặc thậm chí qua cách sắp xếp các từ hoặc câu.

Ví dụ dưới đây sử dụng chơi chữ hoặc các từ đầu của mỗi dòng trong một đoạn văn để tạo ra một thông điệp bí mật



Hình 2.4: Ví dụ về giấu một đoạn chữ “secret inside” trong đoạn văn

- *Linguistic Steganography*: Đây là việc sử dụng ngôn ngữ và văn phong để ẩn thông tin, chẳng hạn như việc viết thơ hoặc sử dụng ngôn ngữ lập trình để tạo ra một thông điệp có ý nghĩa khác với những gì xuất hiện bì ngoài.
- *Sử dụng Cấu Trúc Tài Liệu*: Cấu trúc của tài liệu, như chương, đoạn văn, hoặc thậm chí là cách sắp xếp các đề mục, cũng có thể được sử dụng để mã hóa thông tin

Đó là những phương pháp chung cho giấu tin trong văn bản, được coi là một trong các phương pháp khó nhất trong các phương pháp giấu tin. Cùng tìm hiểu chi tiết qua các phương pháp chủ yếu được nhóm em tìm hiểu:

2.2.1.1. Phương Pháp Dựa Trên Định Dạng

Trong hình thức steganography này, các đặc điểm vật lý của ký hiệu văn bản được sử dụng. Các đặc điểm được thay đổi theo cách mà mắt người không thể cảm nhận được.

Ví dụ: các dòng trong văn bản được di chuyển lên xuống để che giấu các bit dữ liệu bí mật. Sau đó, các từ được di chuyển sang trái hoặc phải hoặc lên và xuống. Trong một số trường hợp, khoảng trắng giữa các từ hoặc giữa các đoạn văn, dòng được dùng để ẩn dữ liệu. Trong mã hóa dựa trên đặc điểm, đặc điểm vật lý của từ được thay đổi để che giấu thông tin. Điều này phụ thuộc vào các biểu tượng và ngôn ngữ.

2.2.1.1.1. Sử dụng khoảng trắng

Khoảng trắng trong văn bản có thể hiểu là khoảng cách giữa các, các câu hoặc các dòng Trong phương pháp sử dụng khoảng trắng, các khoảng trắng có thể được thêm vào sau mỗi từ, mỗi câu, mỗi đoạn.

INTER-WORD SPACING

The world under heaven, after a long period of division, tends to unite; after a long period of union, tends to divide. This has been so since antiquity. When the rule of the Zhou Dynasty weakened, seven contending kingdoms sprang up, warring one with another until the kingdom of Qin prevailed and possessed the empire. But when Qin's destiny had been fulfilled, arose two opposing kingdoms, Chu and Han, to fight for the mastery. And Han was the victor.

EOL SPACING

INTER-PARAGRAPH SPACING

The rise of the fortunes of Han began when Liu Bang the Supreme Ancestor slew a white serpent to raise the banners of uprising, which only ended when the whole empire belonged to Han (BC 202). This magnificent heritage was handed down in successive Han emperors for two hundred years, till the rebellion of Wang Mang caused a disruption. But soon Liu Xiu the Latter Han Founder restored the empire, and Han emperors continued their rule for another two hundred years till the days of Emperor Xian, which were doomed to see the beginning of the empire's division into three parts, known to history as The Three Kingdoms.

INTER-SENTENCE SPACING

Hình 2.5: Một số vị trí khoảng trắng có thể lựa chọn để giấu tin

Phương pháp sử dụng khoảng trắng giữa các từ sử dụng một hoặc hai khoảng trắng đặt giữa các từ liên tiếp, với quy ước một khoảng trắng tương ứng với bit 0 và hai khoảng trắng tương ứng với bit 1. Tuy nhiên, một vấn đề có thể xảy ra với phương pháp này là trong trường hợp hai từ cuối của một dòng có duy nhất một khoảng trắng (để cẩn lè chính xác) nhưng bit cần ẩn dấu lại là bit 1 (yêu cầu hai khoảng trắng). Để giải quyết vấn đề này, một thuật toán sử dụng các khoảng trắng để giấu thông tin được đề xuất như sau:

- Một khoảng trắng + một từ + hai khoảng trắng tương ứng với bit 0 được giấu.
- Hai khoảng trắng + một từ + một khoảng trắng tương đương với bit 1 được giấu.
- Một khoảng trắng + một từ + một khoảng trắng tương đương với không có thông tin được ẩn giấu.
- Hai khoảng trắng + một từ + hai khoảng trắng tương đương với không có thông tin được ẩn giấu.

Từ các quy ước trên dẫn đến kỹ thuật giấu tin và tách tin trong văn bản sử dụng khoảng trắng giữa các từ như sau:

Đầu vào:

- Thông điệp bí mật
- Văn bản phủ

Các bước thực hiện:

- Bước 1: Chuyển đổi thông điệp giàu thành dạng nhị phân
- Bước 2: Đọc dạng nhị phân của thông điệp bí mật và với mỗi bit nhị phân được chuyển đổi, thêm các khoảng trắng vào văn bản phủ theo quy ước được mô tả ở trên.

Đầu ra:

- Văn bản phủ có chứa thông điệp bí mật

Để hiểu rõ hơn về phương pháp này, hãy xét ví dụ dưới đây.

Với hai dữ liệu đầu vào là:

- Thông tin bí mật là chữ “H” chuyển sang dạng nhị phân có dạng “01001000”.
- Văn bản gốc:

Happy\families\are\all\alike\every\unhappy\family\is\unhappy
in\its\own\way\everything\was\in\confusion\in\the\oblonsk
ys\house.

Áp dụng nguyên tắc giàu tin trong văn bản sử dụng khoảng trắng giữa các từ thì thu được văn bản chử tin mật như sau:

Happy\families\are\all\alike\every\unhappy\family\is\nappy\in\its\own\way\every\thing\was\in\confusion\in\the\oblonskys\house.

Nhận xét: từ văn bản được giàu tin thấy được rằng, văn bản sau khi giàu tin nếu có độ rộng khoảng trắng giữa 02 từ là đủ lớn thì có thể quan sát được bằng mắt thường, chính vì vậy, người giàu tin phải định nghĩa lại độ rộng của mỗi khoảng trắng sao cho 02 khoảng trắng trông như 01 khoảng trắng nếu quan sát bằng mắt thường. Đây là phương pháp giàu tin đơn giản và dễ thực hiện. Tuy nhiên, nếu văn bản đã giàu tin được gõ lại bằng tay hoặc được xử lý bởi các trình xử lý tự động loại bỏ khoảng trắng thừa thì thông tin mật sẽ bị hủy. Chính vì vậy phương pháp này không được đánh giá cao và chỉ phù hợp với văn bản in.

2.2.1.1.2. Giấu tin vào cuối mỗi dòng

Nguyên tắc giấu tin vào cuối mỗi dòng dựa trên việc tận dụng các khoảng trắng thêm vào sau mỗi dòng có thể lưu trữ được một lượng lớn các bit. Các khoảng trắng ở cuối mỗi dòng có thể bị bỏ qua và không hiện lên các bởi các ứng dụng đọc văn bản. Trong toàn bộ văn bản, nếu giấu tin vào cuối mỗi dòng thì lượng bit thu được là rất lớn, có thể có đủ không gian để lưu trữ chuỗi bí mật.

Đầu vào:

- Thông điệp bí mật
- Văn bản phủ

Các bước thực hiện:

- Bước 1: Chuyển thông điệp bí mật thành dạng nhị phân
- Bước 2: Đọc dạng nhị phân của thông tin bí mật và thêm khoảng trắng vào cuối mỗi dòng theo quy ước: 0 dấu cách sẽ tìm đến câu tiếp theo và tương đương không có bit thông tin nào được giấu trong đó; 1 dấu cách sẽ mã hóa 0; 2 dấu cách sẽ mã hóa 1.

Đầu ra:

- Văn bản phủ có chứa thông điệp

Để hiểu rõ hơn về phương pháp này, hãy xét ví dụ dưới đây:

Thông điệp bí mật là chữ “H” chuyển sang dạng nhị phân có dạng “01001000”

Văn bản gốc:

“Quê hương là một tiếng ve
Lời ru của mẹ trưa hè à ơi
Dòng sông con nước đầy voi
Quê hương là một góc trời tuổi thơ
Quê hương ngày ấy như mơ
Tôi là cậu bé dại khờ đáng yêu
Quê hương là tiếng sáo diều
Là cánh cò trắng chiều chiều chân đê.”

Văn bản đã giàu tin sử dụng kỹ thuật :

“Quê hương là một tiếng ve_
Lời ru của mẹ trưa hè à ơi_ _
Dòng sông con nước đây voi_ _
Quê hương là một góc trời tuổi thơ_ _
Quê hương ngày ấy như mơ_ _
Tôi là cậu bé dại khờ đáng yêu_ _
Quê hương là tiếng sáo diều_ _
Là cánh cò trắng chiều chiều chân đê.”

Chú ý: (dấu “_” ở đây là dấu cách)

Nhận xét: Thêm một hoặc hai khoảng trắng vào cuối mỗi dòng trong văn bản cũng là một phương pháp giàu tin đơn giản. Những khoảng trắng được thêm vào sẽ không xuất hiện khi văn bản được in ra nhưng có thể dễ dàng bị phát hiện bởi các bộ xử lý và thông điệp bí mật sẽ bị hủy bỏ khi mà văn bản được gõ lại bằng tay hoặc được xử lý bởi một chương trình xử lý tự động loại bỏ khoảng trắng thừa. Chính vì vậy, phương pháp này cũng không được đánh giá cao.

2.2.1.1.3. Dịch chuyển vị trí dòng

Trong phương pháp này, các dòng của văn bản sẽ được dịch chuyển theo chiều đọc với một độ dài nhất định, ví dụ mỗi dòng sẽ được dịch chuyển một khoảng rất nhỏ khoảng 1/300 inch lên hoặc xuống (inch là một đơn vị chiều dài trong hệ thống đo lường) và thông tin sẽ được ẩn giấu bằng việc tạo ra các hình dạng của khoảng dịch chuyển của văn bản. Thông điệp sẽ được giấu vào khoảng dịch chuyển đó bằng cách chèn vào các bit 0 hoặc 1 tùy theo quy ước. Điều này rất khó có thể phát hiện bằng mắt thường vì khoảng cách thay đổi khá nhỏ.

Đầu vào:

- Thông điệp bí mật
- Văn bản phủ

Các bước thực hiện:

- Bước 1: Chuyển thông tin bí mật thành dạng nhị phân
- Bước 2: Đọc dạng nhị phân của thông điệp bí mật và dịch chuyển vị trí các dòng theo quy ước dòng được dịch giảm đi tương ứng bit 0 được giấu, dòng được dịch tăng lên tương ứng bit 1 được giấu.

Đầu ra:

- Văn bản chứa thông điệp.

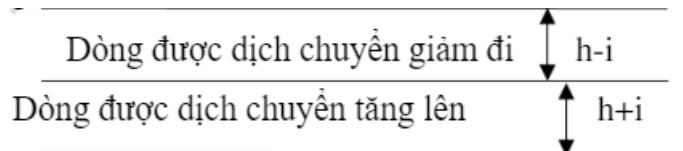
Ví dụ:

Thông điệp cần giấu là 2 bit “01”

Văn bản gốc:

Dòng bình thường
Dòng bình thường

Văn bản sau khi giấu tin:



Nhận xét: trong phương pháp dịch chuyển vị trí dòng khoảng cách có thể được chú ý bằng cách sử dụng các công cụ đánh giá khoảng cách đặc biệt. Ngoài ra nếu văn bản được gõ lại bằng tay hoặc nếu chương trình nhận dạng ký tự được sử dụng, thông tin bí mật sẽ bị phá hủy. Chính vì vậy phương pháp này chỉ phù hợp với các văn bản in, để tránh thay đổi định dạng của văn bản

2.2.1.1.4. Dịch chuyển vị trí từ

Kỹ thuật giấu tin trong văn bản sử dụng phương pháp dịch chuyển vị trí các từ dựa trên cơ chế giống như dịch chuyển vị trí dòng, nhưng người giấu tin thay vì dịch chuyển vị trí của các dòng thì sẽ dịch chuyển vị trí các từ. Sau đó tùy vào khoảng dịch

chuyển đó nằm bên trái hay bên phải từ mà quy định nó là bit 0 hay bit 1. Dịch trái sẽ là bit 0 còn dịch phải sẽ là bit 1.

Đầu vào:

- Thông điệp bí mật
- Văn bản phủ

Các bước thực hiện:

- Bước 1: Chuyển thông tin bí mật thành dạng nhị phân
- Bước 2: Đọc dạng nhị phân của thông điệp bí mật và dịch chuyển vị trí các dòng theo quy ước dịch trái sẽ là bit 0 còn dịch phải sẽ là bit 1.

Đầu ra:

- Văn bản phủ có chứa thông điệp

Ví dụ:

Thông điệp bí mật là chữ “A” chuyển sang dạng nhị phân có dạng “01000001”

Văn bản gốc và văn bản sau khi giấu tin:

The quick brown fox jumps over the lazy dog.

Văn bản sau khi đã giấu tin là:

The quick brown fox jumps over the lazy dog.

Trong ví dụ trên, dòng thứ nhất các chữ có vị trí không bị thay đổi. Còn dòng thứ 2 các chữ bị dịch trái hoặc phải để mã hóa cho chuỗi 01000001. Độ dịch chuyển là rất nhỏ và nếu không so sánh với chuỗi ban đầu thì khó có thể nhận biết được chuỗi đó có bị thay đổi hay không. Phương pháp này cũng phù hợp với văn bản in và nếu văn bản được gõ lại bằng tay hoặc nếu chương trình nhận dạng ký tự được sử dụng, thông tin bí mật sẽ bị phá hủy

2.2.1.2. Phương pháp sinh ngẫu nhiên và thống kê

2.2.1.2.1. Sử dụng văn phạm phi ngữ cảnh

Việc ẩn giấu thông tin trong các văn bản nhân tạo phải vượt qua được các cơ chế kiểm tra bằng máy tính. Những văn bản đó phải đáp ứng ít nhất các yêu cầu sau đây:

- Tần suất xuất hiện của các chữ cái trong văn bản phải giống như trong ngôn ngữ tự nhiên. Nếu văn bản sử dụng ngôn ngữ tiếng Anh, chữ E và T phải là những chữ cái thường xuyên xuất hiện nhất, còn Z và Q phải là những chữ cái ít xuất hiện nhất.
- Hầu hết các từ trong văn bản được sử dụng phải là các từ được liệt kê trong các từ điển chính quy. Nếu trong các văn bản có quá nhiều các từ như tên riêng, tiếng lóng hay thuật ngữ khoa học khi bị kiểm tra bởi các máy tính sẽ bị đánh dấu là những văn bản đáng ngờ.
- Các câu trong văn bản phải đúng cú pháp. Nếu một chương trình kiểm tra cú pháp tự động tìm thấy một lỗi cú pháp như hai động từ liên tiếp trong một văn bản thì sẽ bị đánh dấu là một văn bản đáng ngờ.

Phương pháp sử dụng văn phạm phi ngữ cảnh sẽ sử dụng một văn phạm phi ngữ cảnh (CFG - Context Free Grammar) để sinh ra các câu tạo thành văn bản nhân tạo chứa thông điệp bí mật và văn bản này có thể bắt chước các văn bản thực tế (nghĩa là có các thuộc tính thống kê giống nhau). Văn phạm phi ngữ cảnh là một tập hợp hữu hạn các biến (còn gọi là các ký hiệu chưa kết thúc), mỗi biến biểu diễn một ngôn ngữ. Ngôn ngữ được biểu diễn bởi các biến được mô tả một cách đệ quy theo thuật ngữ của một khái niệm khác gọi là ký hiệu kết thúc. Quy tắc quan hệ giữa các biến gọi là luật sinh. Mỗi luật sinh có dạng một biến, ở về trái sinh ra một chuỗi có thể gồm biến lẫn các ký hiệu kết thúc trong văn phạm. Văn phạm phi ngữ cảnh là một hệ thống gồm bốn thành phần, ký hiệu là văn phạm G (V, T, P, S), trong đó:

- V là tập hữu hạn các biến (hay ký tự chưa kết thúc).
- T là tập hữu hạn các ký tự kết thúc, $V \cap T = \emptyset$
- P là tập hữu hạn các luật sinh mà mỗi luật sinh có dạng $A \rightarrow a$ (với A là biến và a là chuỗi các ký hiệu $\in (V \cup T)^*$)
- S là một biến đặc biệt gọi là ký hiệu bắt đầu văn phạm

Các quy tắc sau chỉ ra cách để tạo ra một chuỗi (gồm các ký hiệu kết thúc) từ một CFG cho trước:

- Sử dụng một kí hiệu bắt đầu (một kí hiệu chưa kết thúc đặc biệt) để khởi đầu. Chọn một luật có kí hiệu khởi đầu ở bên trái và chọn một kí hiệu ở phần bên phải của luật để thay thế kí hiệu khởi đầu này. Kí hiệu được sử dụng sẽ là một thành tố của văn bản sẽ được tạo sau này.
- Chọn một kí hiệu chưa kết thúc trong văn bản, tìm một luật có kí hiệu chưa kết thúc này ở bên trái và thay thế kí hiệu này bằng một kí hiệu ở phần bên phải của luật.
- Lặp lại bước 2 cho đến khi văn bản chỉ còn toàn các kí hiệu kết thúc

Sau đây là một ví dụ của một CFG dùng để giải chuỗi 0100110 trong đó các kí hiệu chưa kết thúc là các kí hiệu in đậm, các kí hiệu kết thúc là các kí hiệu viết thường.

Start → adjective noun tense verb

adjective → the size | a size

size → tiny | small | large | big

noun → saw | ladder | truth | boy

tense → is | was

verb → waiting | standing

Kí hiệu chưa kết thúc đầu tiên là **adjective**. Luật cho kí hiệu chưa kết thúc này có hai sự lựa chọn tương đương (2^1) do đó một bit có thể được ẩn giấu bằng cách chọn một sự lựa chọn bên phải. Bit đầu tiên được giấu là 0, sự lựa chọn đầu tiên là (the size). Kí hiệu kết thúc “the” được nối vào văn bản còn kí hiệu chưa kết thúc size sẽ được thay thế tiếp. Từ size có bốn sự lựa chọn tương đương (2^2) nên có thể giấu được hai bit. Hai bit tiếp theo trong chuỗi cần giấu là bit 1 và bit 0, nên sự lựa chọn thứ ba, “large” sẽ được chọn. Kí hiệu chưa kết thúc tiếp theo là **noun**. Theo luật, có bốn sự lựa chọn nên có thể dùng hai bit tiếp theo trong chuỗi ẩn giấu. Ở đây, hai bit tiếp theo là bit 0 và bit 1 nên “ladder” được chọn. Sự lựa chọn cho **tense** là “was” (được chỉ ra bởi bit 1 tiếp theo), và sự lựa chọn cho **verb** là “waiting”, bởi vì bit cuối cùng là 0. Sau khi giấu tin thu được câu “the large ladder was waiting”. Câu này có thể gây nghi ngờ khi được đọc bởi một người bình thường nhưng nó lại dễ dàng vượt qua các kiểm tra của máy tính. Bộ mã hóa có thể dễ dàng kết thúc mỗi câu (tức là

sau các lựa chọn của **verb**) với một dấu chấm câu và bắt đầu câu tiếp theo với một chữ cái viết hoa, để tăng tính thực tế cho các văn bản nhân tạo được sinh ra.

Cụ thể thuật toán như sau:

Đầu vào:

- Thông điệp bí mật
- Bộ từ điển và các luật sinh

Các bước thực hiện:

- Bước 1: Chuyển đổi thông điệp bí mật sang dạng nhị phân
- Bước 2: Sinh chuỗi văn bản (G):
 - ✓ Nếu G rỗng: xuất “”
 - ✓ Nếu G là 1 ký tự kết thúc A: xuất “A”
 - ✓ Nếu G là 1 ký tự không kết thúc: với từng c trong luật sinh của G:
bắt đầu một sinh chuỗi (c) cho tới khi tất cả các luật sinh đều được duyệt
- Văn bản nhân tạo chứa thông điệp bí mật

Một CFG lớn với nhiều lựa chọn có thể ẩn nhiều bit hơn và tạo ra những câu có ý nghĩa tự nhiên hơn. Một CFG là không rõ ràng nếu một câu có thể được tạo ra bằng cách chọn những luật theo những thứ tự khác nhau. Điều này được thể hiện qua ví dụ sau:

Start → name action | who does

name → Alice | Bob

action → is here | was there

who → Alice is | Bob was

does → here | there

Câu “Alice is here” có thể sinh ra bằng cách thay thế những kí hiệu không kết thúc **name action** bằng “Alice” và “is here”, đồng thời cũng có thể sinh ra bằng cách thay thế **who does** bằng “Alice is” và “here”. Hiển nhiên, một CFG sẽ tạo ra những văn bản có thể giải mã theo những cách khác nhau. CFG này là không phù hợp để ẩn giấu thông tin.

Một CFG ở dạng Greibach normal form (GNF) nếu kí hiệu chưa kết thúc luôn là sự lựa chọn cuối cùng trong các lựa chọn của luật sinh. Ví dụ, luật sinh có dạng **something** → A **B** | C **D** thuộc dạng GNF nhưng luật sinh dạng “**blah** → the **size** sum | a **size bell**” thì không thuộc dạng GNF. Tuy nhiên có thể sửa một luật không thuộc dạng GNF thành GNF bằng cách thêm các luật như sau:

adjective → the **sizesum** | a **sizebell**

sizesum → tiny sum | small sum | large sum | big sum

sizebell → tiny bell | small bell | large bell | big bell

Bộ giải mã sử dụng 1 CFG trong GNF để phân tích cú pháp, như trong ví dụ sau đây:

Start → **noun verb**

noun → Alice | Bob

verb → sent mail **to** | sent email **to**

to → to **rel recipient**

rel → all | some

recipient → friends | relatives

Để ẩn giấu chuỗi nhị phân 01010, bộ mã hóa chọn “Alice” cho bit đầu tiên (0) và “sent email to” cho bit thứ hai (1). Không từ nào có thể giấu thông tin khi luật thứ tư được áp dụng (bởi vì không có sự lựa chọn nào) nhưng bộ mã hóa sinh ra được kí hiệu kết thúc “to”, sau đó sử dụng luật sinh cho rel để chọn “all” cho bit thứ ba (0), và luật sinh cho recipient để chọn “relatives” cho bit thứ tư (1). Để ẩn bit thứ năm, bộ giải mã bắt đầu câu tiếp theo. Câu “Alice sent email to all relatives” có thể dễ dàng giải mã theo các thành phần cú pháp của câu và xác định được các bit ẩn.

Nhận xét: Ở ví dụ trên, chỉ có thể ẩn giấu được 4 bit trong câu “Alice sent email to all relatives” có độ dài 33 kí tự (bao gồm các khoảng trắng). Khả năng giấu tin trong ví dụ này chỉ là $4 / (33 * 8) \approx 0.015$ bit ẩn giấu trên mỗi bit của văn bản được tạo ra để giấu tin. Phương pháp này có thể tăng được tính hiệu quả nếu mỗi luật sinh có nhiều lựa chọn ở phần bên phải. Một luật sinh với $2n$ sự lựa chọn có thể giấu được n bit. Nếu một lựa chọn là một từ có 4 chữ cái (tức là 32 bit) và có $1024 = 210$ lựa chọn trong luật sinh, thì 10 bit có thể ẩn giấu trong mỗi 32 bit của văn bản được sinh ra, dẫn đến khả năng giấu tin là $10 / 32 = 0.3125$ bits/bit (bpb). Phương pháp này đã được cài đặt và

kiểm thử rộng rãi bởi tác giả Peter Wayner. So sánh với các thuật toán giấu tin khác, phương pháp này được đánh giá là đơn giản và có hiệu quả tốt.

2.2.1.2.2. Dựa trên tính phản xạ đối xứng của ký tự

Trong hầu hết các thuật toán giấu tin, thông điệp bí mật được ẩn giấu bằng cách thay đổi cấu trúc của văn bản chứa do đó khả năng bị nghi ngờ hay mất mát dữ liệu khi gõ lại văn bản theo cấu trúc chính xác là có thể xảy ra. Để tránh xảy ra khả năng này cũng như tăng cường tính bảo mật, thay vì giấu các bit bí mật bằng cách thay đổi cấu trúc của văn bản chứa, phương pháp này sẽ giấu các thông điệp bí mật bằng cách tạo ra một văn bản tóm tắt thu thập từ các bài báo hay bất kỳ một phương tiện văn bản thông tin đại chúng. Quá trình tạo ra văn bản tóm tắt phụ thuộc vào tính phản xạ đối xứng của bảng chữ cái tiếng Anh. Dựa vào tính chất này, bảng chữ cái được chia thành các bộ khác nhau, mỗi bộ đại diện cho một cặp bit. Để thực hiện điều này cần phân tích tính phản xạ đối xứng của bảng chữ cái tiếng Anh và phân loại chúng để thể hiện các bit. Các thuộc tính và các bit thể hiện tương ứng được trình bày như sau **Error! Reference source not found.**

Phân loại bảng chữ cái tiếng Anh:

Để phân loại bảng chữ cái tiếng Anh theo tính phản xạ đối xứng, đầu tiên chọn chiều ngang là trực đối xứng và phân chia các chữ cái tiếng Anh thành hai nhóm. Các chữ cái sau khi chia theo chiều ngang, nếu thu được hai phần giống hệt nhau, ví dụ chữ ‘B’, ‘H’,... xếp vào một nhóm, ngược lại, các chữ cái sau khi chia theo chiều ngang, nếu thu được hai phần không giống nhau, ví dụ chữ ‘A’, ‘F’,... xếp vào nhóm còn lại. Toàn bộ phân loại dựa trên logic này được trình bày trong bảng phân nhóm dựa trên tính phản xạ đối xứng theo trực ngang sau:

ID Nhóm	Tên Nhóm	Chữ cái trong nhóm	Bit được giấu
1	Tính phản xạ đối xứng không được tuân thủ	A, F, G, J, L, M, N, P, Q, R, S, T, U, V, W, Y, Z	0
2	Tính phản xạ đối xứng được tuân thủ	B, C, D, E, H, I, K, O, X	1

Áp dụng tương tự với trực dọc, thu được hai nhóm như bảng sau:

ID Nhóm	Tên Nhóm	Chữ cái trong nhóm	Bit được giấu
1	Tính phản xạ đối xứng không được tuân thủ	B, C, D, E, F, G, J, K, L, N, P, Q, R, S, Z	0
2	Tính phản xạ đối xứng được tuân thủ	A, H, I, M, O, T, U, V, W, X, Y	1

Kết hợp cả hai khái niệm được mô tả trong 2 bảng có thể phân loại bảng chữ cái tiếng Anh thành bốn nhóm dựa trên cả hai chiều ngang và chiều dọc của chữ cái. Các chữ cái không đối xứng trên cả hai trực, ví dụ chữ ‘F’, chữ ‘G’,... được chia vào một nhóm. Các chữ cái đối xứng theo trực ngang, ví dụ chữ ‘B’, chữ ‘D’,... được chia vào một nhóm. Các chữ cái đối xứng theo trực dọc, ví dụ chữ ‘A’, chữ ‘M’ được chia vào một nhóm. Các chữ cái đối xứng theo cả hai trực, ví dụ chữ ‘H’, chữ ‘I’ được chia vào một nhóm. Cụ thể, thu được bảng phân nhóm dựa trên tính phản xạ đối xứng theo trực ngang và trực dọc sau:

ID Nhóm	Tên Nhóm	Chữ cái trong nhóm	Bit được giấu
1	Tính phản xạ đối xứng không được tuân thủ trên cả hai trực	F, G, J, L, N, P, Q, R, S, Z	00
2	Tính phản xạ đối xứng được tuân thủ trên trực ngang	B, C, D, E, K	01
3	Tính phản xạ đối xứng được tuân thủ trên trực dọc	A, M, T, U, V, W, Y	10
4	Tính phản xạ đối xứng được tuân thủ trên cả hai trực	H, I, O, X	11

Giấu tin trong văn bản sử dụng tính phản xạ đối xứng của bảng chữ cái tiếng Anh

Theo như sự phân loại bảng chữ cái tiếng Anh mô tả như trong bảng trên dựa vào thông điệp bí mật và văn bản phủ được lựa chọn để sinh ra một văn bản tóm tắt có chứa thông điệp bí mật. Văn bản tóm tắt sẽ gồm các câu được trích ra từ văn bản lựa chọn. Các câu này sử dụng chữ cái đầu tiên của từ đầu tiên làm đại diện cho các cặp bit của thông điệp cần ẩn giấu.

Đầu vào:

- Thông điệp bí mật
- Một văn bản bằng ngôn ngữ tiếng Anh

Các bước thực hiện:

- Bước 1: Biến đổi thông điệp bí mật thành chuỗi bit nhị phân
- Bước 2: Kiểm tra xem tổng độ dài của chuỗi bit là chẵn hay lẻ. Nếu lẻ, phải thêm 1 bit ‘0’ vào cuối chuỗi bit nhị phân. Bây giờ có thể chia chuỗi bit tổng thành các cặp bit liên tiếp.
- Bước 3: Chuyển đổi toàn bộ các ký tự của văn bản đầu vào thành các chữ cái viết hoa.
- Bước 4: Với từng cặp bit, xem xét chữ cái đầu tiên của từ đầu tiên trong câu:
 - ✓ Nếu chữ cái đó nằm trong nhóm đại diện cho cặp bit đang xem xét, chọn câu này và đưa vào văn bản chúa.
 - ✓ Nếu chữ cái đó không nằm trong nhóm đại diện cho cặp bit đang xem xét, bỏ qua câu này và chọn câu tiếp theo
- Bước 5: Quá trình tiếp diễn cho đến khi toàn bộ chuỗi bit của thông điệp bí mật được thực thi hết
- Bước 6: Văn bản mã hóa thu được là bản tóm tắt của văn bản đầu vào và được gửi đến người nhận

Đầu ra:

- Văn bản chúa thông điệp bí mật, hay chính là văn bản tóm lược của văn bản đầu vào

Giấu tin trong văn bản sử dụng tính phản xạ đối xứng của bảng chữ cái tiếng Việt

Phương pháp giấu tin dựa trên tính phản xạ đối xứng của ký tự có thể áp dụng cho việc giấu tin của các văn bản bằng tiếng Việt. Về bản chất, bảng chữ cái tiếng Việt cũng cùng hệ Latin với bảng chữ cái tiếng Anh, do đó có thể cải tiến phương pháp này để giấu tin cho văn bản tiếng Việt như sau:

- Không cần phải kiểm tra các mạo từ như “A”, “The” như trong phương pháp giấu thông tin bằng Tiếng Anh
- Với thông điệp ẩn giấu, phải sử dụng bảng mã Unicode thay cho bảng mã ASCII. Tuy nhiên, để tối ưu hóa thông điệp ẩn giấu, sử dụng bảng mã ASCII để chuyển

các ký tự thông dụng như các chữ cái trùng với bảng chữ cái tiếng Anh, chữ số, các dấu chấm câu,... Các ký tự này khi chuyển sang dạng chuỗi bit sẽ có độ dài là 1 byte (8-bit). Đối với các ký tự đặc biệt của riêng bảng chữ cái tiếng Việt (bao gồm cả chữ hoa, chữ thường và các dấu của tiếng Việt như huyền, sắc, hỏi, ngã, nặng), ví dụ như “Á”, “à”, “ê”, “đ”,... dùng phép mã hóa UTF-8 và với tùy ký tự sẽ thu được chuỗi bit có độ dài là 2 bytes (16-bit) hoặc 3 bytes (24-bits). Do các ký tự đặc biệt có độ dài lớn hơn các ký tự thuộc bảng mã ASCII nên văn bản dùng để chứa thông điệp cần giàu sẽ có kích thước lớn hơn.

- Với hệ thống bản đồ để mã hóa các cặp bit, phải thêm các chữ cái tiếng Việt, ví dụ như “Á”, “À”, “Ã”, “Â”, “Đ”, “Ê”, “Ô”, “U”,... và phải bỏ các chữ cái “F”, “J”,... của bảng chữ cái tiếng Anh... Chi tiết bản đồ mã hóa các ký tự thành các cặp bit như sau:

ID Nhóm	Tên nhóm	Chữ cái trong nhóm	Bit được giấu
1	Tính đối xứng không được tuân thủ trên cả hai trục	Á,À,Ã,Â,Ă,Ă,Ă, Ă,À,Ã,Â,Ã,É,È, Ê,È,É,Ê,É,Ê,Ê, Í,Ì,Ï,Î,G,L,N,Ó, Ò,Õ,Ô,Ó,Ô,Õ,Ô, Ó,Ó,Ò,Õ,Ó,Ô,P, Q,R,S,Ú,Ù,Ü,Ü, U,Ú,Ù,Ü,U,Ü,Ý, Ỳ,Ỳ,Ỳ	00
2	Tính đối xứng tuân thủ trên trục ngang	B,C,D,Đ,E,K	01
3	Tính đối xứng được tuân thủ theo trục dọc	A,À,Ã,Â,Ă,Ă,I, M,Ó,Ô,T,U,U,V, Y,Ỳ	10
4	Tính đối xứng được tuân thủ trên cả hai trục	H,I,O,X	11

2.2.1.3. Phương pháp sử dụng tính chất ngôn ngữ

2.2.1.3.1. Sử dụng cú pháp

Phương pháp giấu tin sử dụng cú pháp dựa trên hai ý tưởng:

- Sử dụng các dấu chấm câu một cách không rõ ràng:

Ý tưởng của quá trình sử dụng các dấu chấm câu không rõ ràng khá đơn giản nhưng dễ bị phát hiện bởi vì việc sử dụng những dấu chấm câu không nhất quán sẽ gây ra sự chú ý. Một ví dụ về ý tưởng sử dụng dấu chấm câu nhập nhằng là việc sử dụng cụm từ “Bánh mì, bơ, và sữa” và sử dụng cụm từ “Bánh mì, bơ và sữa.”. Cả hai cụm từ trên đều đúng về mặt cú pháp, thay đổi luân phiên giữa hai hình thức có thể ẩn giấu dữ liệu nhị phân và có thể quy ước thành cú mỗi lần xuất hiện cấu trúc đầu tiên (dấu “,” xuất hiện ở trước chữ “và”) sẽ ẩn giấu bit “1”, khi xuất hiện cấu trúc thứ hai (không có dấu “,” trước chữ “và”) sẽ ẩn dấu bit “0”. Sử dụng phương pháp này có ưu điểm là khó bị phát hiện nhưng nhược điểm là hiệu suất giấu tin không cao và chỉ áp dụng được cho cú pháp nhất định.

- Sửa đổi văn bản sao cho ý nghĩa của văn bản vẫn được bảo toàn:

Ý tưởng của việc sửa đổi này an toàn, nhưng lại khó thực thi vì khả năng “hiểu” của các máy tính với ngôn ngữ của con người là hạn chế. Việc chỉnh sửa văn bản mà vẫn giữ được ý nghĩa của văn bản là một thủ thuật tinh vi để ẩn giấu thông tin. Nghĩa của hai cụm từ “Khi con học xong, con có thể đi chơi” và “con có thể đi chơi khi con học xong.” là tương đương và thông tin có thể được ẩn giấu dựa vào cú pháp thay đổi này. Phương pháp này giấu được ít thông tin và có hiệu suất thấp vì phải có sự điều chỉnh thủ công. Tuy nhiên phương pháp này an toàn vì rất khó để máy tính có thể phát hiện ra sự thay đổi cú pháp giữa hai cụm từ. Kẻ tấn công phải thực sự đọc các tin nhắn một cách thủ công và xác định những cụm từ có liên quan để trích xuất dữ liệu.

2.2.1.3.1. Sử dụng ngữ nghĩa

Phương pháp sử dụng ngữ nghĩa là phương pháp mà dữ liệu được giấu trong văn bản bằng cách sử dụng các từ đồng nghĩa. Ở phương pháp này thì người gửi và người nhận sẽ sử dụng cùng một bộ từ điển trực tuyến nhất định. Đây có thể coi là một tập dữ liệu chứa những từ quy ước và bộ giải mã sẽ đọc từng từ trong văn bản và tìm kiếm những từ tương ứng trong từ điển:

Sử dụng từ đồng nghĩa:

Các bộ giải mã đọc từng từ trong văn bản và tìm kiếm từ đồng nghĩa trong từ điển, nếu một từ chẳng hạn như “god child”, không có từ đồng nghĩa nào thì các bộ giải mã giả định rằng không có dữ liệu nào được giấu trong đó. Nếu từ “child” là đầu vào, và chính nó xuất hiện trong từ điển như một từ đồng nghĩa trong danh sách (bud, chick, child, kid, minor), sau đó, danh sách này sẽ được xem như là để che giấu hai bit và “child” (là từ thứ 3 trong danh sách) được hiểu là ẩn giấu số 2 bit 2 (01 trong hệ nhị phân).

Đầu vào:

- Thông điệp bí mật
- Văn bản phủ

Các bước thực hiện:

- Bước 1: Chuyển thông tin mật thành dạng nhị phân.
- Bước 2: Kiểm tra văn bản và chọn một từ xuất hiện nhiều lần
- Bước 3: Sử dụng từ điển đặc biệt chọn những từ đồng nghĩa để thay cho từ được chọn theo quy ước và chuỗi nhị phân.

Đầu ra:

- Văn bản phủ có chứa thông điệp.

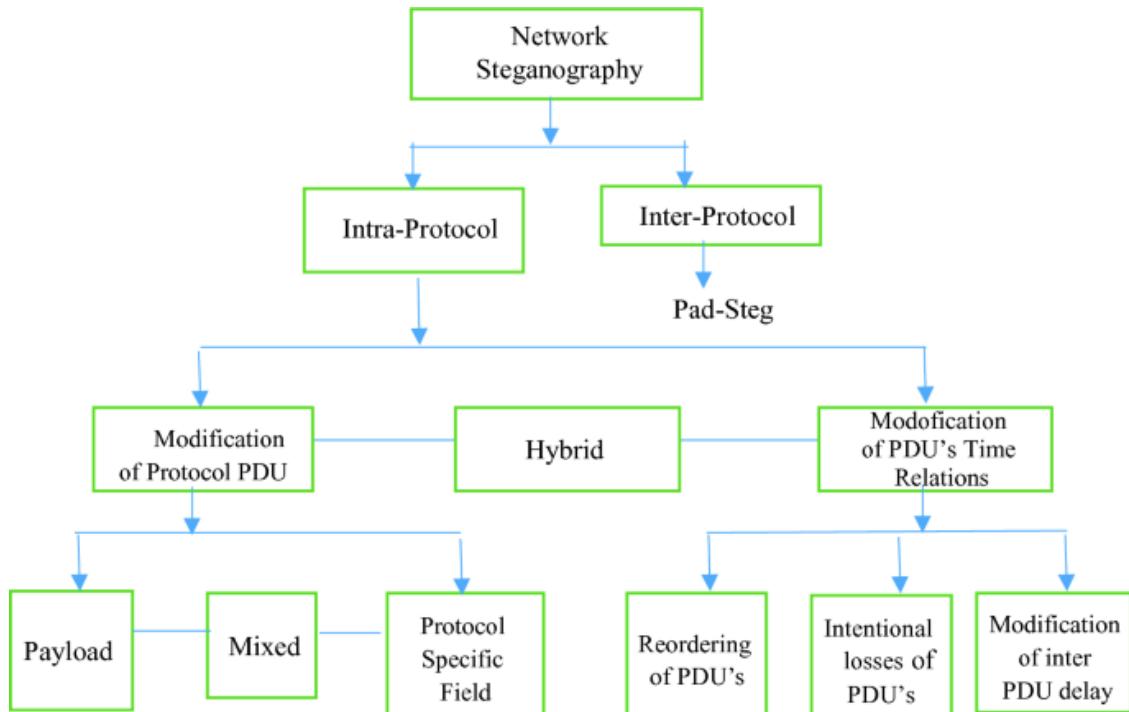
Trích rút câu:

Một hướng tiếp cận khác của phương pháp giấu tin bằng ngữ nghĩa là định nghĩa một hàm rút một câu thành một bit dữ liệu. Cách làm ở đây có thể là kiểm tra tính chẵn lẻ của mã ASCII trong tất cả các ký tự ở trong câu. Trong một ứng dụng thực tế của phương pháp này, bộ xử lý văn bản đã được chỉnh sửa như sau:

- Đầu tiên, bộ xử lý văn bản nhận thông tin cần giấu và chứa chúng trong một chuỗi bit.
- Bộ xử lý văn bản nhận những ký tự của văn bản thường được nhập vào theo chu kỳ, đưa vào hàm xử lý và so sánh kết quả của nó với bit thông tin được ẩn giấu tiếp theo.
- Nếu tính chẵn lẻ của tất cả các mã ASCII của câu hiện tại có một bit lệch với bit của chuỗi bit cần giấu thông tin, bộ xử lý văn bản sẽ từ chối không cho gõ thêm các ký tự khác. Lựa chọn duy nhất là người dùng sẽ phải viết lại câu.

2.2.2. Giấu Tin trong Mạng (Network Steganography)

Giấu tin trong mạng (Network Steganography) là một lĩnh vực của steganography, nơi thông tin được ẩn một cách kín đáo trong giao tiếp mạng. Mục đích chính của Network Steganography là để truyền tải thông tin bí mật thông qua các mạng máy tính mà không làm thay đổi hành vi giao tiếp thông thường của mạng đó.



Hình 2.6: Sơ đồ phân loại các phương pháp Giấu Tin trong Mạng

- Intra-Protocol Steganography:

- o Đây là phương pháp giấu tin bằng cách sử dụng một giao thức mạng duy nhất.

Các kỹ thuật cụ thể bao gồm:

- Modification of Protocol PDU (Protocol Data Unit): Thay đổi các đơn vị dữ liệu của giao thức để chứa thông tin bí mật.
 - Payload: Dữ liệu bí mật được ẩn trong phần payload của gói tin.
 - Mixed: Kết hợp nhiều phương thức khác nhau để ẩn thông tin trong cùng một giao thức.
 - Protocol Specific Field: Sử dụng các trường đặc thù của giao thức để chứa thông tin bí mật.

- Inter-Protocol Steganography:
 - o Pad-Steg: Một kỹ thuật nơi thông tin bí mật được chèn vào các gói tin đệm hoặc padding của các gói tin giao thức khác nhau.
- Hybrid:
 - o Kết hợp cả Intra- và Inter-Protocol Steganography để tạo ra một phương pháp giấu tin phức tạp hơn.
- Modification of PDU's Time Relations:
 - o Reordering of PDU's: Thay đổi thứ tự của các đơn vị dữ liệu giao thức để truyền thông tin bí mật.
 - o Intentional losses of PDU's: Tạo ra mất mát có ý của các đơn vị dữ liệu giao thức để mã hóa thông tin bí mật.
 - o Modification of inter PDU delay: Thay đổi độ trễ giữa các đơn vị dữ liệu giao thức để mã hóa thông tin.

Trong lịch sử, hầu hết các phương pháp phát hiện mật mã mạng đều là một phần của nghiên cứu về các kỹ thuật mật mã mới. Trong những năm gần đây, đã xuất hiện các phương pháp phát hiện mới không phải là biện pháp đối phó với một kỹ thuật giấu tin cụ thể nhưng cung cấp một góc nhìn rộng hơn. Dựa trên các nguồn dữ liệu gần đây, ta có thể phân biệt hai loại chính đối với các phương pháp phát hiện kỹ thuật giấu tin mạng: kỹ thuật cụ thể và kỹ thuật chung

2.2.3. Giấu Tin trong Hình Ảnh (Image Steganography)

Giấu tin trong ảnh là việc thực hiện giấu thông tin với môi trường chứa là các file ảnh. Là kỹ thuật ẩn thông tin bí mật trong một hình ảnh sao cho sự hiện diện của thông tin đó không thể dễ dàng bị phát hiện bởi người quan sát. Hiện nay, giấu tin trong ảnh chiếm tỉ lệ lớn trong các ứng dụng giấu tin trong dữ liệu đa phương tiện bởi vì lượng thông tin được trao đổi bằng hình ảnh là rất lớn. Giấu tin trong ảnh có nhiều ứng dụng trong thực tế, ví dụ như trong việc xác định bản quyền sở hữu, chống xâm nhập thông tin hay truyền dữ liệu một cách an toàn,...

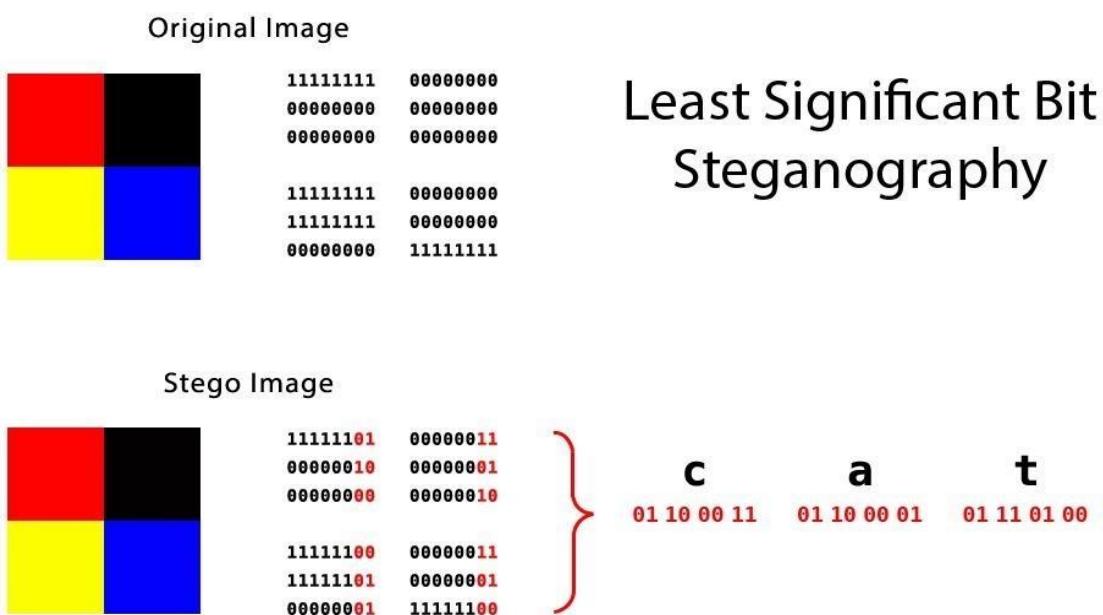
Một ví dụ thực tế cho việc giấu tin trong ảnh đã được Leonardo da Vinci sử dụng để nhúng các thông điệp bí mật vào những bức tranh của mình



Hình 2.7: Tác phẩm Mona Lisa ẩn chứa thông điệp bí mật của tác giả

2.2.3.1. LSB (Least Significant Bit) Steganography

Phương pháp này bao gồm việc thay thế bit ít quan trọng nhất (LSB) trong một hoặc nhiều byte của hình ảnh bằng bit của thông tin bí mật. Vì LSB thay đổi chỉ ảnh hưởng nhỏ đến màu sắc, nên sự thay đổi này thường không đáng kể và khó phát hiện bằng mắt thường.



Hình 2.8: Sơ đồ này hiển thị hai hình ảnh 4 pixel ở cả giá trị màu và nhị phân.

Mỗi khối nhị phân đại diện cho giá trị của pixel tương ứng.

Cách tiếp cận này dựa trên thực tế là thay đổi một chút trong giá trị của một pixel (ví dụ, từ 11011101 thành 11011100) thường không đủ để làm thay đổi màu sắc một cách có thể nhận biết được bằng mắt thường. Điều này cho phép thông tin bí mật được ẩn đi mà không làm thay đổi đáng kể đến nội dung hay chất lượng của hình ảnh.

MÃ VẠCH

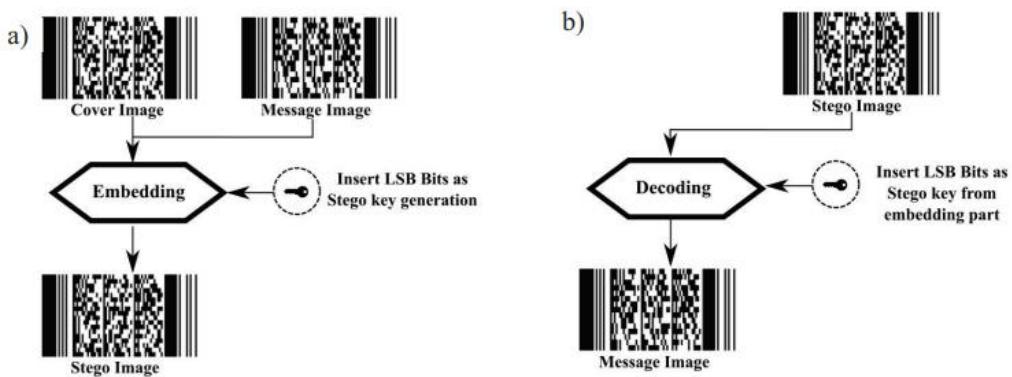
Một ví dụ thực tiễn trong đời sống của phương pháp này chính là dùng trong mã vạch. Mã vạch phổ biến trong mọi lĩnh vực như công nghiệp, thương mại, vận tải và tài chính vì đặc điểm là nhận dạng tự động hoàn thiện (ID tự động), độ chính xác và chức năng vượt trội.

Có một số loại mã vạch: 1 chiều (1D), 2 chiều (2D) và 3 chiều (3D). Mã vạch 1D thể hiện thông tin theo trực ngang, chỉ hiển thị hình và bảng chữ cái và có dung lượng dữ liệu hạn chế. Mã vạch 2D hiển thị thông tin theo cả trực ngang và trực dọc, dung lượng dữ liệu lớn hơn 100 lần so với mã vạch 1D. Mã vạch 3D in theo trực ngang và dọc, có độ sâu và độ dày. Mã vạch 3D này là duy nhất và khó sao chép.



Hình 2.9: Mã vạch 1D và 2D

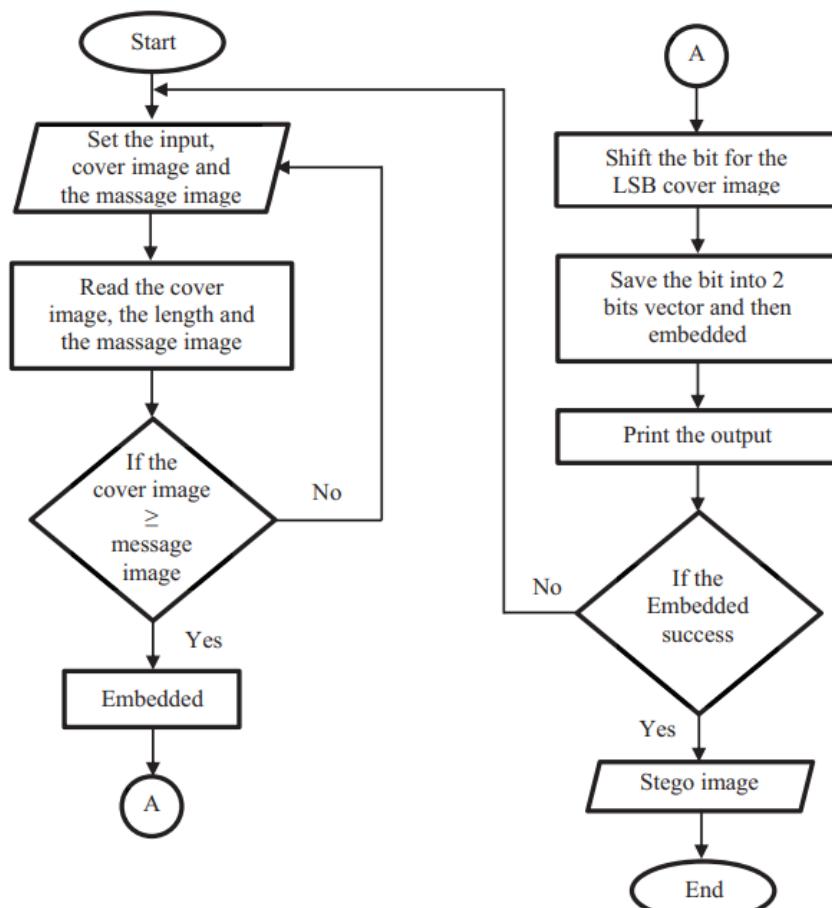
Mã vạch mã hóa dữ liệu mà chỉ có máy quét quang học hoặc bộ giải mã mới có thể đọc được. Mã vạch giúp người dùng thực hiện theo dõi điện tử hàng tiêu dùng tại cửa hàng tạp hóa, thiết bị đầu cuối thanh toán, công cụ quản lý tài liệu, quản lý hàng tồn kho, bán vé, gắn thẻ di động, thẻ lên máy bay di động và giảm lỗi của con người. Trong ngành hàng không, mã vạch này được in trên thẻ lên máy bay để nhúng dữ liệu hành khách như số điện thoại, số hành khách thường xuyên, chuyến bay sắp tới, khách sạn, địa điểm hồ sơ hành khách và nhiều thông tin khác. Mã vạch này còn giúp hàng không tự động phát hiện danh sách cấm bay.



Hình 2.10: Quá trình mã hóa (a) và giải mã mã vạch (b)

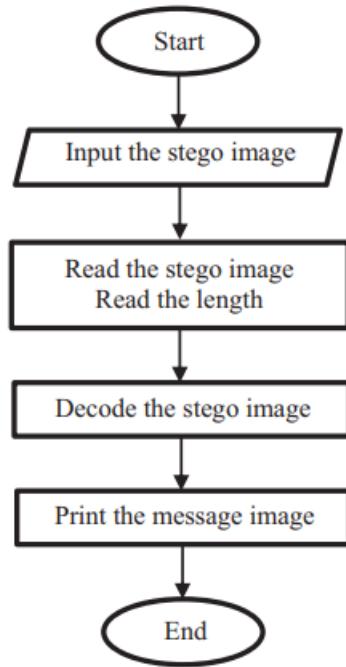
PHƯƠNG PHÁP

Trong báo cáo này, chương trình đã thực hiện mã hóa và giải mã. Đầu tiên, chương trình đặt bit LSB từ 1 đến 4. Bit này không được cao hơn 4 vì hình ảnh văn bản sẽ đóng ảnh bìa. Sơ đồ chương trình để nhúng mã vạch (hình ảnh tin nhắn)



Hình 2.11: Sơ đồ chương trình nhúng hình ảnh tin nhắn

Chương trình sử dụng ảnh bí mật (mã vạch thẻ lên máy bay) định dạng BMP 1183 x 1183 pixel, dung lượng 5467 kb, ảnh bìa định dạng BMP 1183 x 1183 pixel, dung lượng 5467 kb. Có hai cuộc phẫu thuật; mã hóa và giải mã hình ảnh mã vạch thẻ lên máy bay. Hình ảnh này được xử lý theo bit có trọng số thấp nhất là 1, 2, 3 và 4. Chương trình đã giải mã hình ảnh mã vạch thẻ lên máy bay và thay đổi nó thành hình ảnh stego. Các chương trình cũng cung cấp quá trình mã hóa. Vì vậy, ảnh stego có thể được đổi lại thành ảnh bìa.



Hình 2.12: Sơ đồ giải mã ảnh stego

Chất lượng của ảnh stego sau đó được xác nhận bằng cách so sánh nó với ảnh bìa . Việc so sánh được thực hiện bằng cách tính Sai số bình phương trung bình (MSE), Sai số bình phương trung bình tỷ lệ (RMSE) và Tỷ lệ tín hiệu trên nhiễu cực đại (PSNR). MSE là tỷ lệ lỗi giữa ảnh bìa và ảnh stego. MSE có thể được ký hiệu như sau:

$$MSE = \frac{1}{mn} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [f(i,j) - g(i,j)]^2,$$

Trong đó M là số hàng ảnh bìa, N là số cột ảnh bìa, f(i,j) là pixel của ảnh bìa, g(i,j) là giá trị pixel của ảnh. Giá trị MSE cao hơn cho thấy sự khác biệt giữa ảnh bìa và ảnh stego. RMSE là lỗi giữa độ lệch giữa hai hình ảnh đó. RMSE có thể được viết như sau:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{mn} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [f(i,j) - g(i,j)]^2},$$

Khi giá trị của MSE và RMSE gần bằng 0 thì ảnh bìa và ảnh stego được đóng giống nhau. PSNR là tỷ lệ giữa pixel tối đa và nhiễu tín hình bằng đơn vị decibel (dB). Nó còn được tính để so sánh độ giống nhau của ảnh bìa và ảnh stego. Khi PSNR thấp hơn 30 dB thì hình ảnh không giống nhau. Chất lượng của ảnh stego sau đó được xác nhận bằng cách so sánh nó với ảnh bìa. Việc so sánh được thực hiện bằng cách tính Sai số

bình phương trung bình (MSE), Sai số bình phương trung bình tỷ lệ (RMSE) và Tỷ lệ tín hiệu trên nhiễu cực đại (PSNR). MSE là tỷ lệ lỗi giữa ảnh bìa và ảnh stego. MSE có thể được ký hiệu như sau PSNR được ký hiệu trong phương trình sau:

$$PSNR = 10 \log_{10} \left[\frac{I^2}{MSE} \right].$$

Trong đó MSE là căn bậc hai trung bình và I là số pixel tối đa. Trong nghiên cứu này, pixel tối đa là 255

KẾT LUẬN

Chương trình cung cấp hai quy trình; hình ảnh mã vạch được mã hóa và giải mã như hình dưới Chương trình giải mã hình ảnh mã vạch thẻ lên máy bay. Do đó, kẻ trộm dữ liệu không thể truy cập vào dữ liệu của hành khách vì mã vạch đã thay đổi thành hình ảnh stego mà ứng dụng điện thoại di động cũng như đầu đọc mã vạch trực tuyến không thể quét được. Chương trình cũng cung cấp quá trình mã hóa. Do đó, hình ảnh stego có thể được thay đổi lại thành hình ảnh mã vạch.



Hình 2.13: Chương trình mã hóa ảnh cover và đếm MSE, RMSE, PSNR



Hình 2.14: Hình ảnh steganography được giải mã

Kết quả của quá trình giải mã được xử lý với 1, 2, 3 và 4 bit có trọng số nhỏ nhất được so sánh, như trong bảng dưới. Dữ liệu trong bảng cũng được hiển thị giá trị MSE, RMSE và PSNR của mỗi bit. Kết quả cho thấy 1 bit ít quan trọng nhất cho kết quả tốt nhất. MSE và RMSE là 0,192314 và 0,438537 trong khi đóng ở mức 0. PSNR là 55,3247 dB, nghĩa là cao hơn 30 dB. Những dữ liệu này chứng minh rằng ảnh bìa và ảnh stego giống nhau và mắt người không thể nhận thấy được. Nhịp thứ 4 kém quan trọng nhất là nhịp yếu vì MSE và RMSE lần lượt là 6,89151 và 2,62517; những giá trị này rất cao. PSNR là 39,7817dB, cao hơn 30 dB. Từ dữ liệu được hiển thị trong bảng, chúng ta có thể kết luận rằng bit ít quan trọng nhất mang lại hình ảnh stego tốt nhất.

The parameter	The cover image	The Stego Image	MSE	RMSE	PSNR
1 bit			0.192314	0.438537	55.3247
2 bits			0.53323	0.730226	50.8957
3 bits			1.90664	1.38081	45.3621
4 bits			6.89151	2.62517	39.7817

Hình 2.15: Bảng kết quả so sánh ảnh cover và ảnh stego

Trong bài này, chương trình bảo vệ dữ liệu nhúng trong mã vạch được thực hiện với kỹ thuật steganography 1, 2, 3 và 4 bit ít quan trọng nhất. Ảnh bìa và ảnh stego được so sánh và bit có ý nghĩa nhỏ nhất 1 hiển thị hình ảnh stego tốt nhất trong khi MSE và RMSE đóng về 0 và PSNR cao hơn 30 dB. Vì vậy, chúng giống nhau và mắt người không thể nhận ra sự khác biệt giữa ảnh bìa này và ảnh stego.

Kết quả đã chứng minh rằng chương trình sử dụng phương pháp steganography 1 bit có ý nghĩa nhỏ nhất là phù hợp để ẩn ảnh bìa.

2.2.3.2. DCT (Discrete Cosine Transform) Steganography

Là một phương pháp ẩn tin trong đó thông tin được giấu trong miền tần số của hình ảnh thay vì miền không gian như làm trong LSB Steganography. Discrete Cosine Transform (DCT) là một kỹ thuật chuyển đổi một hình ảnh từ miền không gian sang miền tần số. Kỹ thuật này là nền tảng của chuẩn nén JPEG, vì nó giúp loại bỏ thông tin mà mắt người không nhạy cảm, từ đó giảm kích thước của tệp ảnh mà không giảm chất lượng nhìn thấy.

Trong DCT Steganography, thông tin ẩn được chèn vào trong các hệ số DCT của hình ảnh. Vì con người ít nhạy cảm với sự thay đổi trong các tần số cao, thường các hệ số DCT tương ứng với tần số cao sẽ được chọn để chứa thông tin ẩn.

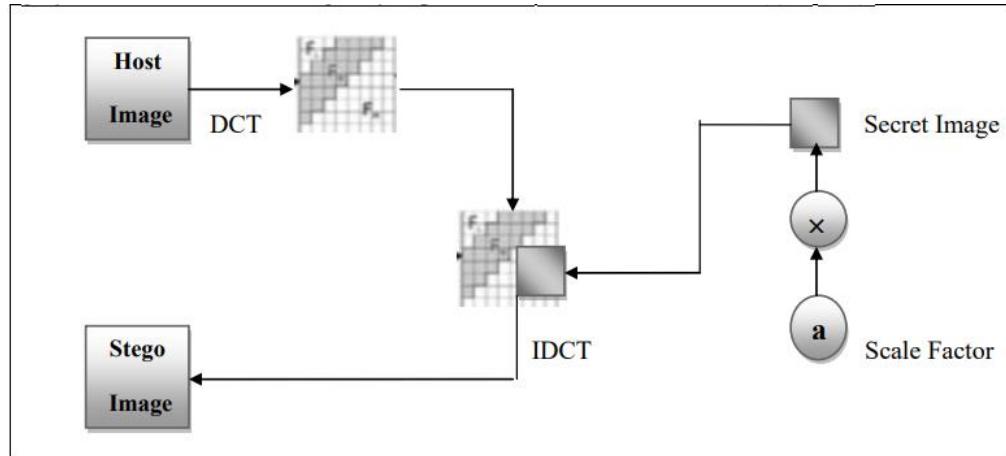
Biến đổi hình ảnh rất quan trọng trong xử lý kỹ thuật số, chúng cho phép thực hiện ít hơn với nhiều hơn. Ví dụ: Biến đổi Fourier có thể được sử dụng để tính toán hiệu quả các phép biến đổi hình ảnh hoặc Biến đổi Cosine rời rạc có thể được sử dụng để giảm đáng kể không gian bị chiếm giữ bởi hình ảnh mà không làm giảm chất lượng đáng kể. DCT biến đổi tín hiệu hoặc hình ảnh từ miền không gian sang miền tần số. Nó tách hình ảnh thành các phần (hoặc các dải phổ con) có tầm quan trọng khác nhau (liên quan đến chất lượng hình ảnh của hình ảnh). Nó có thể tách Hình ảnh thành các thành phần Tần số Cao, Trung bình và Tần số Thấp .

Ngoài ra, còn có thuật toán mở rộng hơn “The Two-Dimensional (2D-DCT)” nhưng vẫn dựa trên các nguyên lý của DCT. Trong báo cáo này sẽ không đề xuất tới phương pháp mở rộng này mà sẽ chỉ đi sâu vào chi tiết của DTC 1-D (viết tắt là DTC).

Phương pháp

Lược đồ Steganography phải cực kỳ an toàn và không làm giảm chất lượng hình ảnh của Ánh bìa khi che giấu ảnh bí mật. Quá trình che giấu tổng thể của sơ đồ đề xuất của chúng tôi được hiển thị trong Hình dưới. Thuật toán được đề xuất phụ thuộc vào việc chuẩn hóa hình ảnh bí mật bằng cách nhân nó với hệ số tỷ lệ (a) sau đó nhúng hình

anh bí mật vào dải tần số cao của hình ảnh chủ sau khi áp dụng Biến đổi Cosine rời rạc. DCT nghịch đảo được áp dụng để có được hình ảnh stego. Thuật toán để xuất ẩn ba hình ảnh bí mật có thang màu xám thành một hình ảnh màu. Bằng thử nghiệm, giá trị tốt nhất của (a) là (0,8).



Hình 2.16: Sơ đồ phương pháp nhúng giấu liệu

Thuật toán nhúng:

Đầu vào: Ba hình ảnh có thang màu xám kích thước $\frac{N}{2} \times \frac{N}{2}$ (ảnh một màu) có kích thước ($N \times N$).

Đầu ra: Ảnh Stego

Đối với mỗi dải (R, G, B) của ảnh cover:

- Bước 1: Áp dụng phép biến đổi Cosine rời rạc
- Bước 2: Chuẩn hóa ảnh bí mật bằng cách nhân nó với hệ số (a)
- Bước 3: Thay thế các hệ số của băng ảnh bìa trong phần tư thứ tư nơi băng tần cao cùng với hệ số của ảnh mật mã chuẩn hóa theo công thức:

$$C(m, n) = a * s(k, j)$$

Với: $m, n = \frac{N}{2} + 1 \text{ to } N$

$$k, j = 1 \text{ to } \frac{N}{2},$$

C là ảnh cover

S là ảnh bí mật

- Bước 4: Xây dựng lại hình ảnh stego bằng cách áp dụng Biến đổi rời rạc nghịch đảo

Thuật toán trích xuất (ngược lại của nhúng):

Đầu vào: Ảnh Stego

Đầu ra: Ba ảnh bí mật

Đối với mỗi dải (R, G, B) của ảnh màu Stego:

- Bước 1: Dùng biến đổi Cosine rời rạc (Discrete Cosine Transform)
- Bước 2: Trích xuất hình ảnh bí mật được nhúng bằng cách áp dụng:

$$S(k, j) = \frac{Stego(m, n)}{a}$$

Với: $m = \frac{N}{2} + 1$ $k, j = 1 \rightarrow \frac{N}{2}$

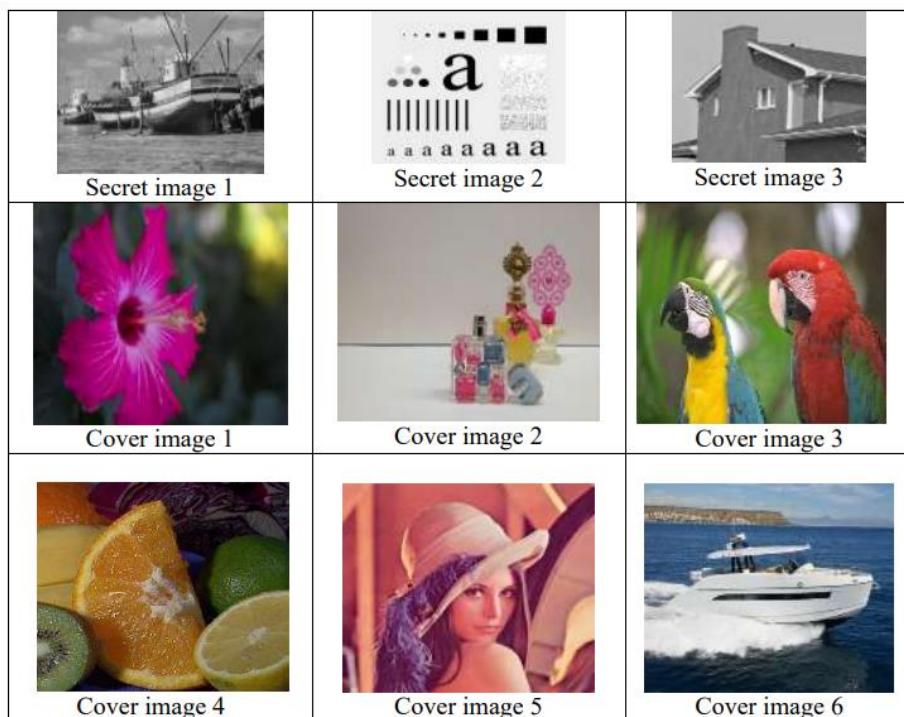
$n = \frac{N}{2} + 1 \rightarrow N$ Stego là ảnh stego

S là ảnh bí mật

Thử Nghiệm

Thuật toán đề xuất được triển khai và thử nghiệm trên một số hình ảnh tiêu chuẩn.

Ảnh bìa có tỷ lệ màu (256×256) và ảnh bí mật có kích thước (128×128) (tỷ lệ xám). Hình dưới hiển thị ở hàng đầu tiên các ảnh thử nghiệm bí mật, hàng thứ hai và thứ ba dành cho ảnh bìa thử nghiệm.



Hình 2.17: Ảnh Cover và ảnh bí mật

Ba hình ảnh bí mật tương tự được nhúng vào ảnh bìa mỗi lần . Phép đo Tỷ lệ Tín hiệu trên Nhiễu (PSNR) và Độ đo Tương quan được sử dụng để đánh giá chất lượng của ảnh bìa so với ảnh bìa gốc. Khi hình ảnh stego giống với hình ảnh bìa gốc về mặt cảm nhận thì mối tương quan bằng một . Các số liệu này được định nghĩa như sau:

$$PSNR = 10 \log 10 \left(\frac{255^2}{MSE} \right)$$

$$MSE = \left(\frac{1}{N} \right)^2 \sum \sum (x_{ij} - x'_{ij})^2$$

Trong đó x_{ij} biểu thị giá trị pixel gốc, x'_{ij} biểu thị các giá trị pixel đã sửa đổi và N là kích thước của hình ảnh.

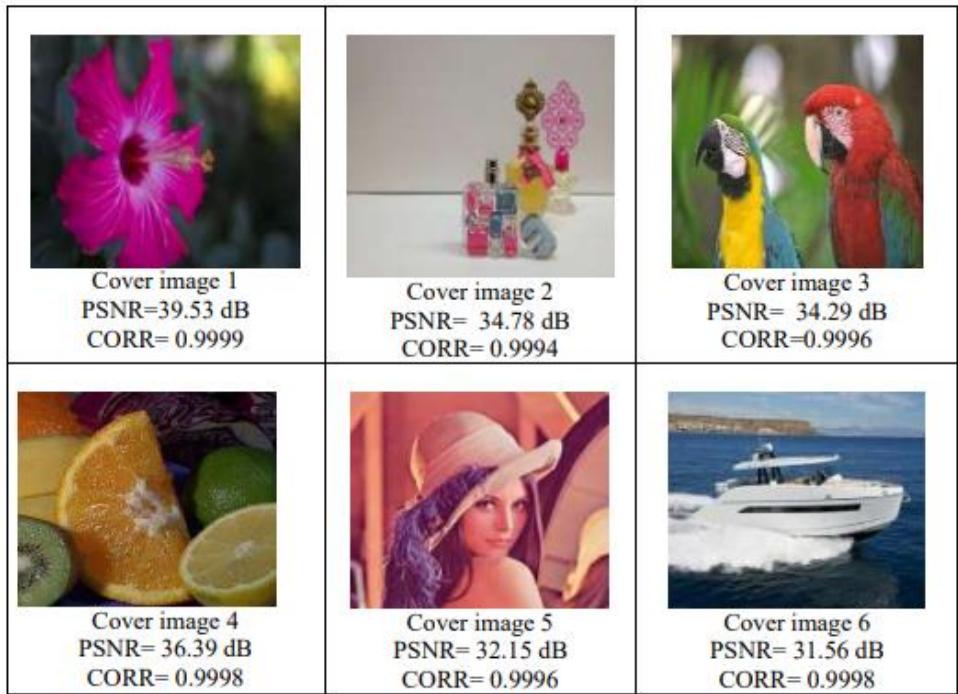
Corr

$$= \frac{\sum_{i=1}^M \sum_{j=1}^N (x(i,j) - \bar{x})(y(i,j) - \bar{y})}{\sqrt{[\sum_{i=1}^M \sum_{j=1}^N (x(i,j) - \bar{x})^2][\sum_{i=1}^M \sum_{j=1}^N (y(i,j) - \bar{y})^2]}}$$

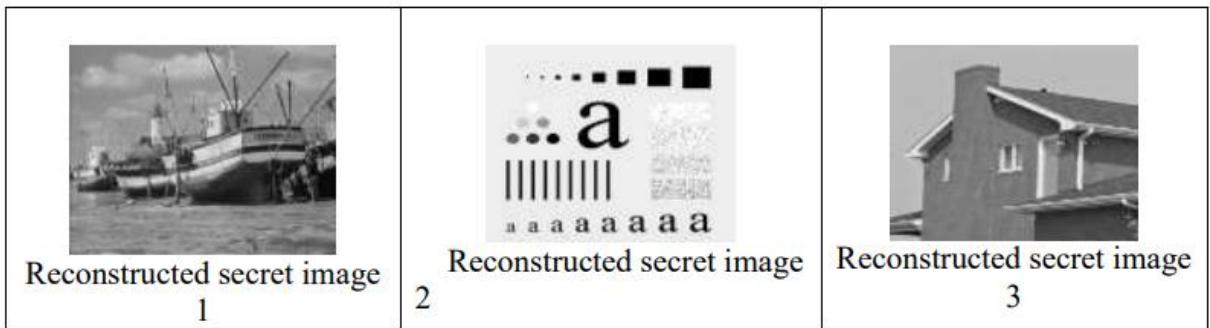
Trong đó MN: chiều cao và chiều rộng của hai ảnh.

$x(i,j)$ là ảnh gốc $y(i,j)$ là ảnh đã được chỉnh sửa
 \bar{x} và \bar{y} là giá trị ảnh gốc và ảnh đã được chỉnh sửa tương ứng

Hình dưới hiển thị kết quả của thuật toán đề xuất. Như được hiển thị, hình ảnh stego có chất lượng tốt.Hình dưới nữa hiển thị các hình ảnh bí mật được tái tạo giống nhau từ mỗi hình ảnh stego, chúng có chất lượng rất tốt và không bị mất thông tin



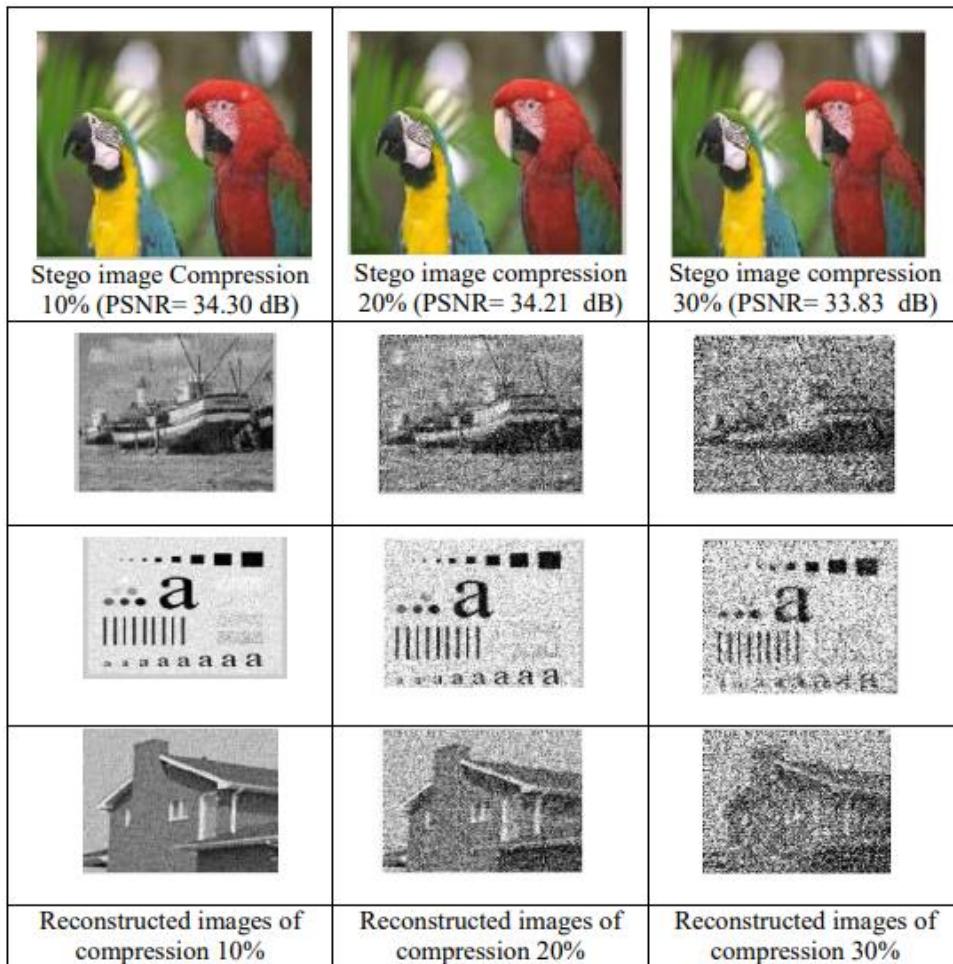
Hình 2.18: PSNR và CORR của ảnh Stego



Hình 2.19: Ảnh bí mật trích xuất ra từ ảnh Stego

Các ảnh Stego được nén bằng hệ thống nén JPEG để nghiên cứu ảnh hưởng của việc nén lên hệ thống được đề xuất. Hình dưới hiển thị hình ảnh Stego (Vết) ở các mức nén JPEG khác nhau và các hình ảnh bí mật được tái tạo cho tỷ lệ nén 10%, 20%, 30%. Tỷ lệ nén được sử dụng là .

$$CR = (\text{kích thước file nén}/\text{kích thước file không nén}) * 100 \%$$



Hình 2.20: Kết quả của Stego(Parrot) và các ảnh bí mật được tái tạo để nén JPEG

Từ kết quả nén bị mất JPEG trên ảnh stego, chúng ta có thể nhận thấy rằng hệ thống cung cấp phản hồi tốt ở tỷ lệ nén thấp trong khi nó cung cấp phản hồi rất kém ở tỷ lệ nén trung bình và cao. Lý do là vì thuật toán ẩn phụ thuộc vào việc ẩn thông tin bí mật ở tần số băng tần cao mà trước tiên nó bị ảnh hưởng bởi quá trình nén.

Kết Luận

Trong thuật toán đề xuất, sơ đồ nhúng ảnh bí mật có kích thước (1/4) của ảnh bìa được đề xuất, phụ thuộc vào việc áp dụng phép biến đổi cosine rồi rác cho ảnh bìa, nhúng ảnh bí mật vào dải tần số cao sau khi nhân nó với một hệ số tỷ lệ (a) để ngăn chặn các hiện tượng thị giác, giữ nguyên các băng tần con tần số thấp. Thuật toán được bảo mật vì hạn chế của thuật toán nhúng (nghĩa là sử dụng phép biến đổi và giá trị của hệ số tỷ lệ).

2.2.3.3. Palette-Based Steganography

Là một phương pháp steganography áp dụng cho các hình ảnh sử dụng palette màu. Trong đồ họa máy tính, một palette là một tập hợp hạn chế các màu. Hình ảnh sử dụng palette màu được gọi là hình ảnh chỉ màu (indexed color images), nơi mỗi pixel không trực tiếp chứa thông tin về màu sắc mà là chỉ số tham chiếu đến một màu trong palette.

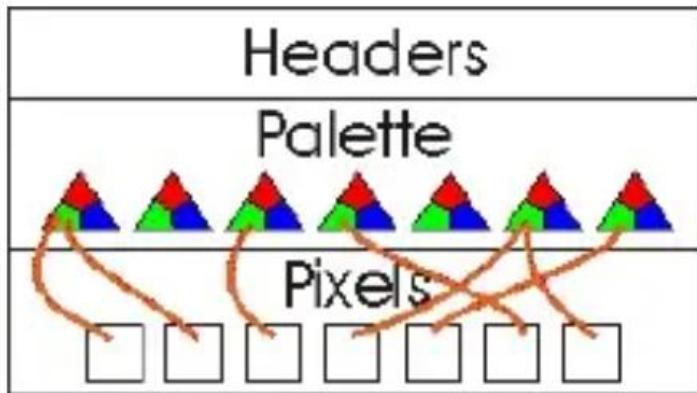
Hình ảnh màu dựa trên bảng màu hoặc được lập chỉ mục cho phép 8 bit trên mỗi pixel trở xuống để trông gần giống như 24 bit trên mỗi pixel. Kỹ thuật này xác định 256 màu được sử dụng thường xuyên nhất trong hình ảnh và tạo bảng tra cứu màu, còn được gọi là bản đồ màu hoặc bảng màu, được lưu trữ cùng với hình ảnh. Thay vì mỗi pixel trong ảnh có cả ba màu RGB (một màu đỏ 8 bit, một màu xanh lá cây 8 bit và một màu xanh lam 8 bit), mỗi pixel chứa một số 8 bit được lập chỉ mục vào bảng tra cứu 256 màu, chứa các giá trị RGB. Điều này làm giảm hình ảnh xuống kích thước nhỏ nhất và những hình ảnh này được sử dụng phổ biến nhất trên các trang Web vì chúng nhỏ và tải nhanh. Bảng màu 256 màu được ánh xạ để mang lại kết quả tốt nhất trên Internet.

Khi màn hình máy tính thời kỳ đầu thường bị giới hạn ở 256 màu, các phương pháp màu được lập chỉ mục là rất cần thiết. Ngay cả khi đó, hai bức ảnh được lập chỉ mục trên màn hình cùng lúc với cách phối màu rất khác nhau sẽ làm quá tải dung lượng màu của phần cứng và hiển thị không đúng. Ngày nay, phần cứng máy tính dễ dàng hiển thị đầy đủ màu 24 bit, nhưng hình ảnh được lập chỉ mục 8 bit vẫn được sử dụng rộng rãi vì kích thước tệp vẫn là điều quan trọng nhất và kích thước tệp nhỏ hơn là tối ưu trong giao tiếp mạng mặc dù tốc độ mạng ngày càng tăng và các vấn đề về băng thông đang giảm. Do đó, việc sử dụng hình ảnh 24-bit steganographic hiện nay dẫn đến việc truyền thông chậm hơn và việc phát triển định dạng hình ảnh 8-bit sẽ có lợi.

Sơ lược về hình ảnh dựa trên bảng màu

Báo cáo này đề cập tới loại hình ảnh dựa trên bảng màu phổ biến nhất. Đó là GIF.

Định dạng trao đổi đồ họa (GIF) là định dạng hình ảnh bitmap 8 bit cho mỗi pixel được CompuServe giới thiệu vào năm 1987 và từ đó được sử dụng rộng rãi trên World Wide Web do tính hỗ trợ rộng rãi và tính di động của nó.



Hình 2.21: 8 bit trên mỗi pixel, bảng có kích thước tùy ý (1 - 256)

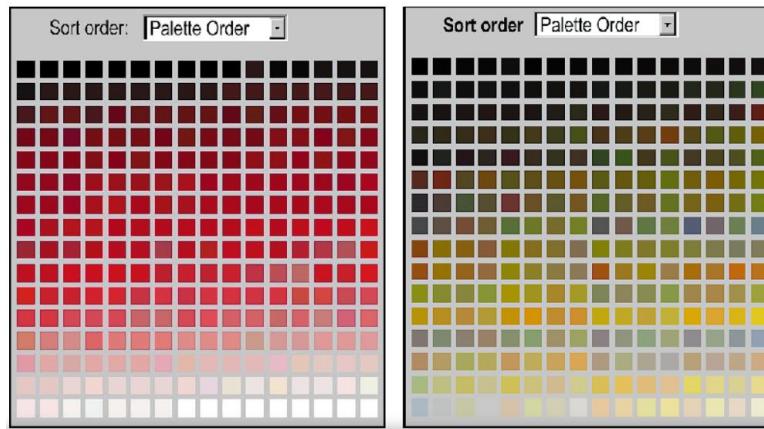
Định dạng này sử dụng bảng màu lên tới 256 màu riêng biệt từ không gian màu RGB 24 bit. Nó cũng hỗ trợ hoạt ảnh và cho phép một bảng màu riêng biệt gồm 256 màu cho mỗi khung hình. Giới hạn màu sắc khiến định dạng GIF không phù hợp để tái tạo ảnh màu và các hình ảnh khác có màu liên tục, nhưng nó rất phù hợp cho các hình ảnh đơn giản hơn như đồ họa hoặc logo có các vùng đặc color. mặc dù bất kỳ lựa chọn bảng màu nào cũng có thể có hàng triệu sắc thái, nhưng số lượng tối đa có thể được sử dụng trong một khung là 256. Chúng được lưu trữ trong một "bảng màu", có thể liên kết từng số lựa chọn bảng màu với RGB cụ thể giá trị. Giới hạn 256 màu có vẻ hợp lý vào thời điểm GIF được tạo ra vì ít người có đủ khả năng mua phần cứng để hiển thị nhiều hơn. Đồ họa đơn giản, hình vẽ, phim hoạt hình và ảnh thang màu xám thường cần ít hơn 256 màu. Ngoài ra, một trong các màu trong bảng màu có thể tùy ý bao quanh là hoàn toàn trong suốt. Một pixel trong suốt sẽ lấy màu của pixel ở cùng vị trí so với nền, điều này có thể đã được xác định bởi khung hoạt ảnh trước đó

Một định dạng khác mới hơn là PNG Viết tắt của Portable Network Graphics, chuẩn đồ họa thứ ba được Web hỗ trợ (mặc dù không được tất cả các trình duyệt hỗ trợ). PNG được phát triển như một giải pháp không có bảng sáng chế cho định dạng GIF nhưng cũng là một cải tiến về kỹ thuật GIF. Hình ảnh trong tệp PNG không mất dữ liệu có thể được nén nhiều hơn 5%-25% so với tệp GIF của cùng một hình ảnh. PNG xây

dựng dựa trên ý tưởng về độ trong suốt của ảnh GIF và cho phép kiểm soát mức độ trong suốt, được gọi là độ mờ. Việc lưu, khôi phục và lưu lại hình ảnh PNG sẽ không làm giảm chất lượng của nó. PNG không hỗ trợ hoạt ảnh như GIF

Bảng màu và bô cục hình ảnh

Bảng màu và bô cục của hình ảnh cũng góp phần vào việc công cụ stego thực hiện công việc của nó tốt như thế nào. Hình ảnh có dải màu dần dần hoặc ở thang độ xám là tốt nhất cho tốc ký vì dễ chèn các “lỗi” nhỏ vào hơn. Các thay đổi cũng xuất hiện dần dần và do đó ít có khả năng được phát hiện hơn.



Hình 2.22: Chuyển màu bảng màu

Quan sát các bảng màu khác nhau và bảng màu bên trái thay đổi dần dần như thế nào và phù hợp với ảnh bìa hơn bảng màu bên phải

Steganography sử dụng hình ảnh dựa trên bảng màu

Phần lớn hình ảnh trên Internet có sẵn ở các định dạng dựa trên bảng màu, chẳng hạn như GIF hoặc PNG. Có hai cách tiếp cận để ẩn thông điệp trong hình ảnh dựa trên bảng màu:

- Nhúng tin nhắn vào bảng màu (palette).
- Nhúng vào dữ liệu hình ảnh.

Ưu điểm của phương pháp đầu tiên là có thể dễ dàng thiết kế một phương pháp an toàn hơn với một số giả định về đặc tính nhiều của nguồn hình ảnh (máy quét, camera

CCD, v.v.). Nhược điểm rõ ràng là dung lượng không phụ thuộc vào hình ảnh và bị giới hạn bởi kích thước bảng màu.

Các phương thức từ nhóm thứ hai có dung lượng cao hơn, nhưng nhìn chung khó thiết kế một sơ đồ an toàn hơn

Để chứng minh tính bảo mật của một lược đồ nhúng, chúng ta cần hiểu rõ các thuật toán để tạo ảnh dựa trên bảng màu. Hầu hết tất cả các thuật toán đều bao gồm hai bước: lượng hóa màu (còn gọi là lượng hóa vectơ) và dithering.

- Lượng hóa màu chọn bảng màu của ảnh bằng cách cắt bớt tất cả các màu của ảnh gốc, 24 bit thành một số lượng màu hữu hạn (256 cho ảnh GIF, 216 cho GIF phiên bản Netscape, 2 cho ảnh đen trắng, v.v.).
- Dithering được sử dụng để tăng độ sâu màu rõ ràng. Nó sử dụng các đặc tính tích hợp của hệ thống thị giác của con người và tạo ra ảo ảnh về các màu bổ sung bằng cách đổi độ phân giải không gian lấy độ sâu màu. Kết quả tốt nhất thu được bằng cách sử dụng các thuật toán dithering dựa trên khuếch tán lõi.

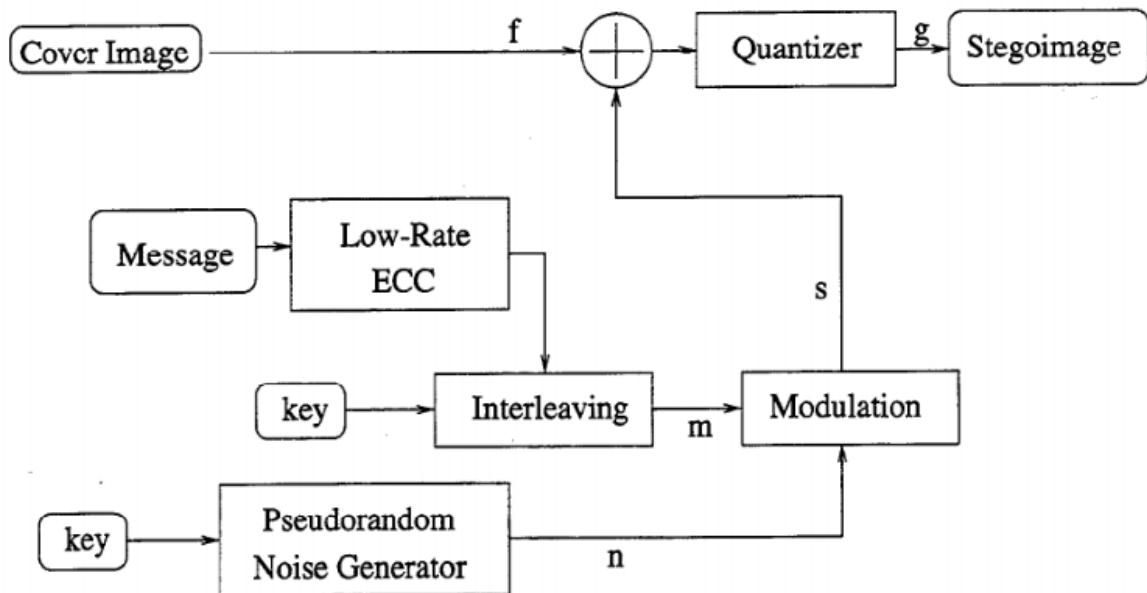
Có một số thuật toán để lượng hóa màu. Hai thuật toán thường được sử dụng nhất dựa trên việc chia tách lặp lại khối lập phương màu ba chiều thành hai hộp có số lượng màu xấp xỉ bằng nhau. Nửa có kích thước lớn nhất được chọn cho lần lặp tiếp theo cho đến khi đạt được số hộp (màu) mong muốn. Sau đó, trọng tâm của mỗi hộp được làm tròn thành các màu nguyên đại diện cho màu sắc của bảng màu. Nếu kích thước lớn nhất được thay thế bằng độ lệch chuẩn lớn nhất, thì sẽ thu được một thuật toán khác tốt hơn một chút.

Thuật toán phổ biến nhất để dithering dựa trên khuếch tán lõi. Ảnh được quét theo một cách thức đều đặn nào đó, chẳng hạn như theo hàng, cột hoặc đường chéo. Khi màu của một pixel được làm tròn về màu gần nhất trong bảng màu, một lõi sẽ được tạo ra (lõi này âm nếu làm tròn giảm giá trị pixel và dương nếu ngược lại). Lõi được nhân với các trọng số và thêm vào các pixel xung quanh chưa được xử lý. Bằng cách này, lõi làm tròn được lan truyền sang các pixel lân cận và thu được một hình ảnh đẹp mắt, không có hiện tượng tạo đường viền.

2.2.3.4. Spread Spectrum Steganography

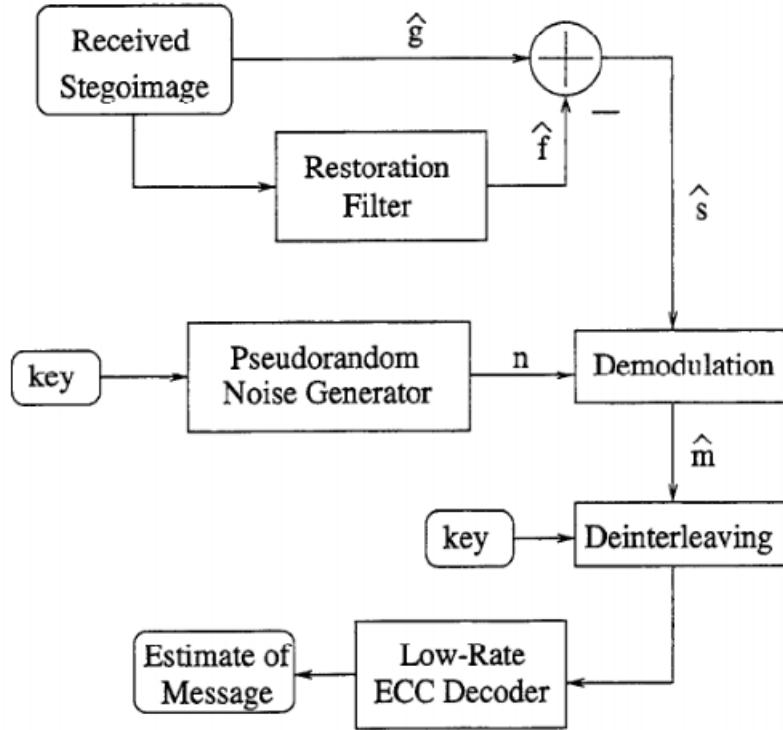
Spread Spectrum Steganography hoạt động bằng cách lưu trữ thông điệp dưới dạng nhiễu Gaussian trong một hình ảnh (Marvel, Bonelet và Retter 1998, Marvel et al. 1999). Ở mức công suất nhiễu thấp, mắt người không thể phát hiện được sự suy giảm hình ảnh, trong khi ở mức công suất cao hơn, nhiễu xuất hiện dưới dạng đốm hoặc “tuyết”. Quá trình này bao gồm các bước chính sau:

1. Tạo tin nhắn được mã hóa bằng cách thêm dữ phòng thông qua mã sửa lỗi.
2. Thêm phần đệm để làm cho thông báo được mã hóa có cùng kích thước với hình ảnh.
3. Xen kẽ vào tin nhắn được mã hóa
4. Tạo chuỗi nhiễu giả ngẫu nhiên, n
5. Sử dụng thông điệp được mã hóa m để điều chỉnh chuỗi tạo ra nhiễu, s .
6. Kết hợp nhiễu với ảnh gốc, f .



Hình 2.23: Sơ đồ mô tả quá trình nhúng (quá trình mã hóa)

Ngược lại với quá trình trên, quá trình dưới thể hiện quá trình trích xuất, lưu ý rằng không cần phải có hình ảnh gốc để khôi phục tin nhắn ẩn. Một bộ lọc được sử dụng để trích xuất nhiễu từ ảnh stegoimage, dẫn đến hình ảnh gần đúng với ảnh gốc. Bộ lọc này hoạt động càng tốt thì thông báo được trích xuất càng ít lỗi.



Hình 2.24: Sơ đồ quá trình trích xuất (quá trình giải mã)

Quá trình ngược lại, trích xuất và khôi phục tin nhắn gốc, tất nhiên là rất giống nhau:

1. Lọc ảnh stegoimage, g , để có được giá trị gần đúng của ảnh gốc, f .
2. Trừ giá trị gần đúng của ảnh gốc khỏi ảnh stego để có được ước tính về độ nhiễu, s , được thêm vào bởi trình nhúng.
3. Tạo ra chuỗi nhiễu giả ngẫu nhiên tương tự, n .
4. Giải điều chế bằng cách so sánh nhiễu đã trích xuất với nhiễu được tái tạo.
5. Khử xen kẽ ước tính của thông báo được mã hóa, m và loại bỏ phần đệm.
6. Sử dụng bộ giải mã sửa lỗi để sửa tin nhắn khi cần thiết.

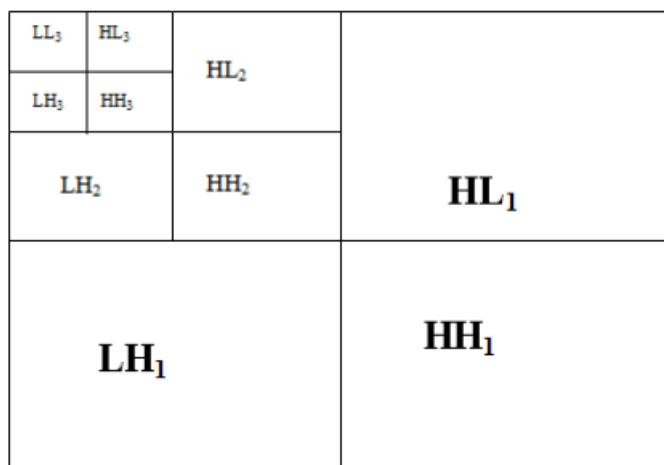
Thông tin bí mật được phân tán trên toàn bộ hình ảnh theo một cách mà nó trở nên rất khó phát hiện. Điều này tương tự như kỹ thuật phổ lan dùng trong truyền thông không dây.

2.2.3.5. DWT (Discrete Wavelet Transform) Steganography

Tương tự như DCT, DWT là một phép biến đổi khác có thể sử dụng trong steganography. Phương pháp này phân tích hình ảnh thành các thành phần tần số khác nhau và sau đó ẩn thông tin trong các coefficients của wavelet.

Biến đổi wavelet mô tả quá trình phân rã đa độ phân giải về mặt mở rộng hình ảnh thành một tập hợp các hàm cơ sở wavelet. Phép biến đổi Wavelet rời rạc có đặc tính định vị tần số không gian tuyệt vời của riêng nó. DWT chia tín hiệu thành các phần tần số cao và thấp. Phần tần số cao chứa thông tin về các thành phần biên, trong khi phần tần số thấp lại được chia thành các phần tần số cao và tần số thấp.

Các thành phần tần số cao thường được sử dụng cho steganography vì con người ít nhạy cảm với sự thay đổi ở các cạnh. Trong các ứng dụng hai chiều, chúng ta thực hiện DWT theo hướng đọc trước, tiếp theo là DWT theo hướng ngang. Như chúng ta có thể thấy trong hình dưới, sau mức phân giải đầu tiên, có bốn dài phụ: LL1, LH1, HL1 và HH1. Cho mỗi mức phân giải tiếp theo, dài phụ LL của mức trước đó được sử dụng làm đầu vào. Để thực hiện phân giải mức thứ hai, DWT được áp dụng vào dài LL1, phân giải dài LL1 thành bốn dài phụ: LL2, LH2, HL2 và HH2.

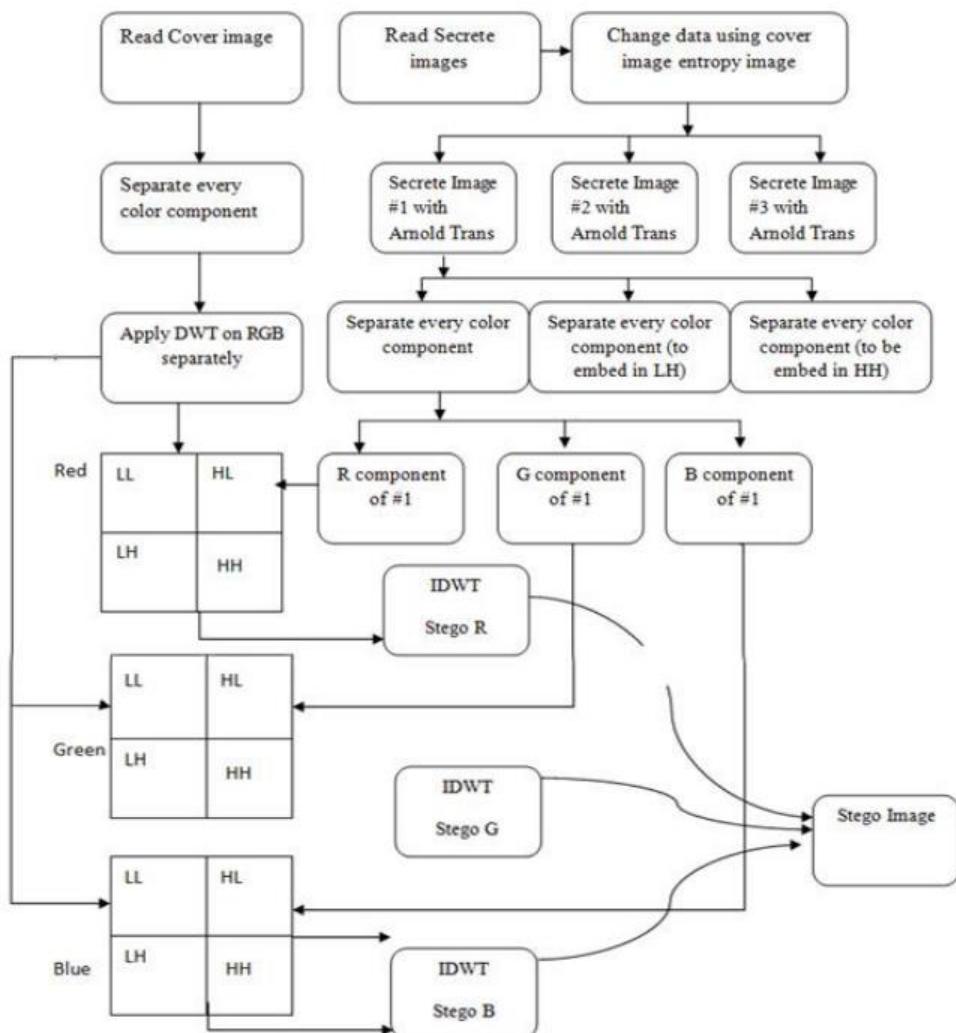


Hình 2.25: Phân giải ba pha sử dụng DWT

PSNR, dung lượng và tương quan là những khía cạnh quan trọng trong steganography. Cụ thể hơn, PSNR được coi là cao, nhưng nó phụ thuộc vào ứng dụng cụ thể. PSNR tỷ lệ nghịch với dung lượng và trực tiếp tỷ lệ với tương quan. Trong quá

trình nghiên cứu, chúng tôi phát hiện ra một vấn đề là sự kết hợp thích hợp của PSNR, dung lượng và tương quan là cần thiết để có thể gửi dữ liệu qua kênh không an toàn mà không sợ hãi người thứ ba. Kết quả trong steganography chủ yếu phụ thuộc vào dữ liệu bí mật. Giá trị lớn hơn của dữ liệu bí mật, ảnh hưởng nhiều hơn đến chất lượng của hình ảnh stego so với giá trị nhỏ hơn của dữ liệu bí mật. Cơ sở dữ liệu được lựa chọn để triển khai phương pháp đề xuất gồm 3 ảnh bìa và 9 ảnh mật định dạng .png.

Trong phương pháp đề xuất, DWT được sử dụng để phân giải hình ảnh thành các dải tần số cao và thấp. Dữ liệu bí mật được biến đổi sử dụng Arnold transformation. Sơ đồ dòng dữ liệu của phương pháp đề xuất được hiển thị trong Hình 2. Trong phương pháp đề xuất, hình ảnh bí mật được chia thành các thành phần R/G/B và nhúng vào HL sub band của RGB tương ứng. Cùng một quy trình phải được thực hiện cho hình ảnh bí mật số 2 và số 3.



Hình 2.26: Sơ đồ luồng dữ liệu của phương pháp đề xuất

Qua hình trên, đọc bìa và tiết ra các hình ảnh và chia ảnh bìa thành các thành phần của nó. Áp dụng DWT trên cả ba thành phần. Thay đổi hình ảnh tiết bằng cách sử dụng phép biến đổi Arnold và tách từng thành phần màu của hình ảnh tiết đã thay đổi. Nhúng các thành phần hình ảnh tiết vào băng con HL, HH và LH. Áp dụng DWT nghịch đảo và thu được hình ảnh stego.

Quy trình khôi phục ngược lại với việc nhúng. Đọc bìa và hình ảnh stego. Chia ảnh bìa và ảnh stego thành các thành phần. Áp dụng phép biến đổi nghịch đảo của Arnold và thu được hình ảnh tiết ra

Từ nghiên cứu, người ta thấy rằng có một số số liệu có thể được sử dụng để kiểm tra hiệu suất của một phương pháp cụ thể. Vì vậy, các tác giả đã xem xét bốn thước đo để đánh giá phương pháp tiếp cận của họ, được thảo luận dưới đây:

1. Tỉ Lệ Tín Hiệu so với Nhiều (PSNR): PSNR biểu thị độ chính xác của việc tái tạo hình ảnh sau khi biến đổi. Chỉ số này được sử dụng để phân biệt giữa hình ảnh gốc và hình ảnh stego. PSNR được tính bằng công thức:

$$PSNR = \frac{10 \log 255^2}{MSE}$$

2. Sai Số Bình Phương Trung Bình (MSE): MSE có thể được định nghĩa là độ chính xác trung bình của các bình phương sai số giữa cường độ của hình ảnh stego và hình ảnh gốc. Nó thường được sử dụng bởi vì tính toán toán học dễ dàng mà nó cung cấp. Nó được biểu diễn như sau:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (f(i,j) - f'(i,j))^2$$

Trong đó: $f(i,j)$ là ảnh gốc

$f'(i,j)$ là ảnh stego

Giá trị MSE càng lớn thì chất lượng càng kém và ngược lại

3. Dung lượng (Capacity): Dung lượng có thể được xác định như sau

$$Capacity = \frac{\text{Số lượng pixel của hình ảnh bí mật được ẩn}}{\text{Số lượng pixel của hình ảnh gốc được sử dụng để ẩn dữ liệu}}$$

4. Tương quan: Nếu chúng ta có một loạt các phép đo của X và Y được ký hiệu là x_i và y_i nơi $i = 1, 2, \dots, n$ thì hệ số tương quan mẫu có thể được sử dụng để ước lượng hệ số tương quan tổng thể Pearson r giữa X và Y. Hệ số tương quan mẫu được viết là:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Phương pháp đề xuất đã được thực hiện trong MATLAB 7.0 và các hình ảnh màu được sử dụng trong việc triển khai. Các hình ảnh khác nhau sử dụng cho thực nghiệm được mô tả như sau:

- **Hình Ảnh Gốc:** Trong quá trình thực hiện, Lena được sử dụng làm hình ảnh gốc, có kích thước 512x512. Hình ảnh ở định dạng .png.
- **Hình Ảnh Stego:** Sau khi nhúng dữ liệu (hình ảnh) vào hình ảnh gốc, hình ảnh stego được thu được. Hình ảnh Gốc và Stego được hiển thị bên dưới trong Hình dưới
- **Hình Ảnh Bí Mật:** Chín hình ảnh bí mật có kích thước 256x256 (Penguin, House, và Girl) được sử dụng làm dữ liệu để ẩn trong hình ảnh gốc. Mỗi hình ảnh ở định dạng .png
- **Hình Ảnh Phục Hồi:** Bằng cách áp dụng quy trình trích xuất, chúng tôi đã phục hồi được các hình ảnh bí mật từ hình ảnh stego. Tất cả hình ảnh bí mật và phục hồi được hiển thị trong Hình sau



Cover Image (Lena)



Stego Image (Lena)

Hình 2.27: Ảnh Cover và ảnh Stego của Lena



Ảnh 2.28: Hình ảnh bí mật ((a), (b), (c)) và

Hình ảnh được phục hồi tương ứng của chúng ((d), (e), (f))

Kết quả của phương pháp đề xuất thu được dưới dạng PSNR, Dung lượng và Tương quan cho hình ảnh. Thời gian thực hiện thuật toán cũng được tính toán.

Tất cả các thuật toán được triển khai trong MATLAB 7.0 trên máy tính có cấu hình Pentium(R) Dual CPU 1.73GHz, RAM 1 GB.

Công trình được trình bày trong bài báo này đề cập đến kỹ thuật giấu ảnh sử dụng biến đổi wavelet rời rạc. DWT được áp dụng trên ảnh màu. Phép biến đổi Arnold được sử dụng để cải thiện tính bảo mật. Phương pháp đề xuất có gắng khắc phục những nhược điểm của các phương pháp giấu tin hình ảnh tương tự trước đó.

Các kết quả được hiển thị dưới dạng stego và hình ảnh được phục hồi. Kết quả PSNR, công suất và mối tương quan được thể hiện dưới dạng bảng. Việc phân tích thuật toán được thực hiện bằng cách so sánh phương pháp được đề xuất với các phương pháp tương tự hiện có. Từ kết quả, có thể rút ra kết luận rằng phương pháp đề xuất là vượt trội hơn về PSNR và khả năng nhúng cao.

2.2.3.6. Các phương pháp khác

Ngoài các phương pháp được liệt kê chi tiết ở trên, còn có một số phương pháp khác cũng được ứng dụng vào trong giấu tin trong ảnh. Tuy nhiên, trong khuôn khổ bài báo cáo này sẽ không miêu tả cụ thể mà chỉ khái quát tổng quát nhất về các phương pháp này.

- *Masking and Filtering*: là hai kỹ thuật được sử dụng để ẩn thông tin trong hình ảnh kỹ thuật số. Cả hai kỹ thuật này tận dụng các đặc tính của hình ảnh và cảm nhận của con người để cải thiện hiệu quả của việc giấu tin.
Với
 - *Masking* (che giấu) là một kỹ thuật steganography liên quan đến việc che đậy thông tin quan trọng (như một văn bản bí mật hoặc một hình ảnh khác) trực tiếp vào trong hình ảnh. Điều này thường được thực hiện bằng cách sử dụng các vùng của hình ảnh có độ tương phản cao hoặc các đặc điểm nổi bật khác, như vùng biên hoặc khu vực có chi tiết hình ảnh phức tạp. Kỹ thuật này hoạt động tốt với hình ảnh màu sắc đậm hoặc có nhiều chi tiết, vì sự thay đổi do việc giấu thông tin sẽ kém nổi bật hơn.
 - *Filtering* (lọc) trong steganography liên quan đến việc sửa đổi thông tin của một hình ảnh thông qua quá trình xử lý hình ảnh, như làm mờ, làm sắc nét, hoặc áp dụng các bộ lọc khác để giấu thông tin. Bộ lọc có thể được thiết kế để chỉnh sửa các pixel cụ thể một cách khéo léo để chừa thông tin bí mật mà không làm thay đổi đáng
- *Error Level Analysis*: Phương pháp này dựa trên việc phân tích mức độ lỗi khi một hình ảnh được nén lại nhiều lần. Thông tin bí mật có thể được thêm vào bằng cách chỉnh sửa mức độ lỗi của các pixel để khi hình ảnh được nén, những thay đổi này sẽ tạo ra một mẫu có thể nhận biết được

2.2.4. Giấu Tin Trong Video (Video Steganography)

Giấu tin trong video là một phần của steganography, nơi thông tin bí mật được nhúng vào trong các video. Video là một phương tiện phức tạp hơn để giấu tin so với

hình ảnh tĩnh vì nó bao gồm nhiều khung hình và thường có cả âm thanh đi kèm. Điều này cung cấp nhiều lựa chọn hơn để ẩn thông tin, nhưng cũng đặt ra những thách thức về việc duy trì chất lượng video và đồng thời giấu thông tin một cách khéo léo.

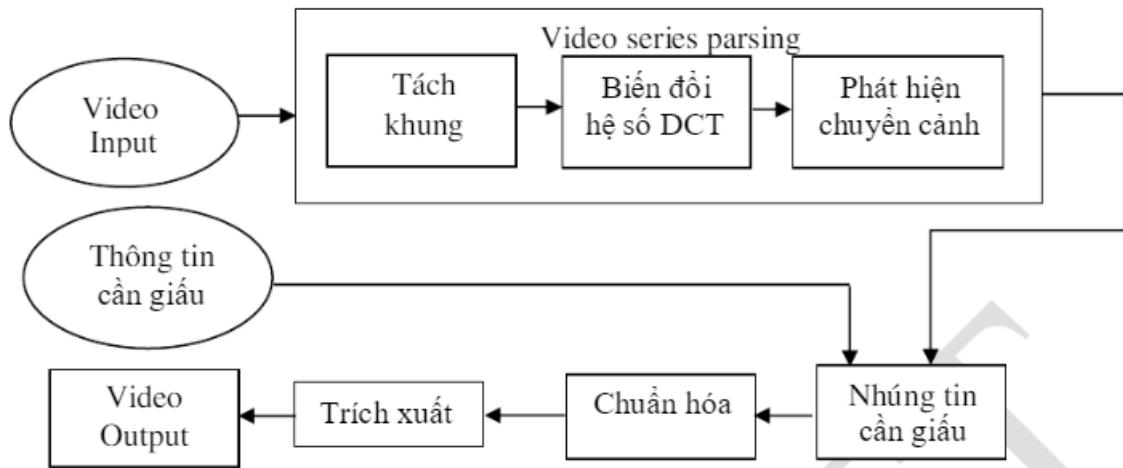
Về cơ bản, video là một loạt các hình ảnh được chạy theo tốc độ được cài đặt sẵn, càng nhiều ảnh chạy cùng một lúc, chất lượng video càng chân thực hơn. Chính vì vậy, các phương pháp giấu tin trong video cũng tương tự như giấu tin trong ảnh. Cho nên các phương pháp được nghiên cứu dưới đây sẽ tập chung vào các phương pháp khác chỉ ứng dụng được trên video.

2.2.4.1. Phương pháp phát hiện thay đổi khung cảnh

Phương pháp giấu tin trong video trên cơ sở phương pháp phát hiện chuyển cảnh là phương pháp giấu tin vào các khung hình của video. Phương pháp này dựa vào sự thay đổi các khung cảnh trong video để giấu thông tin. Cảnh được định nghĩa là những bức hình liên tục chứa các đối tượng (vật thể trên cảnh đó), với mỗi khung hình liên tục thì một cảnh sẽ bao gồm những đối tượng đó. Bình thường video sẽ phân thành các shots. Mỗi shots sẽ thể hiện một sự kiện hay hành động. Trình tự của các khung hình sẽ sắp xếp theo việc ghi hình và chỉnh sửa. Sự khác biệt giữa các khung sẽ đều chỉ ra các điểm chuyển cảnh. Trong chuyển cảnh sẽ bao gồm 2 loại:

- Chuyển cảnh đột ngột (nhanh): Đây là những chuyển cảnh gây ra bởi việc chỉnh sửa của người làm video.
- Chuyển cảnh từ từ (chậm): Đây là những chuyển cảnh do việc quay của người làm video

Các kỹ thuật chuyển cảnh trên rất khó có thể phát hiện bằng mắt thường. Có nhiều phương thức có thể phát hiện được sự chuyển cảnh ví dụ như: dựa vào biểu đồ màu sắc, hệ số DCT,... Sau đây bài giảng sẽ trình bày về thuật toán phát hiện chuyển cảnh dựa vào hệ số DCT. Sơ đồ sau trình bày tổng quan về quy trình giấu tin dựa trên sự thay đổi khung cảnh.



Hình 2.29: Quy trình giấu tin trong video dựa trên kỹ thuật phát hiện chuyển cảnh

Việc giấu tin dựa trên phát hiện chuyển cảnh trải qua 3 giai đoạn chính:

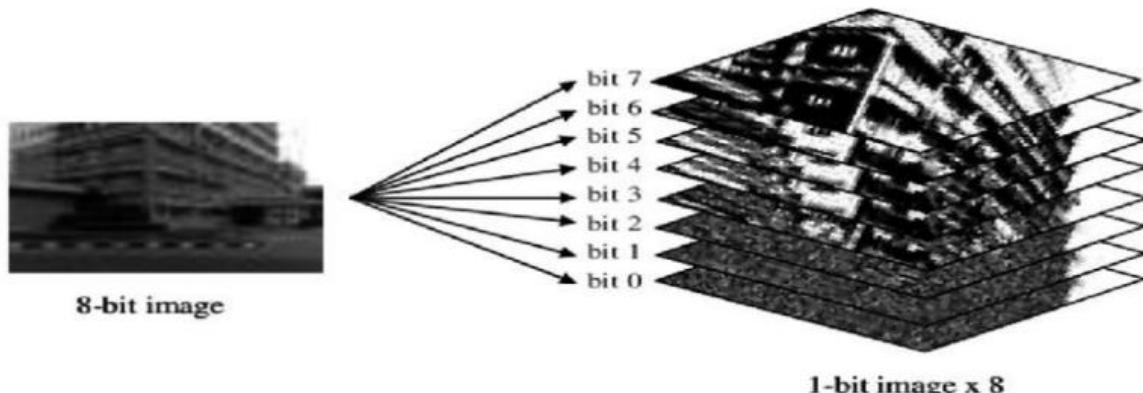
- Video series parsing (Phân tích chuỗi video): Ở giai đoạn này video đầu vào là vật chứa sẽ được phân tích thành các frames (khung) riêng biệt. Sau đó từ các frames sẽ thực hiện biến đổi DCT để thu được các hệ số cosin rời rạc. Sau đó từ những hệ số đã biết của các khối trên những khung hình, sẽ tiến hành phát hiện chuyển cảnh (detect scene change)
- Giấu tin: Sau khi đã phát hiện ra các khung cảnh thay đổi, có thể thỏa thuận với đối tượng cần trao đổi như: sẽ giấu vào frames chuyển cảnh nào, từ những frames đó sẽ xét xem thứ tự để giấu tin như thế nào, ở đây có thể dùng LSB hoặc một số kỹ thuật khác để giấu.
- Chuẩn hóa: Bước chuẩn hóa này nhằm mục đích hạn chế dư thừa dữ liệu, loại bỏ những phần tử cấu trúc phức tạp, nhưng vẫn đảm bảo không làm mất dữ liệu, tiết kiệm không gian lưu trữ.

2.2.4.2. Phương pháp mặt phẳng bit

Bit Plane Complexity Segmentation steganography (BPCS): phương pháp giấu tin trong mặt phẳng bit là phương pháp giấu tin trong video dựa trên sự biến đổi các khung hình của video. BPCS là các mặt phẳng bit trong mỗi khung hình của video

Mặt phẳng bit: Dựa trên độ sâu màu của điểm ảnh. Giả sử một khung hình ($n \times n$ pixel) với độ sâu màu 8 bit sẽ có 8 mặt phẳng. Tương tự với độ sâu màu là 24 và 32 thì

sẽ có 24 mặt phẳng và 32 mặt phẳng. Hình 4.2 dưới đây biểu diễn bit điểm ảnh thành mặt phẳng bit.



Hình 2.30: Biểu diễn 1 điểm ảnh bit thành 8 mặt phẳng bit

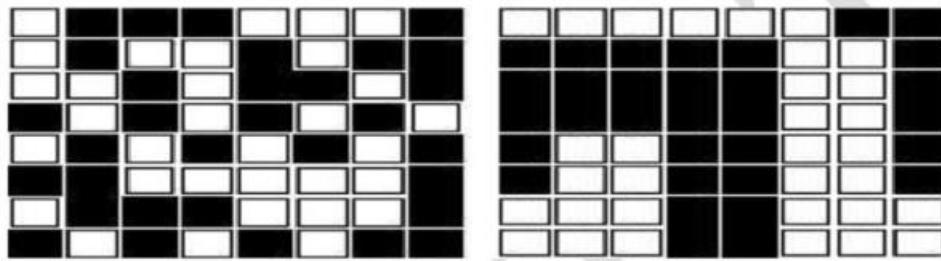
Với các giá trị nhị phân và một phần của khung hình sẽ tạo được các mặt phẳng bit. Mỗi mặt phẳng bit là cấu trúc dữ liệu được làm từ tất cả các bit quan trọng nhất định từ mỗi chữ số nhị phân, với vị trí không gian được giữ nguyên. Ví dụ với khung hình 8×8 pixel với độ sâu màu 8 bit. Trong mặt phẳng bit sẽ biểu diễn như sau: Màu đen biểu diễn bit 0 và màu trắng biểu diễn bit 1. Điểm ảnh đầu tiên biểu diễn dưới dạng 01001110:

- Mặt phẳng bit thứ nhất tại $(0,0)$ là ô màu đen (giá trị 0)
- Mặt phẳng bit thứ hai tại $(0,0)$ là ô màu trắng (giá trị 1)
-
- Mặt phẳng bit thứ 8 tại $(0,0)$ là ô màu đen (giá trị 0)

Mỗi mặt phẳng bit nếu là nhiễu có thể giấu được 1 bit thông điệp cần gửi đi. Theo phương pháp giấu tin dựa trên mặt phẳng bit thì thông tin sẽ được giấu vào các mặt phẳng bit mà có độ nhiễu cao. Để xác định được mặt phẳng bit có khói nhiễu cao hay thấp, có thể áp dụng phương pháp để tính ra độ phức tạp của mặt phẳng bit. Quy trình tính toán như sau:

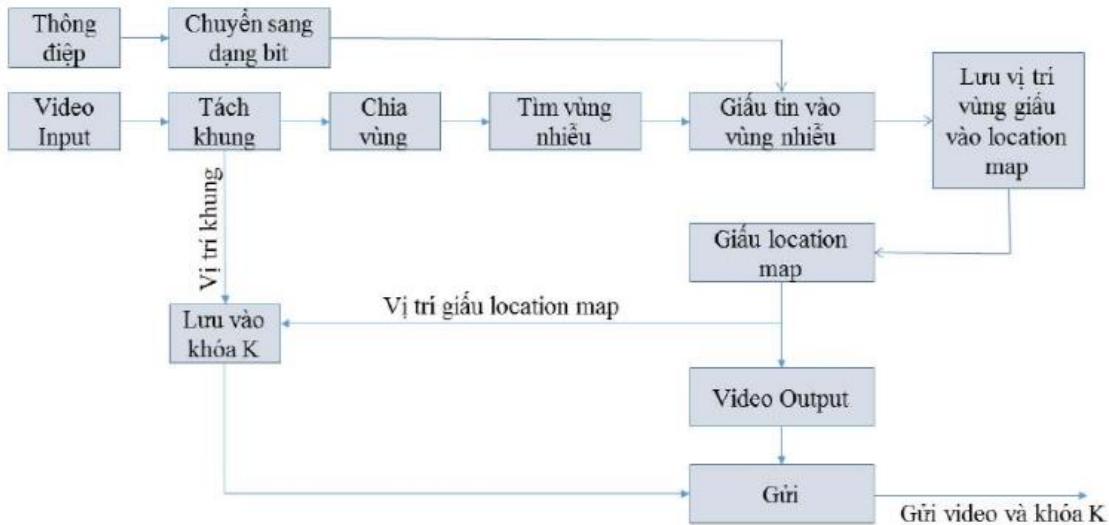
- Độ phức tạp của mặt phẳng bit: Là sự chuyển tiếp từ bit 1 thành bit 0 và từ bit 0 thành bit 1 bao gồm cả chiều ngang và chiều dọc, không liên quan đến số lượng các giá trị 0 và 1.

- **Ngưỡng phức tạp:** là ranh giới phân biệt độ phức tạp cao và độ phức tạp thấp. Trong một số trường hợp, ngưỡng phức tạp được áp dụng để xác định vị trí các mảng bit để giàu thông tin
- **Khối nhiều thông tin:** là vùng có độ phức tạp thấp hơn ngưỡng phức tạp. Nếu thay đổi thông tin ở đây sẽ xảy ra sự thay đổi hình dạng của khung hình. Đây là vùng có nhiều thông tin quan trọng của hình ảnh, dẫn đến sự thay đổi lớn nếu thay đổi thông tin ở mảng bit này.
- **Khối nhiễu:** là vùng có độ phức tạp cao hơn ngưỡng phức tạp. Đây là vùng để giàu thông điệp vì đây là vùng ít thông tin quan trọng của hình ảnh. Do đó hệ thống thị giác của con người khó phát hiện được sự thay đổi. Nếu thay đổi không làm thay đổi quá nhiều đến chất lượng của hình ảnh. Trong thực tế phải chọn các mảng bit được gọi là nhiễu để giàu thông điệp vào đó. Ví dụ về vùng nhiễu và vùng nhiều thông tin được giới thiệu như hình 4.3 dưới đây. Trong đó màu trắng là giá trị 1 và màu đen là giá trị 0.



Hình 2.31: Phân loại vùng nhiễu và vùng nhiều thông tin

Tùy quy tắc tính như trên có thể thấy: Đối với hình a được coi là vùng nhiễu vì độ phức tạp của mảng bit là 69. Đối với hình b được coi là vùng nhiều thông tin vì độ phức tạp của mảng bit là 29. Như vậy thông tin sẽ được nhúng vào vùng nhiễu như ở hình trên:



Hình 2.32: Quy trình giấu tin trong video vào mặt phẳng bit

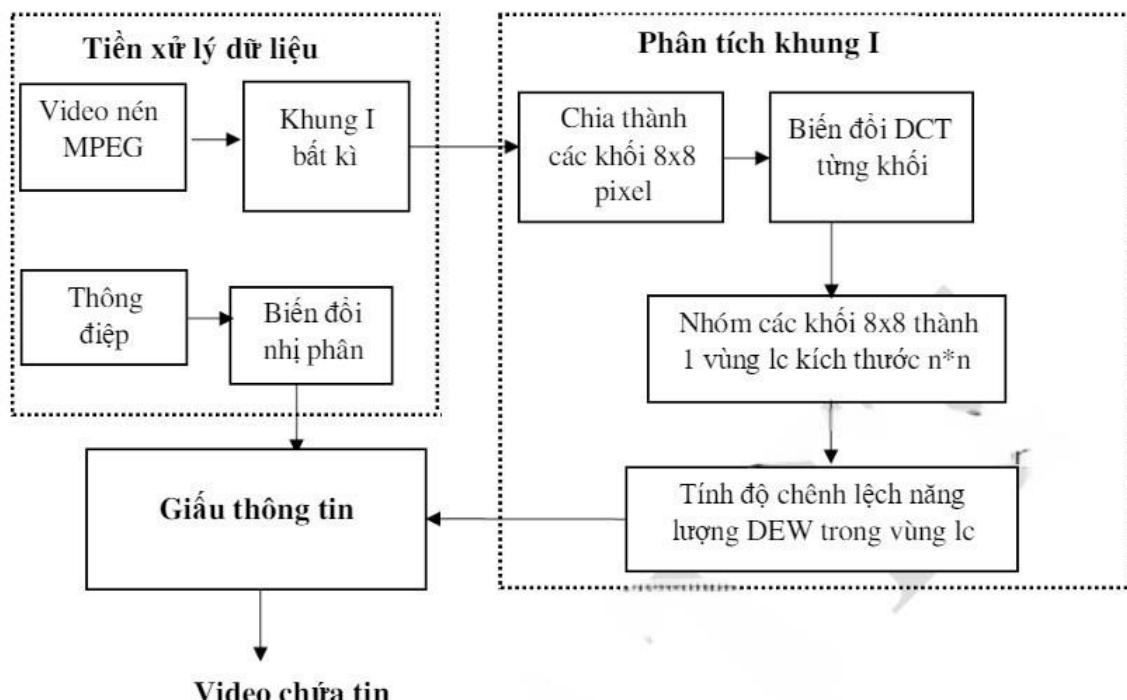
Từ sơ đồ quy trình giấu tin trong video như hình 4.4 thấy được các bước chính của kỹ thuật giấu tin vào mặt phẳng bit như sau:

- ✓ **Bước 1: Tiền xử lý dữ liệu:** với 2 thông tin đầu vào là video input và thông tin mật
 - *Đối với thông tin mật:* chuyển thông tin mật thành dạng nhị phân.
 - *Đối với video input:* tiến hành tách video thành các khung hình. Chọn một khung ảnh bất kì để chuẩn bị giấu thông tin mật. Việc chọn vị trí khung sẽ được lưu vào khóa K. Vị trí này sau này sẽ hỗ trợ cho người tách tin tìm thấy khung hình để tách tin
 - *Chia vùng:* Sau khi chọn được khung hình sẽ tiến hành chia vùng để tạo thành các mặt phẳng bit. Mỗi pixel có độ sâu màu là 8, 24, 32 bit thì sẽ có 8, 24, 32 mặt phẳng bit tương ứng.
 - *Tìm vùng nhiễu:* Tại khung hình vừa lựa chọn, sau khi đã xác định độ sâu của ảnh, người giấu tin sẽ tính toán độ phức tạp của mặt phẳng để tìm xem đâu là vùng nhiễu đâu là vùng nhiều thông tin. Quy trình tính toán để xác định vùng nhiễu và vùng nhiều thông tin đã được trình bày ở bước trên
- ✓ **Bước 2. Giấu tin mật:** Thông điệp được chuyển dạng nhị phân rồi giấu vào vùng nhiễu đã được tìm ra ở trên. Phương pháp giấu thông tin vào vùng nhiễu có thể lựa chọn sử dụng phương pháp thay thế LSB. Tiếp đến người giấu tin cần lưu vị trí các khối nhiễu vào location map để làm cơ sở cho người tách tin tìm ra các vị trí tin giấu. Người giấu tin cũng có thể nhúng cả location map cùng các khối bí mật và chỉ

lưu vị trí của khối này hoặc lưu trữ riêng cả location map này vào khóa K. Cuối cùng người giấu tin sẽ chuyển video đã giấu tin và khoá K cho bên nhận.

2.2.4.3. Phương pháp giấu trong miền video nén dựa trên sự khác biệt năng lượng

Các kỹ thuật nhúng thủy văn dựa trên mối tương quan có lợi thế là có thể lấy thủy văn ra được từ giải mã các luồng video hoặc mã hóa lại chúng. Tuy nhiên để nhúng hoặc phát hiện một thủy văn dựa trên mối tương quan thì giải mã MPEG là điều bắt buộc. Điều này có thể quá đòi hỏi quá trình tính toán phức tạp. Ngược lại thấy rằng thuật toán LSB có tính hiệu quả về mặt tính toán cao. Trên thực tế, các ứng dụng đòi hỏi mức độ bảo mật mạng ngang với kỹ thuật nhúng thủy văn dựa trên mối tương quan và có hiệu quả tính toán giống như phương pháp dựa trên LSB. Bởi vậy DEW (Difference Energy Watermarking) được phát triển để thỏa mãn nhu cầu này. DEW có thể áp dụng trực tiếp trên video nén MPEG/JPEG cũng như trên video nguyên thủy



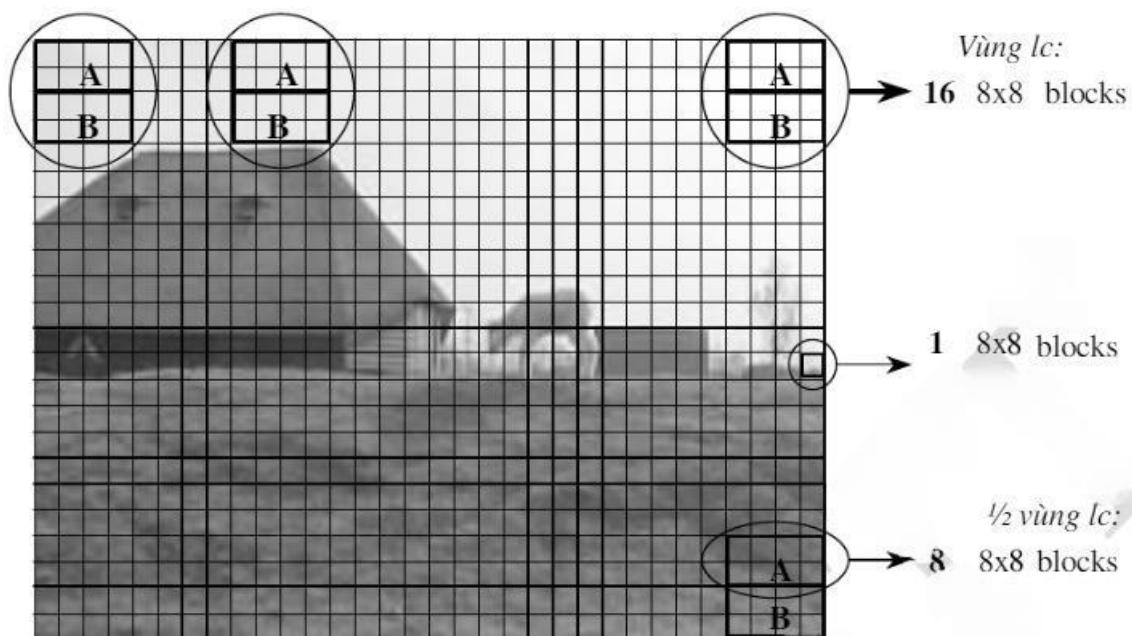
Hình 2.33: Sơ đồ tổng quát phương pháp giấu tin trong miền video nén dựa bằng DEW

Tùy sơ đồ giấu tin trong video theo phương pháp DEW bao gồm các bước sau:

- ✓ Bước 1: Tiền xử lý dữ liệu: với 2 thông tin đầu vào là video input và thông tin mật.

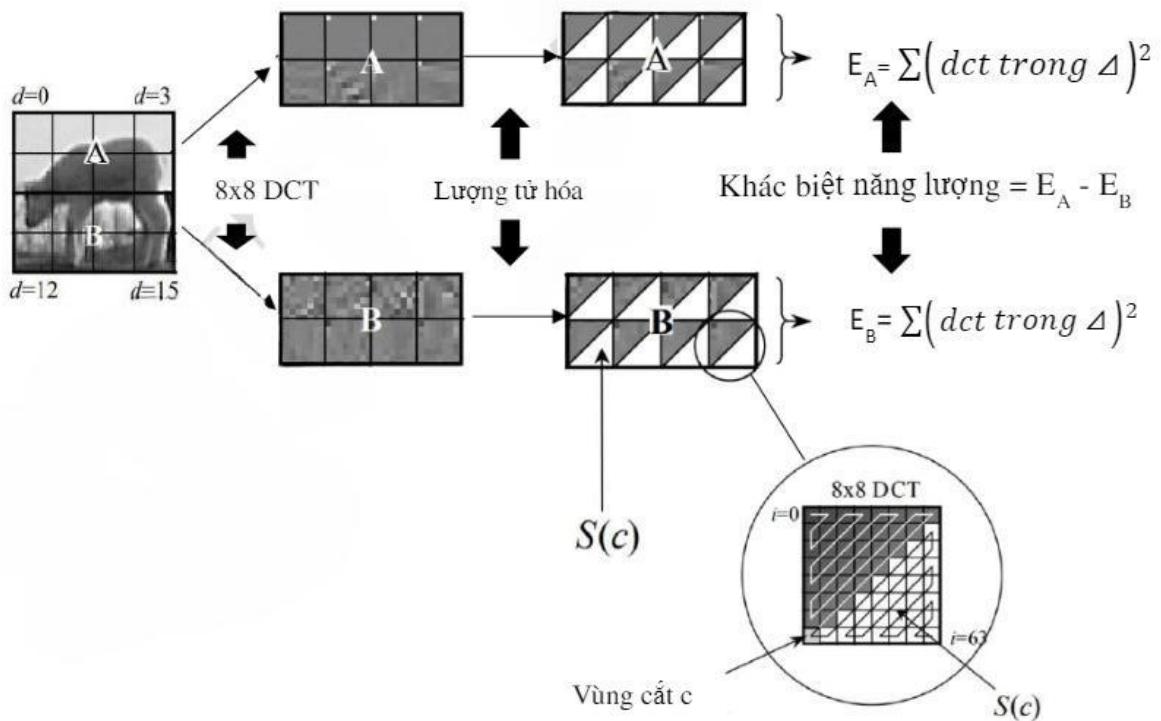
- Đối với thông tin mật: chuyển thông tin mật thành dạng nhị phân.
 - Đối với video input: tiến hành tách video thành các khung hình (tách khung hình ra khỏi luồng nén). Chọn một khung ảnh bất kì để chuẩn bị giàu thông tin mật. Đối với phương pháp DEW nên chọn khung I
- ✓ Bước 2: Phân tích khung hình. Từ sơ đồ hình 4.5 thấy được các bước tiến hành chính trong việc phân tích và xử lý khung hình như sau:

- Ảnh được chia thành các khối 8×8 pixel. Rồi từ đó đưa về hệ số DCT (các khối 8×8 hệ số DCT).
- Nhóm các khối 8×8 thành một vùng lc kích thước n^*n : Lưu ý: Trong trường hợp trên với $n = 16$ khối 8×8 được gọi là lc-region (khu vực lc). Kích thước của vùng này được gắn với giá trị tương ứng trên nhãn. Một lc-region được chia đều thành hai phần A, B mỗi phần tương ứng $8 \text{ khối } 8 \times 8 \text{ DCT}$. Hình 4.6 mô tả ví dụ về việc chia khối lc.



Hình 2.34: Ví dụ về việc chia khối lc

- Tính độ chênh lệch năng lượng DEW trong vùng lc:



Hình 2.35: Quá trình tính toán năng lượng trong vùng lc

- E_A năng lượng nửa trên: Năng lượng trong một vùng E_A bằng tổng bình phương của một tập con cụ thể của các hệ số DCT trong vùng E_A này
- E_B năng lượng nửa dưới: tính tương tự như E_A
- D là sự khác biệt năng lượng. Sự khác biệt được định nghĩa theo công thức:

$$D = E_A - E_B$$

- Tập con này biểu diễn bởi $S(c)$ (hình tam giác trắng trong hình 4.6). Công thức tính năng lượng tại một vùng như sau:

$$E_A(c, n, Q) = \sum_{d=0}^{\frac{n}{2}-1} \sum_{i \in S(c)} (\lfloor \theta_{i,d} \rfloor_Q)^2$$

Trong đó:

- + E_A là năng lượng tại vùng A.
- + d là vị trí khối DCT trong 1 vùng lc.
- + i là vị trí của hệ số DC trong khối DCT
- + $\theta_{i,d}$ (theta) là hệ số DC thứ i của khối DCT thứ d của khu vực A.
- + Q là bước lượng tử hóa (xấp xỉ giá trị)
- Bước 3: giấu thông tin: Sau khi đã tính toán được sự khác biệt năng lượng giữa các vùng thì người giấu tin sẽ tiến hành giấu thông tin. Nhiệm vụ bây giờ là xác

định giá trị của bit tương đương với sự chênh lệch năng lượng D. Bit 0 được xác định là $D > 0$, bit 1 được xác định nghĩa là $D < 0$. Theo đó:

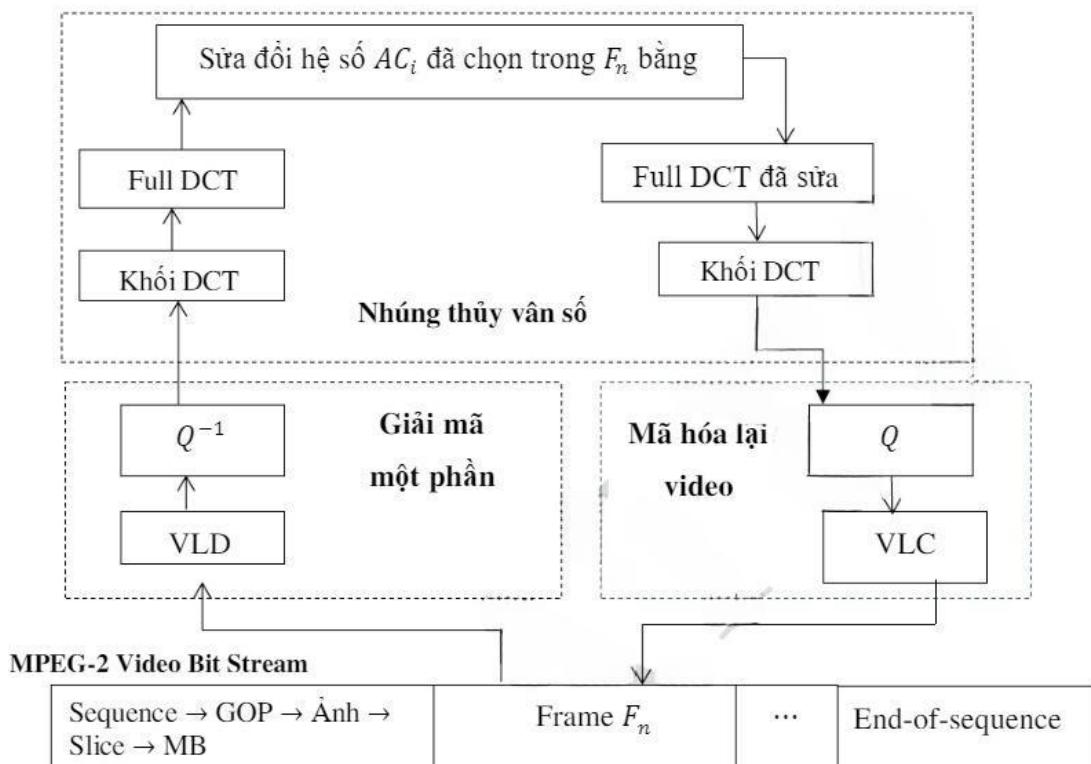
- Nếu bit “0” được giấu, tất cả năng lượng trong vùng “cut-off index c” của vùng B được loại bỏ bằng cách đặt hệ số DCT tương ứng bằng 0. Khi đó:

$$D = E_A - E_B = E_A - 0 = +E_A$$

- Nếu bit “1” được nhúng, tất cả năng lượng trong vùng “cut-off index c” của vùng A được loại bỏ. Khi đó

$$D = E_A - E_B = 0 - E_B = -E_B$$

2.2.4.4. Phương pháp giấu trên miền nén của video chất lượng cao



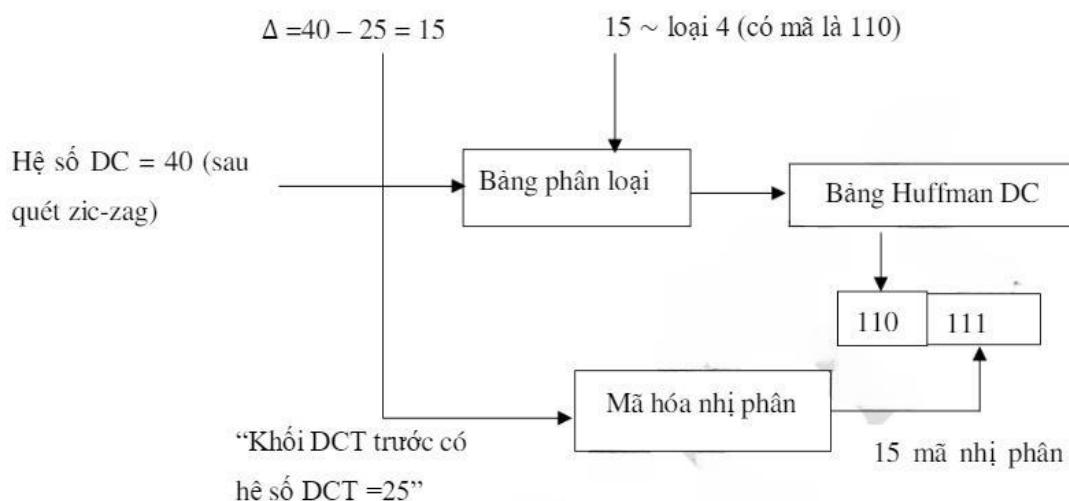
Hình 2.36: Quy trình giấu tin trong nội dung video MPEG -2

Từ quy trình giấu tin trong video thể hiện trên hình thấy được các bước chính trong kỹ thuật giấu tin trên miền nén của video chất lượng cao như sau:

Bước 1. Lựa chọn khung: Chọn một khung bất kỳ để tiến hành nhúng, nên chọn khung I vì khung I là khung cơ sở và có thể coi là ảnh gốc, với khung này khi giải mã thì không cần lấy thông tin từ khung khác.

Bước 2. Giải nén một phần video: sử dụng khung đã chọn ở bước 1. Việc giải nén một phần của video sử dụng phương pháp chính là VLD (Variable Length Decoding- mã hóa có độ dài biến đổi được) và giải lượng tử hóa. Quy trình thực hiện của các phương pháp này như sau:

- VLD: các từ mã có tần suất xuất hiện thấp sẽ được mã hóa bằng các từ mã dài, quá trình này được gọi là phương pháp mã hóa từ mã có độ dài thay đổi. Quá trình mã hóa này được tiến hành trên tất cả các thành phần của hệ số DCT:
 - Với thành phần DC: Giá trị sai lệch hệ số DC sẽ được mã hóa nhờ bảng phân loại và bảng Huffman (dựa vào đặc tính thống kê của tín hiệu). Đây là ví dụ về các bước mã hóa entropy thành phần hệ số DC:

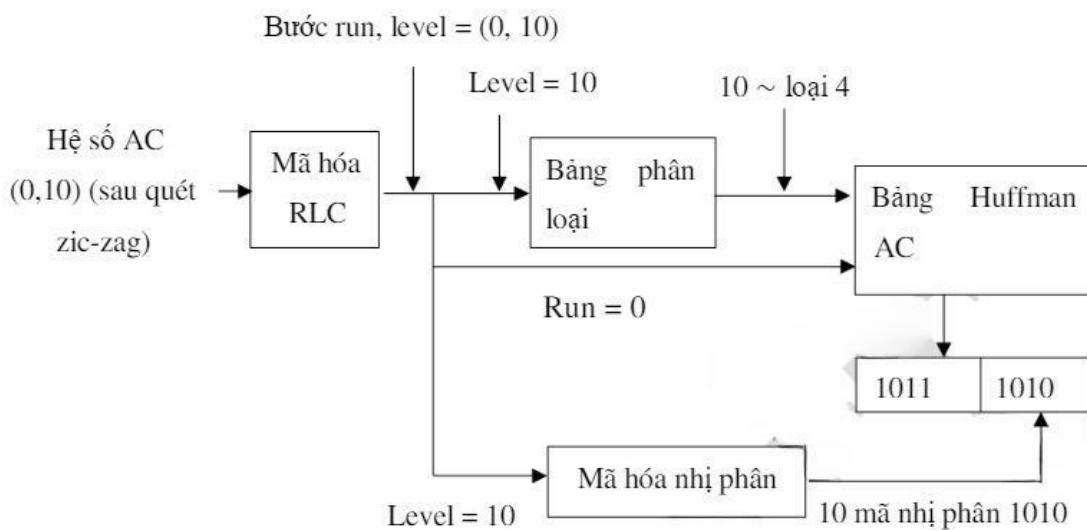


Hình 2.37: Quy trình mã hóa entropy thành phần hệ số DC

Các hệ số DC sai lệch	Phân loại	Tù mã
-255...-128; 128...255	8	1111110
-127...-64; 64...127	7	1111 10
-63...-32; 32...63	6	1111 0
-31...-16; 16...31	5	1110
-15...-8; 8...15	4	110
-7...-4; 4...7	3	101
-3; -2; 2; 3	2	01
-1; 1	1	00
0	0	100

Hình 2.38: Bảng phân loại và bảng Huffman cho thành phần DC

- Với thành phần AC: Hệ số AC cũng được mã hóa nhờ bảng phân loại (giống như DC) và bảng Huffman (nhưng khác DC) như ở hình:



Hình 2.39: Quy trình mã hóa entropy thành phần hệ số AC

Bước run	Phân loại	Độ dài mã	Tử mã
0	1	2	00
0	2	2	01
0	3	3	100
0	4	4	1011
1	1	4	1100
1	2	6	111001
2	1	5	11011
2	2	8	1111 000
3	1	6	111 010
4	1	6	111 011
5	1	7	1111 010
6	1	7	1111 011
EOB		4	1010

Hình 2.40: Bảng Huffman các hệ số AC

- Lượng tử hóa: Đầu vào ở bước này là 64 hệ số DCT của khối 8x8 sẽ được lượng tử hóa dựa trên một bảng lượng tử gồm 64 phần tử $Q(u, v)$ với $0 \leq u, v \leq 7$. Nguyên tắc lượng tử là chia các hệ số $F(u, v)$ cho các hệ số ở vị trí tương ứng trong bảng lượng tử $Q(u, v)$. Bảng lượng tử được xây dựng theo nguyên tắc là mắt người ít cảm nhận được nội dung ở tần số cao và đặc biệt càng kém nhạy với nội dung ở tần số cao của kênh màu. Do đó:

- Các hệ số tương ứng với thành phần DC và các thành phần tần số thấp có giá trị lớn nên phải được lượng tử chính xác.
- Các hệ số tương ứng với thành phần tần số AC có giá trị nhỏ nên cho phép sai số.

Bước 3: Giấu thông tin: Để có thể giấu được thông tin vào khung hình vừa lựa chọn. Tại đây người giấu tin cần thực hiện các biện pháp tiền xử lý như sau:

- Giai đoạn 1: Thực hiện tính toán DCT cho toàn khung hình: Sau khi giải nén một phần video thu được các khối hệ số DCT 8x8 pixel. Ở giai đoạn này người giấu tin cần thực hiện tính toán các DCT toàn khung hình từ khối hệ số DCT 8x8 pixel vừa thu được. Hệ số DCT đầy đủ được tính như sau: giả sử kích thước khung hình là LN × MN và kích thước của một khối Bi, là N. L và M số hàng và cột trong hàng tương ứng.

$$FullDCT = \sqrt{\frac{1}{LM}} A_1 \begin{pmatrix} B_{0,0} & B_{0,1} & \dots & B_{0,M-1} \\ B_{1,0} & B_{1,1} & \dots & B_{1,M-1} \\ \dots & \dots & \dots & \dots \\ B_{L-1,0} & B_{L-1,1} & \dots & B_{L-1,M-1} \end{pmatrix} \cdot A_2^T$$

Trong đó:

LN × MN: kích thước khung hình

L và M số hàng và cột trong hàng tương ứng

N kích thước của một khối $B_{i,j}$. $B_{i,j}$ là ma trận với $N \times N$ yếu tố và đại diện cho tập hợp các hệ số DCT cho khoanh vùng

A1 và A2 là các ma trận vuông với LN × LN và MN × MN kích thước tương ứng và được định nghĩa theo công thức:

$$A_1 = \begin{cases} \frac{\sqrt{1}}{2} a(u, i), & u = 0, \quad i \bmod N \neq 0 \\ \frac{\sqrt{2}}{2} a(u, i), & u \neq 0, \quad i \bmod N = 0 \\ a(u, i), & \text{còn lại} \end{cases} \quad \text{Trong đó: } a(u, i) = \cos\left(\frac{(2i+1)u\pi}{2LN}\right) u, \quad i = 0, 1, \dots, LN-1$$

$$A_2 = \begin{cases} \frac{\sqrt{1}}{2} a(v, j), & v = 0, \quad j \bmod N \neq 0 \\ \frac{\sqrt{2}}{2} a(v, j), & v \neq 0, \quad j \bmod N = 0 \\ a(v, j), & \text{còn lại} \end{cases} \quad a(v, j) = \cos\left(\frac{(2j+1)v\pi}{2MN}\right) v, \quad j = 0, 1, \dots, MN-1$$

- Giai đoạn 2: Điều chỉnh chỉ số lượng tử hóa: Sử dụng phương pháp điều chỉnh chỉ số lượng tử hóa (QIM) để giấu thông tin vào các hệ số tần số thấp của hệ số DCT

toàn khung hình. Để thực hiện được nhiệm này cần thực hiện các quá trình tính kích thước bước Q. Trong thực tế, quá trình tính toàn kích thước bước Q áp dụng công thức dưới đây:

$$\Delta = 2 \max(|\alpha|, |\beta|) = 2 \max \left(2 \left| \sum_{j=1}^n \frac{|X_j - \mu|}{X_{2n, 1 - \frac{\tau}{2}}^2} \right|, 2 \left| \sum_{j=1}^n \frac{|X_j - \mu|}{X_{2n, \frac{\tau}{2}}^2} \right| \right)$$

Trong đó:

α, β : khoảng tin cậy

μ : tham số vị trí (là giá trị của biểu đồ)

τ : tỷ lệ bit lỗi BER

X : chuỗi các biểu đồ khác biệt

X_{2n}^2 : biểu thị định lượng ph của các phân bố X^2 với bậc tự do 2n

- Giai đoạn 3: Chọn vị trí nhúng: Các hệ số xung quanh thành phần DC thường có các giá trị lớn, do đó việc sửa đổi chúng làm giảm chất lượng hình ảnh nghiêm trọng. Ngoài ra, các giá trị hệ số gần thành phần DC thì giá trị của chúng sẽ càng khác nhau sau khi mã hóa lại. Do đó, nên lựa chọn các thành phần tần số trung gian làm vị trí nhúng để cân bằng giữa độ bền và chất lượng hình ảnh. Bên cạnh đó do ảnh hưởng của nén MPEG trên video được nhúng tần số trung bình thấp thích hợp cho việc giấu tin.
- Giai đoạn 4: Giấu thông tin vào hệ số DCT. Sau khi thiết lập các tham số cho QIM, thông tin được nhúng bằng cách thay thế các hệ số DCT bằng các giá trị được lượng tử hóa (xem hình 4.11). Hình mờ bao gồm một chuỗi nhị phân, $w = \{w_1, w_2, \dots, w_n\}$, trong đó $w_k \in \{0, 1\}$ và n có nghĩa là độ dài của thông tin cần giấu. $x = \{x_1, x_2, \dots, x_n\}$ được chọn các hệ số DCT toàn khung của một khung và $y = \{y_1, y_2, \dots, y_n\}$ được sửa đổi hệ số sau khi giấu thông tin. Sử dụng hàm giấu $E(x, w)$ như dưới đây tạo ra các giá trị thay thế có khoảng cách tối thiểu giữa giá trị gốc

$$y_k = E(x_k, w_k) = \text{round} \left(\frac{x_k}{\Delta} \right) \cdot \Delta + d(x_k, w_k)$$

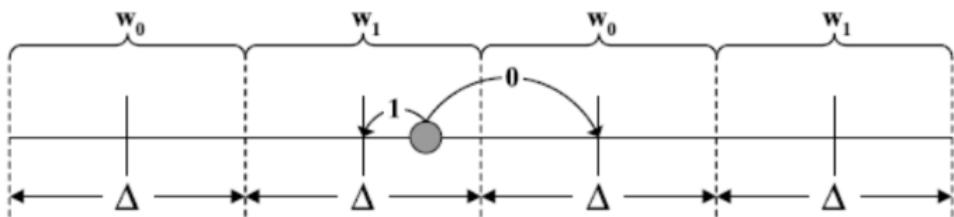
Trong đó:

Δ là kích thước của bước Q

hàm $d(xk, wk)$ biểu thị giá trị dithered tương ứng với bit wk của thông tin mật. Hàm $d(xk, wk)$ được tính theo công thức dưới đây:

$$d(x_k, w_k) = \begin{cases} \frac{\Delta}{2} & \text{if } (R \bmod 2 = 0, w_k = 0) \text{ or } (R \bmod 2 = 1, w_k = 1) \\ -\frac{\Delta}{2} & \text{if } (R \bmod 2 = 0, w_k = 1) \text{ or } (R \bmod 2 = 1, w_k = 0) \end{cases}$$

Trong đó R viết tắt cho $\text{round}\left(\frac{x_k}{\Delta}\right)$



Hình 2.39: Thay thế giá trị cho thông tin cần giấu trong QIM

Giáu thông tin bằng cách sử dụng QIM được mô tả trong hình 4.11. Giả sử rằng vòng tròn màu xám là giá trị hệ số ban đầu. Nếu một bit thông tin cần giấu là “1” được nhúng vào hệ số này, nó được thay thế cho giá trị trung bình của w_1 là giá trị gần nhất với giá trị hệ số ban đầu. Nếu một bit thông tin cần giấu là “0” được nhúng, nó được thay thế cho giá trị trung bình của w_0 gần nhất.

- Giai đoạn 5: biến đổi DCT ngược. Sau khi đã giáu được thông tin bởi QIM bằng cách sử dụng các thông số ở giai đoạn 4 thì các hệ số DCT toàn khung hình đã được sửa đổi và được phân tách thành khối 8×8 pixel các hệ số DCT. Trong giai đoạn biến đổi DCT ngược chính là cần phải tính toán các khối hệ số DCT nghịch đảo. Trong chương 2 của bài giảng đã trình bày chi tiết về quá trình tính toán DCT ngược.

Bước 4: Mã hóa video: Sau khi đã tính toán các khối hệ số DCT nghịch đảo, người giáu tin sẽ tiến hành mã hóa video lại sử dụng VLC và giải lượng tử hóa như đã nói ở quá trình giải nén một phần video để tạo các video MPEG-2 chứa thông tin mật. Lưu ý rằng: Quá trình VLC và giải lượng tử ở phía bộ giải mã được thực hiện ngược lại so với các bước biến đổi ở quá trình giải nén video.

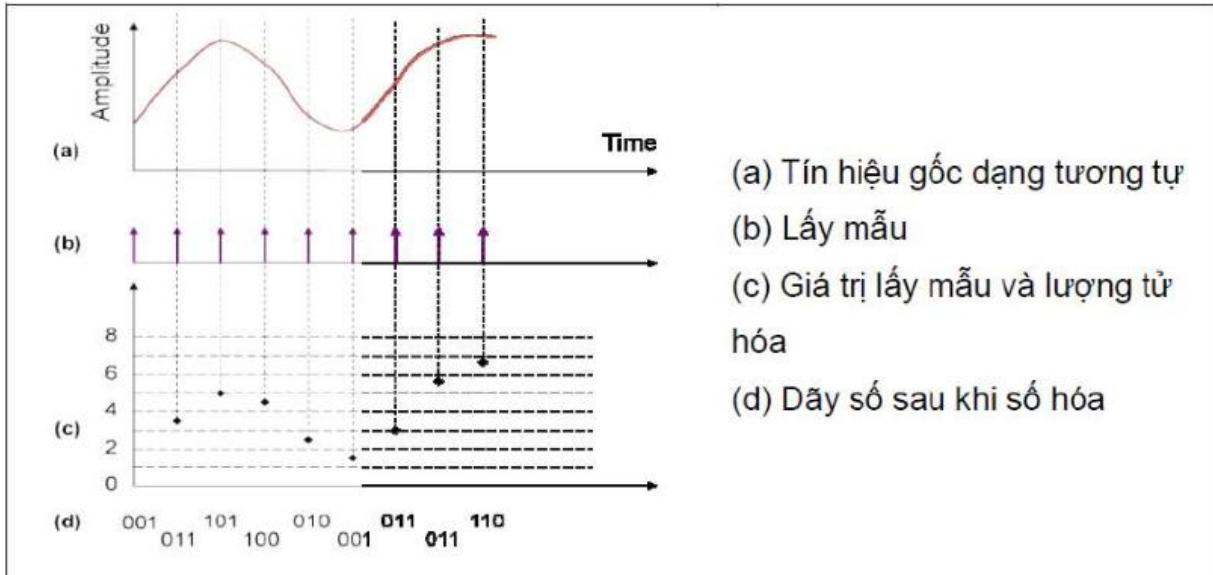
2.2.5. Giấu Tin Trong Âm Thanh (Audio Steganography)

Các vật chứa trong các kỹ thuật giấu tin trong âm thanh là các file âm thanh được đặc trưng bởi tần số, bước sóng, chu kỳ và biên độ, vận tốc lan truyền (tốc độ âm thanh). Lợi dụng đặc điểm đó mà có thể thay đổi theo ý muốn nhằm dấu tin một cách ẩn danh, trong khoảng tần số có sự thay đổi không đáng kể để có thể phát hiện ra.

Đặc điểm của kỹ thuật giấu tin trong âm thanh là giấu thông tin vào các khe hở của âm thanh. Các khe hở ở đây chính là các thành phần như: tần số, biên độ, chu kỳ,... của âm thanh. Các kỹ thuật giấu tin trong âm thanh lợi dụng vào điểm yếu hệ thống thính giác con người (Human Auditory System – HAS)

Một số vấn đề cần lưu ý trong kỹ thuật giấu tin trong âm thanh:

- Tần số mẫu: Để đưa được âm thanh vào các ứng dụng của giấu tin, cần xác định biên độ dao động của sóng âm vào các thời điểm khác nhau. Công việc này gọi là trích/lấy mẫu. Với một giây phát ra âm thanh, trích lấy một số mẫu biên độ đưa vào dữ liệu, con số ấy gọi là tần số trích mẫu (sample rate). Tần số này cho biết biên độ rung mỗi giây của sóng âm thanh. Thí dụ, tần số mẫu là 44,1 kHz thì mỗi giây tín hiệu nhận được bị cắt thành 44100 lát
- Độ dày của bit: Để lưu lại dưới dạng số, mỗi mẫu được biểu diễn bằng một lượng bit dữ liệu nhất định nào đó, gọi là BitDepth. Với tập tin WAV thường là 8 hoặc 16 bits. BitDepth càng lớn thì âm thanh lấy mẫu càng chính xác và người nghe càng thấy sắc nét, trung thực. Giả sử, nếu lấy được mẫu với tần số 44,1kHz (44100 lần/giây), 16 bit (tương đương với chất lượng CD) thì khi đó 1 phút âm thanh sẽ tiêu tốn tới 10MB ổ cứng
- Kích thước mẫu trích: Công thức kích thước mẫu trích (được tính bằng byte) như sau: $\text{LengthOfSample} = \text{Channels} * \text{AudioSampleSize} / 8$
- Âm thanh số: là các mẫu lấy theo phương pháp lượng tử hóa, chuyển đổi giá trị mẫu (liên tục thành các giá trị rời rạc).



Hình 2.40: Ví dụ về tín hiệu âm thanh và mẫu

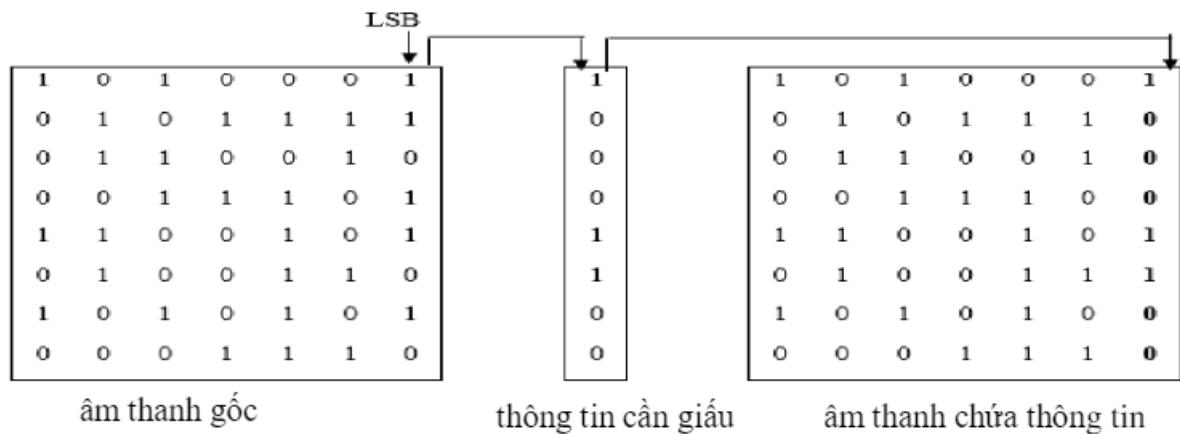
2.2.5.1. Phương pháp LSB

Tương tự như giấu tin trong ảnh và video. Ta cũng có thể chia thông điệp ra thành các bit riêng lẻ và giấu vào trong file nhị phân. Cách thay thế này là cách đơn giản để nhúng thông tin vào một tệp âm thanh kỹ thuật số. Phương pháp LSB cho phép một lượng lớn dữ liệu được nhúng, tốc độ truyền dữ liệu nhanh.

Để thực hiện được điều đó thì người giấu tin cần thực hiện những thao tác sau:

- **Bước 1:** Đọc file âm thanh gốc. Chia âm thanh gốc thành các segment. Thông thường, người giấu tin sẽ chia file âm thanh các segment dựa trên độ dài bit của thông tin cần giấu. Sau đó các segment này được vector giá trị của tín hiệu, rồi lưu vào mảng một chiều để thực hiện giấu tin.
- **Bước 2:** Chuyển đổi thông tin cần giấu sang dạng nhị phân, tính độ dài bit của thông tin rồi lưu vào L
- **Bước 3:** Chọn k là số bit LSB của tín hiệu âm thanh sẽ giấu sao cho phù hợp nhất
- **Bước 4:** Chia chuỗi bit thông điệp thành các chuỗi con có độ dài k bit. Trong đó, mỗi chuỗi con này sẽ được thay thế vào k bit LSB của L/K tín hiệu âm thanh để giấu đủ L bit thông điệp

- **Bước 5:** Thực hiện giấu L bit đã tính vào các segment. Để tăng độ an toàn cho kỹ thuật này, có thể sử dụng bộ sinh số ngẫu nhiên để sinh ra các vị trí các mẫu được chọn giấu chứ không phải các mẫu liên tục. Bộ sinh số này sử dụng một khóa bí mật như là phần tử khởi tạo bộ sinh số. Khóa được sử dụng trong cả quá trình giấu tin và giải tin
- **Bước 6:** Lưu lại tệp âm thanh kết quả F' được thông tin đã giấu



Hình 2.41: Mô tả phương pháp thay thế bit trong thuật toán LSB

Hình trên thể hiện pháp thay thế LSB với trường hợp thay thế 1 bit LSB. Trong thực tế hiện nay cũng có một số hướng tiếp cận khác nhằm nâng cao chất lượng giấu tin trong kỹ thuật LSB. Ví dụ phương pháp sử dụng 4 bit LSB thay vì 1 bit LSB đơn lẻ hoặc phương pháp kết hợp giữa bít quan trọng nhất (MSB -Most Significant Bit) và LSB

2.2.5.2. Phương pháp mã hóa pha

Mã hóa pha trong âm thanh hoạt động bằng cách thay pha của đoạn âm thanh ban đầu với pha được mã hóa của dữ liệu. Phương pháp mã hóa pha dựa vào tính chất là các thành phần của pha không gây ảnh hưởng đến hệ thống thính giác con người như nhiều. Như đã giới thiệu về HAS ở trên, HAS rất nhạy cảm trong miền thời gian nên dễ phát hiện ra thay đổi nhỏ. Nhưng Moore đã chứng minh được rằng HAS lại ít nhạy cảm với các thay đổi pha và đặc tính này được khai thác trong hệ thống nén âm thanh số. Ý tưởng của phương pháp này là chia chuỗi âm thanh gốc thành các khối và nhúng toàn bộ dữ liệu vào phô pha của khối đầu tiên. Việc giấu tin sẽ được thực hiện bằng cách điều chỉnh pha trong phô pha của dữ liệu

Quy Trình

Trong mã hóa pha, mỗi dữ liệu được coi là một dịch pha (phase shift) trong phô pha của tín hiệu sóng mang. Xét tín hiệu sóng mang c , c được chia thành N phần nhỏ và mỗi phần tử $c_i(n)$ có chiều dài l_m . Lúc này áp dụng biến đổi Fourier có:

- Độ lớn tín hiệu được tính bằng công thức:

$$A_i(k) = \sqrt{Re[F\{c_i(k)\}]^2 + Im[F\{c_i\}(k)]^2}$$

- Ma trận độ lớn pha có các phần tử được tính theo công thức:

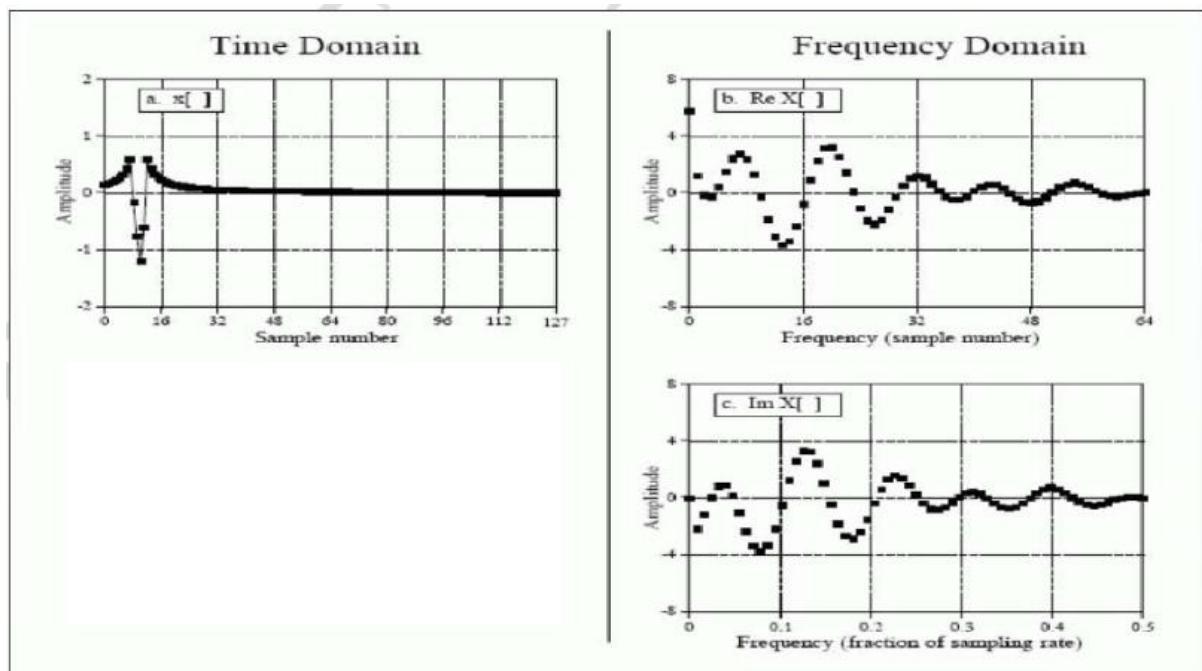
$$\varphi_i(k) = \arctan \frac{Im[F\{c_i\}(k)]}{Re[F\{c_i\}(k)]}$$

Trong đó:

Re : là phần thực

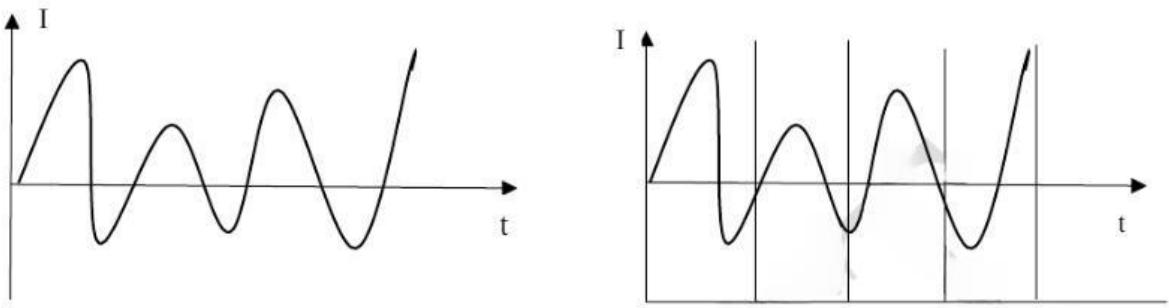
Im : là phần ảo

t : là thời gian



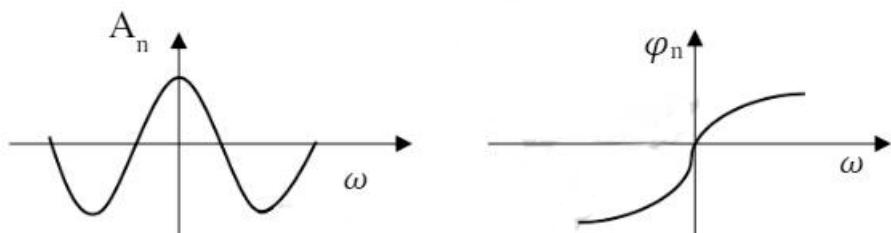
Hình 2.42: Phân tích các thành phần của dữ liệu âm thanh

Bước 1: Dữ liệu âm thanh gốc có chiều dài N được chia thành các segment có chiều dài bằng chiều dài với thông tin cần giấu.



Hình 2.43: Mô tả chia âm thanh gốc thành các segment bằng nhau

Bước 2: Mỗi đoạn segment được biến đổi bằng Fourier DFT với ma trận độ lớn phase là $\varphi_j[\omega_k]$ và ma trận độ lớn tín hiệu là $|A_j[\omega_k]|$ với $0 \leq k \leq n/2 - 1$, $0 \leq j \leq N - 1$



Hình 2.44: Minh họa khi mỗi đoạn được biến đổi bằng DFT

Bước 3: Tính độ lệch pha giữa các đoạn kề nhau bằng công thức sau:

$$\Delta\varphi_j[\omega_k] = \varphi_{j+1}[\omega_k] - \varphi_j[\omega_k] \quad \forall j, k$$

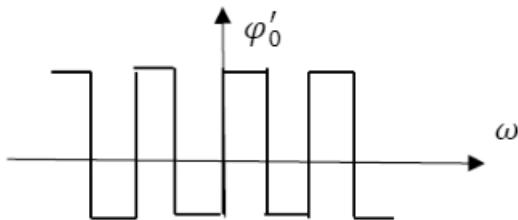
Đây chính là quá trình tính sự khác biệt của ma trận pha với các ma trận xung quanh để tính ra mức độ chênh lệch. Việc tính toán này sẽ đảm bảo sự khác biệt giữa các pha sẽ không quá lớn sau khi tiến hành biến đổi.

Bước 4: Điều chỉnh pha. Giá trị chính xác các pha của các đoạn có thể thay đổi nhưng mối liên hệ về sự khác nhau giữa các segment liên tiếp phải được đảm bảo.

Việc điều chỉnh pha của đoạn đầu được áp dụng dựa trên công thức:

$$Phase_new = \begin{cases} \frac{\pi}{2} & \text{nếu message bit} = 0 \\ -\frac{\pi}{2} & \text{nếu message bit} = 1 \end{cases}$$

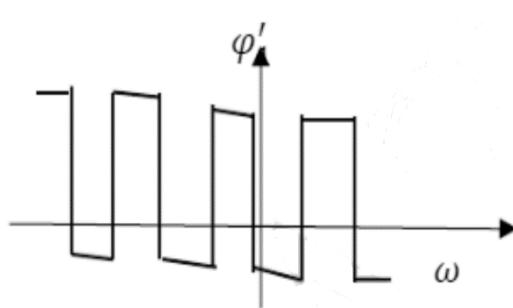
Khi đó thông tin giấu chỉ được phép giấu trong vector pha của đoạn đầu tiên



Hình 2.45: Tín hiệu được giấu trong pha của đoạn đầu tiên

Bước 5: Tiến hành tạo ma trận pha mới thỏa mãn để căn chỉnh lại độ chênh lệch tính ra ở bước 3. Tạo ma trận pha mới thỏa mãn điều kiện:

$$\varphi'_{j+1}[\omega_k] = \Delta\varphi_{j+1}[\omega_k] + \varphi'_j[\omega_k] \quad \forall j, k$$



Hình 2.46: Ma trận với pha mới được tạo

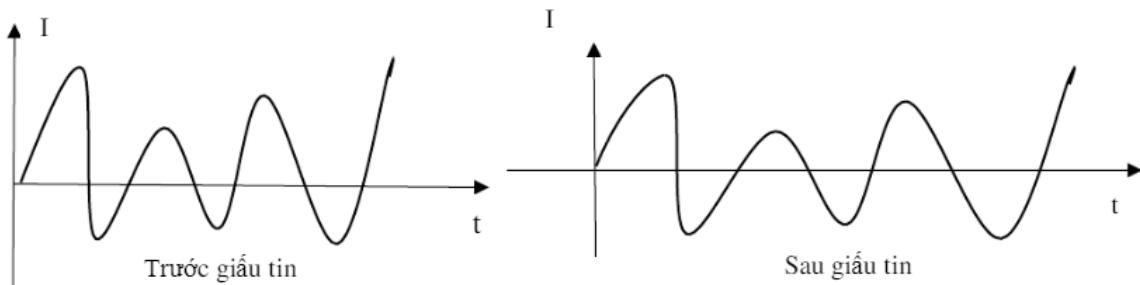
Trong thực tế luôn tìm được cặp $\varphi'_{j+1}[\omega_k] + \varphi'_j[\omega_k]$ thỏa mãn công thức Fourier rời rạc có tính đầy đủ (với mọi $N > 0$, mọi vecto phức N chiều đều có một DFT và một IDFT đồng thời DFT và IDFT đều là các vecto phức nhiều chiều). So sánh 2 hình trên thấy được là ma trận pha mới tạo đã có sự thay đổi so với ma trận pha ban đầu.

Bước 6: Kết hợp với cường độ pha của tín hiệu cũ sau khi đã giấu thông tin. Mục đích của bước này chính là tái tạo lại ma trận pha của các đoạn kề nhau. Pha mới bằng pha kề trước đó cộng với độ lệch pha đã được tính ở trên



Hình 2.47: Pha mới được tạo ra sau khi kết hợp cường độ của pha cũ

Bước 7: Thực hiện ghép các segment lại và DFT ngược để tạo lại dữ liệu âm thanh. Để nhận được tin giấu bằng kỹ thuật này, người nhận phải biết độ dài của segment, sau đó thực hiện DFT để nhận tin.



Hình 2.48: So sánh pha trước và sau khi giấu tin

Từ hình trên thấy được rằng: rõ ràng âm thanh đã bị thay đổi về cấu trúc pha khi giấu thông tin vào trong âm thanh.

Đánh Giá

- Ưu điểm:

- Như đã đề cập ở trên, mã hóa pha với thay đổi đủ nhỏ sẽ không bị phát hiện bởi giác quan của con người do hệ thính giác không nhạy cảm với sự thay đổi của pha âm thanh
- Mã hóa pha không gây nhiễu như các phương pháp với LSB hoặc các phương pháp khác

- Nhược điểm

- Lượng thông tin được giấu nhỏ vì phương pháp mã hóa pha chỉ giấu được thông tin trên một đoạn nhỏ của file âm thanh. Nếu muốn tăng lượng thông tin được giấu thì có thể kéo dài thêm đoạn của âm thanh gốc, tuy vậy việc đó ít được thực hiện bởi nếu vậy khả năng bị phát hiện tin được giấu trong file âm thanh sẽ lớn hơn.
- Khả năng ứng dụng bị hạn chế: Ví dụ nếu sử dụng mã hóa pha để giấu tin trong file âm thanh, file đó có thể dễ dàng bị tấn công và phát hiện do thông tin mật chỉ ở đầu của file âm thanh.
- Thời gian nạp âm thanh tương đối lâu, trong khi chỉ có khối đầu tiên được nhúng thông tin, dữ liệu giấu không được phân bố đều trên toàn bộ tín hiệu âm thanh, sử dụng tài nguyên không hiệu quả.

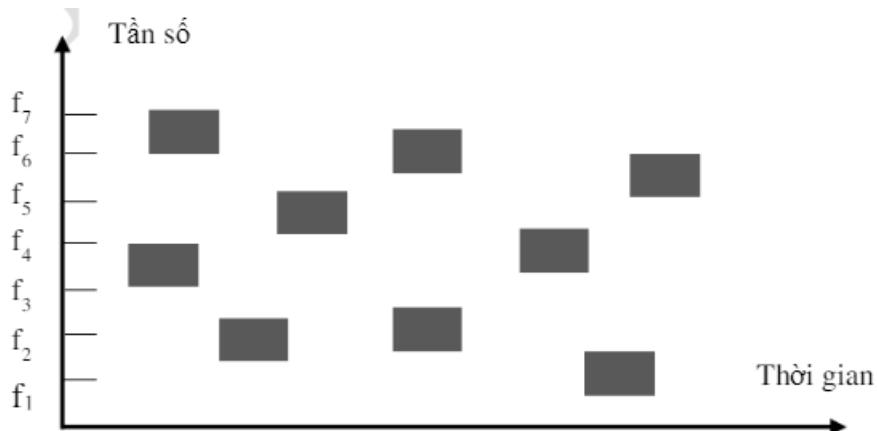
2.2.5.3. Phương pháp trai phô

Trai phô (*Spread Spectrum*) là kỹ thuật truyền tín hiệu được sử dụng rộng rãi trong truyền thông. Trong đó năng lượng của tín hiệu được “trai” trên một băng thông rộng hơn nhiều lần so với lượng băng thông cần thiết tối thiểu nhờ sử dụng mã giả ngẫu nhiên, mã này độc lập với tín hiệu thông tin. Bên nhận thông tin sẽ tiến hành giải trai bằng cách đồng bộ hóa mã giả ngẫu nhiên. Tín hiệu trai phô trông giống như nhiễu, khó phát hiện và thậm chí khó để chặn đứng hay giải điều chế (demodulation) nếu không có các thiết bị thích hợp. Các kỹ thuật trai phô cố gắng trai thông tin mật vào trong phô tần số của dữ liệu âm thanh càng nhiều càng tốt. Nó cũng tương tự như kỹ thuật LSB là trai ngẫu nhiên thông tin giàu trên toàn bộ file âm thanh. Như vậy, một hệ thống thông tin được coi là hệ thống trai phô khi tín hiệu được phát có độ rộng băng tần lớn hơn nhiều so với độ rộng băng tần tối thiểu cần thiết và quá trình trai phô được sử dụng bằng một mã giải độc lập

Có 2 loại chính của phương pháp trai phô chính là:

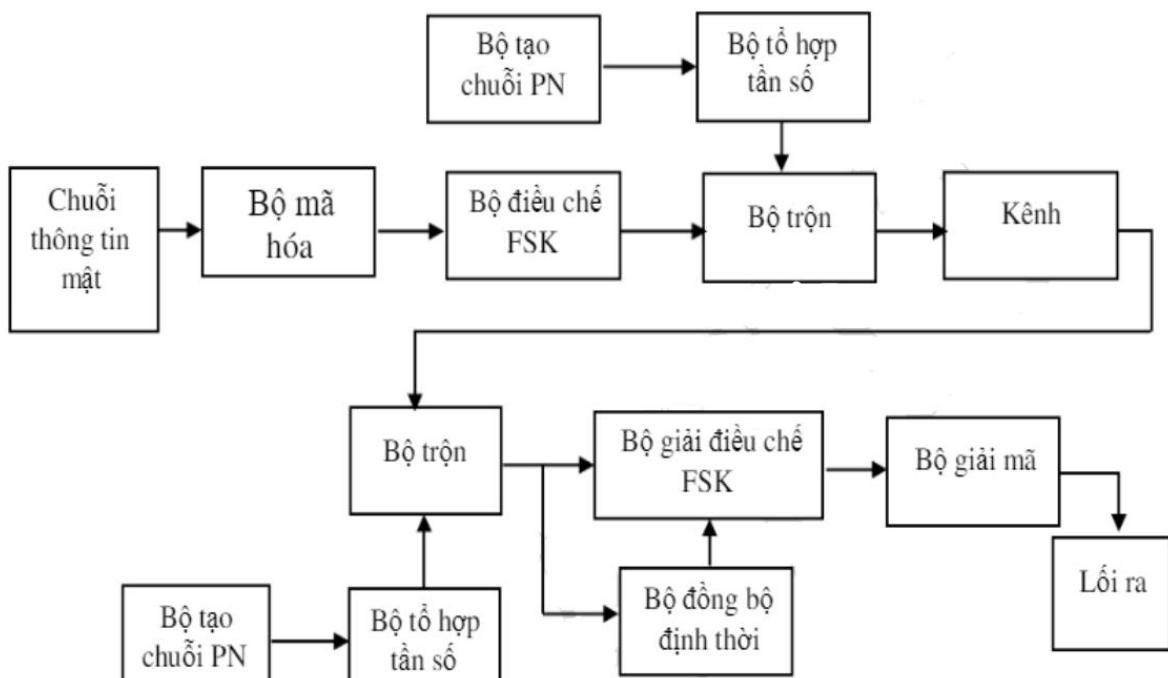
2.2.5.3.1. Trai phô nhảy tần (Frequency Hopping Spread Spectrum- FHSS)

Trai phô nhảy tần là một công nghệ sử dụng bộ phát tần số và có thể thay đổi tần số truyền một cách đột ngột trong dãy băng tần sử dụng. Trong trai phô nhảy tần, độ rộng băng kênh sẵn có sẽ được chia thành một số lớn các khe tần không lấn lên nhau. Tại bất kì khoảng thời gian nào, tín hiệu truyền đi đều chiếm một hay nhiều hơn một khe tần số nói trên. Việc chọn một khe khe hay nhiều khe tần số trong một khoảng thời gian truyền tín hiệu đều được thực hiện một cách giả ngẫu nhiên theo tín hiệu lối ra của một bộ tạo chuỗi giả ngẫu nhiên. Hình dưới đây mô tả về quy trình trai phô nhảy tần.



Hình 2.49: Minh họa về trai phô nhảy tần

Dựa trên tốc độ nhảy của tần số thì phương pháp trai phô nhảy tần được chia làm 2 loại đó là trai phô nhảy tần nhanh (khi tốc độ nhảy nhanh hơn tốc độ dữ liệu) và trai phô nhảy tần chậm (khi tốc độ nhảy chậm hơn tốc độ dữ liệu). Nhìn chung thì cả hai phương pháp này chỉ khác nhau về tốc độ nhảy, còn nguyên lý hoạt động của hai phương pháp tương tự nhau. Hình sau mô tả về nguyên lý hoạt động của trai phô nhảy tần.

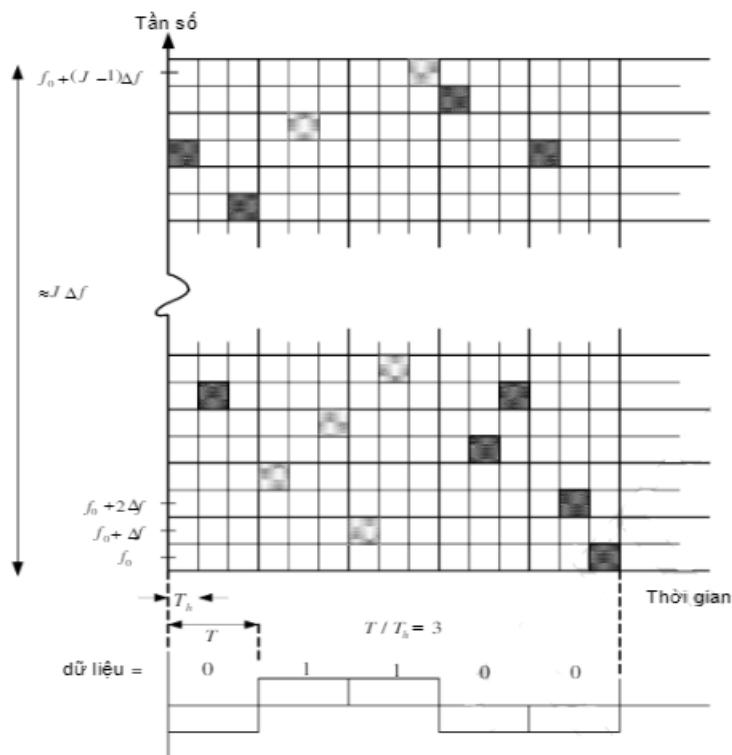


Hình 2.50: Sơ đồ khái của hệ thống trai phô FHSS

Quy Trình

- Ở phía máy phát: tín hiệu đầu vào của hệ thống trai phô nhảy tần bao gồm:
 - Chuỗi thông tin mật cần được truyền đi: Chuỗi thông tin này được đưa vào Bộ mã hóa. Tại đây, tín hiệu được mã hóa bằng khóa riêng trước khi được đưa vào

Bộ điều chế. Đây là bước tùy chọn, nghĩa là tùy người gửi tin cài đặt cho máy phát lựa chọn có mã hóa hay không, nếu có thì chọn kỹ thuật mã hóa nào. Ở một số loại máy phát đòi đầu thì không có bộ mã hóa này. Bước này có nhiệm vụ làm tăng tính bảo mật của thông tin trên đường truyền. Phương pháp giải mã và khóa bí mật sẽ được người gửi và người nhận thỏa thuận bằng một hình thức nào đó. Tín hiệu sau khi được mã hóa sẽ được đưa vào bộ điều chế FSK (điều chế số theo tần số tín hiệu). Tại đây, tín hiệu đã mã hóa sẽ được bộ FSK điều chế thành tín hiệu nhị phân $x(t)$. Trong mỗi bit $x(t)$ có một trong 2 tần số là $f' = (f' + (2k)\Delta f)$ và $(f' + \Delta f) = (f' + (2k + 1)\Delta f)$ tương ứng với bit dữ liệu 0 và bit dữ liệu 1, với $k \in N$. Bộ điều chế sẽ chọn một trong 2 tần số f' và $(f' + \Delta f)$ tương ứng với việc truyền đi bit dữ liệu 0 hay bit dữ liệu 1.



Hình 2.51: Biểu đồ tần số của tần nhanh với FSK

Ví dụ: Trên hình bên là biểu đồ tần số của nhảy tần nhanh với FSK. Trong đó T là độ dài bit dữ liệu, T_h là độ dài 1 lần nhảy. Ở ví dụ này, $T = 3T_h$. Δf là giãn cách tần số giữa 2 tần số lân cận. Đối với hệ thống nhảy tần nhanh, do sự thay đổi nhanh tần số sóng mang, giải điều chế liên kết (coherent) là không thực tế và giải điều chế không liên kết được sử dụng thay. Do đó Δf thường được chọn = $\frac{1}{T_h}$ nghĩa là sử dụng tập tín hiệu trực giao để cho chất lượng tín hiệu tốt hơn (xác suất lỗi bit ít hơn so với tập

không trực giao). Giả sử mỗi lần nhảy T_h giây, một trong j tần số được phát đi, tần số phát trong mỗi lần nhảy được chỉ bởi ô tô nhạt khi bít dữ liệu là 1 hoặc bởi ô tô đậm khi bít dữ liệu là 0. Khi di chuyển theo chiều ngang trên biểu đồ, có thể thấy rằng tần số phát thay đổi cứ mỗi T_h giây.

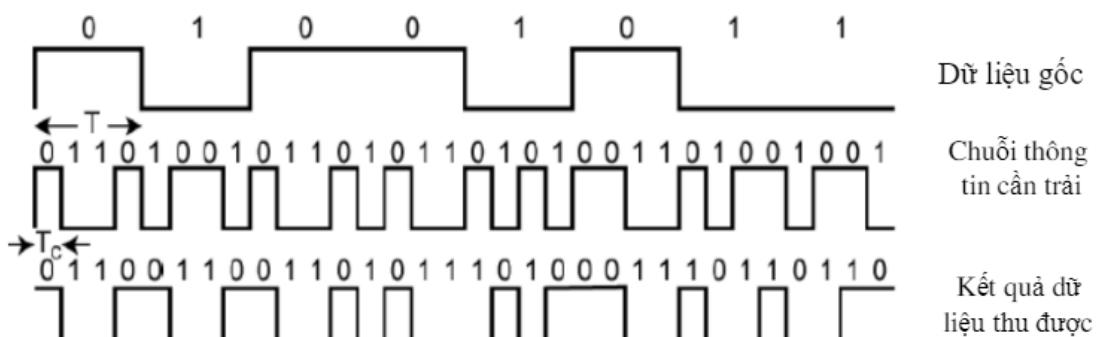
- Bộ tạo chuỗi PN: là một danh sách của nhiều tần số mà sóng mang có thể nhảy để chọn tần số truyền. Khi danh sách tần số đã nhảy hết, bên truyền sẽ lặp lại từ đầu danh sách này. Tại các thời điểm có sự nhảy tần số thì bộ tạo chuỗi giả ngẫu nhiên này tạo ra một đoạn chứa m bit của mã để điều khiển bộ tổng hợp tần số để tạo ra các giá trị tần số nhảy tần cho sóng mang. Ở đây, chuỗi giả ngẫu nhiên không nhất thiết phải là dãy nhị phân. Khác với hệ thống trai phổ trực tiếp, chuỗi giả ngẫu nhiên ở hệ thống trai phổ nhảy tần chỉ dùng để điều khiển hoặc xác định các mẫu nhảy. Sau khi tạo ra đoạn mã có độ dài m bit, đoạn mã này được gửi đến bộ tổ hợp tần số. Tại bộ tổ hợp tần số: Sau khi nhận được tín hiệu điều khiển từ bộ tạo chuỗi PN, bộ tổ hợp tần số tạo ra các giá trị tần số nhảy tần cho sóng mang và nhảy sang hoạt động ở một tần số tương ứng với đoạn mã m bit của mã đưa vào, gọi là $y(t)$. Ứng với m bit thì mã sẽ cho ra 2^m giá trị tần số khác nhau, đoạn m bit này được gọi là một từ tần số và có 2^m giá trị tần số khác nhau. Tần số $y(t)$ thay đổi cứ mỗi T giây theo các giá trị m bit từ bộ tạo chuỗi PN.
- Ở phía máy thu: tín hiệu từ kênh truyền sau khi thu về sẽ được đưa vào bộ trộn. Nhận được tín hiệu truyền về, bộ tạo chuỗi PN sẽ tạo nên chuỗi giả ngẫu nhiên đồng bộ với chuỗi tới (Bộ tạo chuỗi giả ngẫu nhiên ở phía máy phát và máy thu là như nhau và được đồng bộ với nhau giao tiếp với bộ tổ hợp tần số ở phía phát và phía thu.). Chuỗi giả ngẫu nhiên sau khi được tạo ra sẽ được gửi đến bộ tổ hợp tần số để tạo ra các giá trị nhảy tần cho sóng mang, điều khiển lối ra của bộ này. Tín hiệu tần số được tạo ra từ bộ tổ hợp tần số được gửi đến bộ trộn. Tại đây, tín hiệu thu về từ kênh truyền sẽ được trộn với tín hiệu lối ra của bộ tổ hợp tần số, dựa theo dài tần lọc của bộ lọc BPF mà thu được tín hiệu $x(t)$. Tín hiệu này được gửi đồng thời cho bộ giải điều chế FSK và bộ đồng bộ định thời. Tín hiệu sau khi được đưa qua bộ trộn thì được đồng bộ về mặt thời gian tại Bộ đồng bộ định thời. Kết quả tín hiệu sau khi được đồng bộ thời gian được gửi cho bộ giải điều chế FSK. Tại đây, tín hiệu sóng mang $x(t)$ được đưa vào bộ giải điều chế FSK để

tái tạo lại dữ liệu trước khi bị mang đi điều chế. Dữ liệu sau khi được giải điều chế được đưa vào bộ giải mã để giải mã, khôi phục lại dữ liệu gốc ban đầu

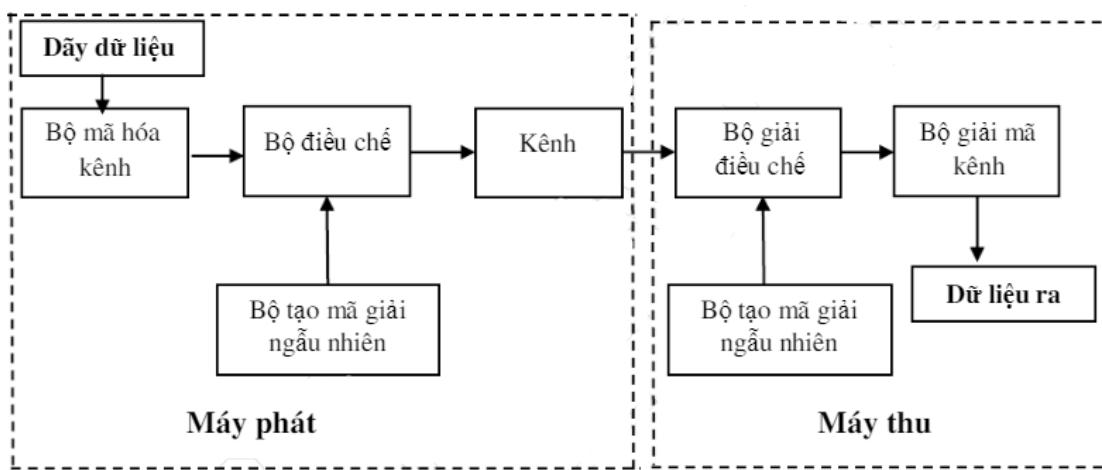
Nhận xét: Tính hiệu quả của phương pháp trai phổ nhảy tần chính là bên nhận và bên gửi sẽ phải thống nhất với nhau chuỗi giả ngẫu nhiên để thu được thông tin một cách chính xác.

2.2.5.3.2. Trai phổ dãy trực tiếp (Direct Sequence Spread Spectrum - DSSS)

DSSS là hệ thống trai phổ dãy trực tiếp, rất phổ biến và được sử dụng rộng rãi trong các công nghệ trai phổ vì nó dễ dàng cài đặt và có tốc độ cao. DSSS là một phương pháp truyền dữ liệu trong đó hệ thống truyền và hệ thống nhận đều sử dụng một tập các tần số có độ rộng 22 MHz. Các kênh rộng này cho phép các thiết bị truyền thông tin với tốc độ cao hơn hệ thống FHSS nhiều.



Hình 2.52: Minh họa trai phổ dãy trực tiếp



Hình 2.53: Sơ đồ khái niệm hệ thống trai phổ DSSS

Quy Trình

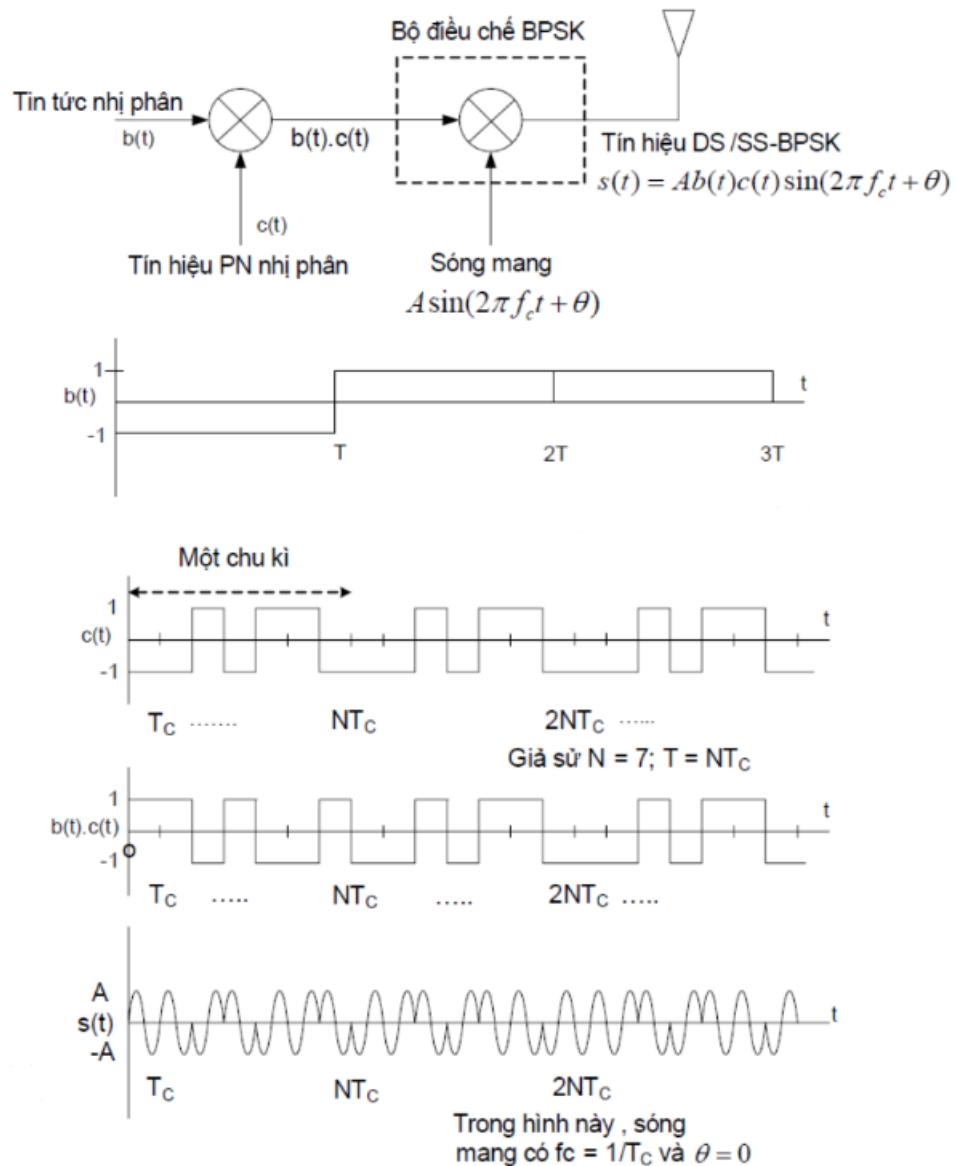
- Ở phía máy phát: tín hiệu đầu vào của hệ thống trai phô trực tiếp là:

- Dãy dữ liệu dạng nhị phân (tín hiệu cần trai phô). Tín hiệu này được đưa vào Bộ mã hóa kênh (còn gọi là bộ lập mã hiệu chỉnh lỗi hay bộ mã hóa sửa sai). Tại đây, tín hiệu đầu vào được mã hóa bằng bộ mã hóa kênh để đưa vào các bit dư nhằm mục đích phát hiện hay sửa các lỗi có thể phát sinh khi truyền dẫn tín hiệu qua kênh tần số vô tuyến. Tín hiệu sau khi được mã hóa bằng bộ mã hóa kênh được đưa vào bộ điều chế.
- Bộ tạo mã giả ngẫu nhiên tạo ra một chuỗi giả ngẫu nhiên dạng nhị phân, chuỗi này được đưa vào bộ điều chế để trai tín hiệu được phát đi về phô. Trên thực tế, hai bộ tạo mã giả ngẫu nhiên từ bên phía máy phát và máy thu phải như nhau và được đồng bộ với nhau giao tiếp với bộ điều chế và giải điều chế

Tại bộ điều chế có hai quá trình diễn ra đó là quá trình trai phô và quá trình điều chế sóng mang. Trong một số tài liệu quá trình trai phô và điều chế có thể bị tách riêng (không cùng nằm trong bộ điều chế) và thứ tự thực hiện trước sau không đồng nhất. Tuy nhiên trên thực tế thì hai quá trình này thường được kết hợp và thực hiện ở một khối duy nhất, thứ tự thực hiện có thể tráo đổi cho nhau, việc này không làm ảnh hưởng đến kết quả của tín hiệu đầu ra. Vì vậy, tại đây cả hai quá trình được đặt trong bộ điều chế.

Giả sử quá trình trai phô được thực hiện trước quá trình điều chế sóng mang, quy trình tiền xử lý tín hiệu sẽ được diễn ra như sau: Sau khi nhận được tín hiệu đã được mã hóa từ bộ mã hóa kênh và chuỗi giả ngẫu nhiên từ bộ tạo mã giả ngẫu nhiên, bộ điều chế sẽ thực hiện nhân hai tín hiệu này với nhau. Quá trình nhân hai tín hiệu với nhau thực chất là quá trình trai phô. Kết quả là phô của tín hiệu nhận được được trai ra trên dải không mong muốn dựa vào chuỗi giả ngẫu nhiên. Sau đó phô của tín hiệu được dịch đến dải tần phát được gán theo phương pháp BPSK hoặc QPSK (đây là quá trình điều chế sóng mang theo phương pháp BPSK hoặc QPSK). Kết quả là tín hiệu trai phô sau khi được điều chế sóng mang thì được đưa lên kênh truyền dẫn. Hình sau mô tả ví dụ minh họa cho bộ điều chế BPSK. Trong ví dụ này, quá trình trai phô được diễn ra trước quá trình điều chế sóng mang. Tín hiệu sau khi điều chế sẽ được phát qua kênh truyền dẫn,

kênh này có thể là kênh dưới đất hoặc kênh vệ tinh. Kênh này có thể gây ra giảm chất lượng như: nhiễu, tạp âm, suy hao công suất tín hiệu.



Hình 2.54: Bộ điều chế BPSK

- Ở phía máy thu: tín hiệu sau khi được lấy trên kênh truyền dẫn về sẽ được đưa vào bộ giải điều chế. Nhận được tín hiệu truyền về, bộ tạo mã giả ngẫu nhiên sẽ tạo nên chuỗi giả ngẫu nhiên đồng bộ với chuỗi tới. Chuỗi giả ngẫu nhiên sau khi được tạo ra sẽ được gửi đến bộ giải điều chế để giải trai phô cho tín hiệu thu được từ kênh truyền. Tín hiệu sau khi được giải trai phô sẽ được giải điều chế sóng mang bằng phương pháp BPSK hoặc QPSK để thu được tín hiệu băng gốc. Tín hiệu băng gốc này sẽ được truyền đến bộ giải mã kênh để giải mã, lấy ra tín hiệu gốc.

Như vậy, tại đầu thu, máy thu cố gắng khôi phục lại tín hiệu gốc bằng cách khử các quá trình sử dụng ở máy phát. Chú ý rằng bộ nén/giải nén dữ liệu và bộ mã sửa sai/giải mã là tùy chọn. Chúng dùng để cải thiện chất lượng hệ thống.

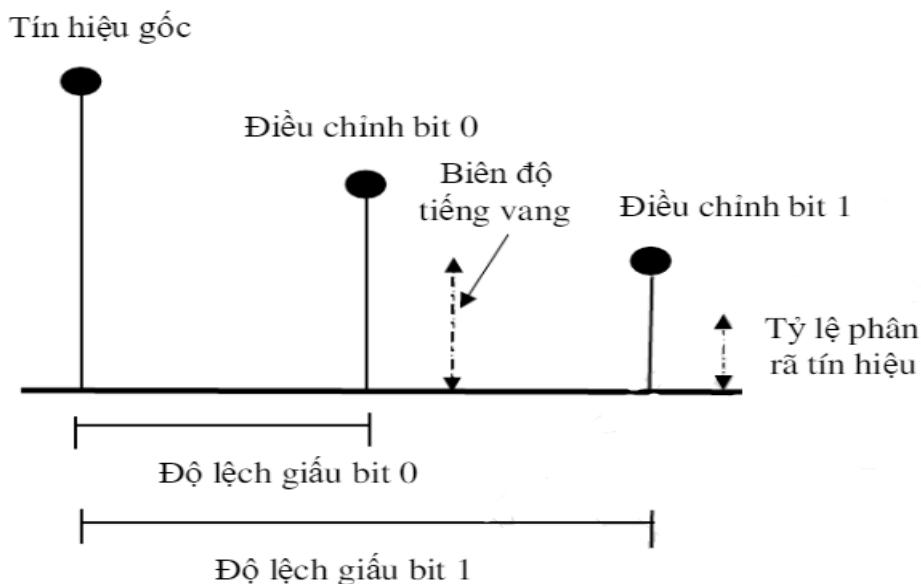
Nhận xét: Với việc giấu thông tin mật áp dụng phương pháp trai phỏ DSSS thì trai tín hiệu mật ra bằng một hằng số gọi tốc độ chip. Đồng thời điều chỉnh độ dài tối đa của tín hiệu giả ngẫu nhiên và thêm vào phương tiện chửa. Phổ của thông tin mật được trai rộng làm cho chuỗi thông tin mật giảm dần và được thêm vào phương tiện chửa như là thêm nhiều ngẫu nhiên. Quá trình thu nhận phỏ để tách thông tin thì người nhận cần phải biết điểm bắt đầu và kết thúc của dữ liệu được trai phỏ, tốc độ chip, tốc độ dữ liệu.

Kết luận: Mỗi hệ thống trai phỏ có những ưu và nhược điểm riêng. Và việc lựa chọn hệ thống nào để sử dụng còn phụ thuộc vào ứng dụng cụ thể. Nếu như DSSS làm giảm công suất nhiễu bằng cách trai nó trên phổ tần rộng thì FHSS tại thời điểm bắt kì đã cho người dùng khác nhau phát các tần số khác nhau vì thế tránh được nhiễu. Các kỹ thuật trai phỏ hiện nay đang được ứng dụng rất rộng rãi và đặc biệt là kỹ thuật này đang được sử dụng trong nhiều ứng dụng mới, như Mạng thông tin cá nhân (Personal Communication Networks – PCN), WLAN (Wireless Local Area Networks), Tổng đài nhánh cá nhân vô tuyến (Wireless Private Branch Exchanges – WPBX), các hệ thống điều khiển kiểm kê vô tuyến, các hệ thống báo động trong tòa nhà và hệ thống định vị toàn cầu (Global Positioning System - GPS). Các công nghệ ứng dụng kỹ thuật trai phỏ cung cấp các khả năng:

- Khả năng chống lại nhiễu cối ý và không cối ý – đặc điểm quan trọng đối với thông tin trong các vùng đông đúc như thành phố
- Có khả năng loại bỏ hoặc giảm nhẹ ảnh hưởng của truyền lan đa đường, có thể là vật cản lớn trong thông tin thành phố
- Có thể chia sẻ cùng bằng tần với các người dùng khác nhờ tính chất tín hiệu giống như tạp âm của nó
- Có thể dùng cho thông tin vệ tinh đã cấp phép trong chế độ CDMA; Cho mức độ riêng tư nhất định nhờ dùng các mã trai giả ngẫu nhiên làm cho nó khó bị nghe trộm.

2.2.5.4. Phương pháp Echo

Kỹ thuật giấu tin bằng phương pháp Echo (tiếng vang) được thực hiện bằng cách thêm tiếng vang vào trong tín hiệu gốc. Dữ liệu nhúng sẽ thay đổi 3 tham số của tiếng vang là biên độ ban đầu, tỉ lệ phân rã và độ trễ. Khi thời gian giữa tín hiệu gốc và tiếng vang giảm xuống lúc đó hai tín hiệu có thể trộn lẫn làm người nghe không thể phân biệt hai tín hiệu. Ngoài ra, số lượng tin giấu còn liên quan đến thời gian trễ của tiếng vang và biên độ của nó.



Hình 2.55: Các tham số chính trong phương pháp mã hóa tiếng vang

Các tham số chính trong quy trình giấu thông tin trong âm thanh bằng phương pháp mã hóa tiếng vang gồm:

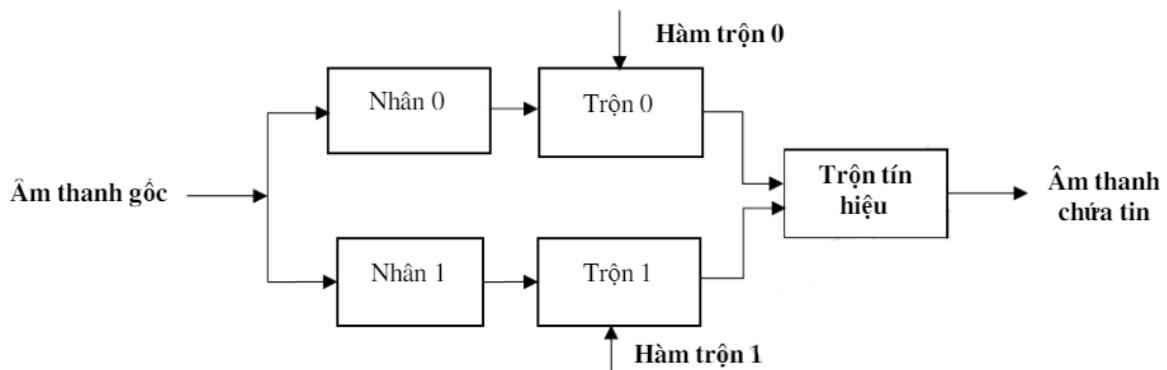
- Tín hiệu gốc.
- Tỷ lệ phân rã (Tốc độ phân rã).
- Độ trễ giữa âm thanh ban đầu và tiếng vang.

Cụ thể với phương pháp này thông tin được giấu trong một tín hiệu rác $f(t)$ bằng cách thêm tiếng vang $f(t - \Delta t)$ vào tín hiệu chứa $c(t)$:

$$c(t) = f(t) + \alpha f(t - \Delta t)$$

Thông tin được mã hóa thành các tín hiệu bằng cách hiệu chỉnh khoảng thời gian Δt . Δt là khoảng thời gian dừng giữa tín hiệu phát và tiếng vang. Tại bước mã hóa, người gửi có thể chọn các giá trị Δt và Δt tương ứng với các bit 0 hoặc 1 được nhúng. Các giá trị này được chọn sao cho tín hiệu tiếng vang không gây ra bất kỳ sự nghi ngờ nào tới cho người nghe.

Trong một số bài toán có thể chỉ cần thêm một tiếng vang vào tín hiệu gốc để giàu tin. Tuy nhiên, trong các phương pháp điều chỉnh tiếng vang cải tiến thì có thể thêm nhiều tiếng vang. Tín hiệu vang có thể là vang trước và vang sau so với tín hiệu gốc để giàu tin. Ví dụ trong đề xuất phương pháp thêm tiếng vang cả trước và sau so với tín hiệu gốc như công thức trên



Hình 2.56: Sơ đồ tổng quát phương pháp mã hóa tiếng vang

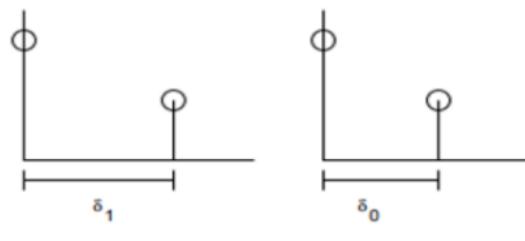
Từ sơ đồ tổng quát cho thấy các tham số chính trong quy trình giàu tin sử dụng phương pháp mã hóa tiếng vang gồm:

- Tín hiệu ban đầu
- Nhân hệ thống mã hóa
- Tín hiệu trộn

Dựa trên các thành phần chính trong sơ đồ tổng quát của phương pháp mã hóa tiếng vang có thể xây dựng quy trình giàu tin sử dụng phương pháp mã hóa tiếng vang như sau:

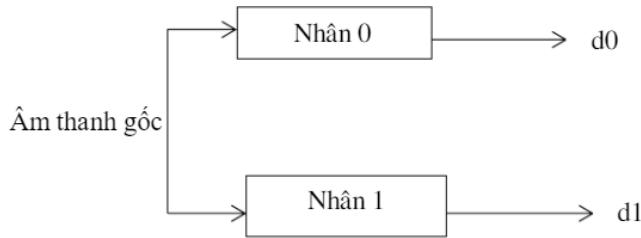
Bước 1: Tín hiệu ban đầu là tệp âm thanh gốc có dạng là hàm rời rạc theo thời gian $F(t)$. Tín hiệu ban đầu được xác định dựa vào hàm $F(t)$, từ tín hiệu ban đầu này để tìm ra được tiếng vang.

Bước 2: Nhân hệ thống mã hóa: Sử dụng nhân 0 và nhân 1 kết hợp với tín hiệu gốc để tạo ra tiếng vang tương tự tín hiệu gốc nhưng trễ hơn.



Hình 2.57: Nhân 0 và nhân 1

Nhân 0 có độ trễ là δ_0 và nhân 1 có độ trễ là δ_1 , dựa vào độ trễ để xác định tiếng vang so với tín hiệu ban đầu. Nhân 0 để mã hóa bit 0, nhân 1 để mã hóa bit 1.



Hình 2.58: Đầu vào và đầu ra bước 2

Kết quả thu được là hai đường tiếng vang $d0$ và $d1$ có dạng:

$$d(t) = F(t) + \beta F(t + \Delta t)$$

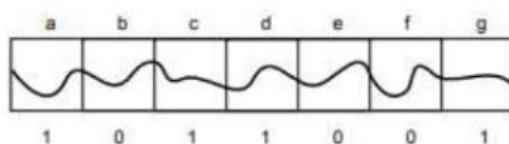
Trong đó

$F(t)$ là hàm rời rạc theo thời gian

β là tỷ lệ phân rã

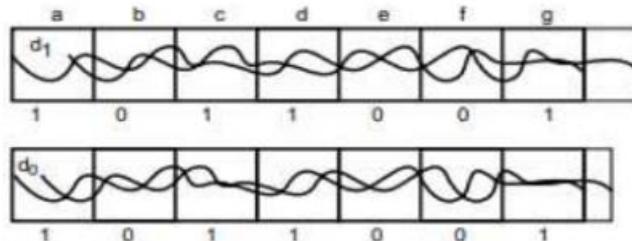
Δt là độ trễ của echo so với âm thanh gốc

Để mã hóa nhiều hơn một bit, âm thanh ban đầu được chia thành từng phần nhỏ hơn. Giả sử phải giấu N bit vào âm thanh, L là chiều dài của đoạn, L được chọn sao cho $N*L$ không lớn hơn độ dài của tín hiệu âm thanh. Mỗi phần có thể được lặp lại với các bit mong muốn bằng cách xem xét mỗi phần như một tín hiệu độc lập. Âm thanh sau khi được giấu tin sẽ là tái kết hợp của tất cả các tín hiệu mã hóa độc lập. Để nối hai đoạn mã hóa khác nhau sử dụng tín hiệu trộn 0 hoặc 1. Ví dụ: tín hiệu được chia thành 7 phần a, b, c, d, e, f, g.



Hình 2.59 Ví dụ giấu bit 0 và bit 1

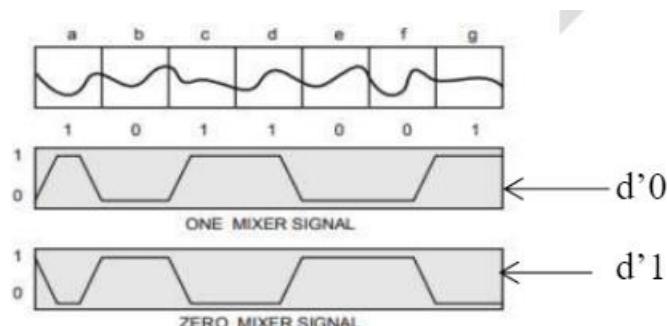
Thấy rằng: Các phần a, c, d, g chứa các bit 1 phần còn lại chứa bits 0. Theo lý thuyết kỹ thuật mã hóa tiếng vang sẽ mã hóa từng phần và sử dụng từng loại nhân phù hợp với bit cần giải nhưng trong thực tế các chuyên gia đã mã hóa toàn bộ sử dụng nhân 0 hoặc nhân 1, nên kết quả sẽ thu được hai tiếng vang đó là d0 và d1.



Hình 2.60: Kết quả tiếng vang sử dụng nhân 0 và nhân 1

Bước 3: Từ kết quả của bước 2, khi này tiếng vang đã được chia thành các đoạn để chứa các bit cần giải. Tiếng vang được nhân với hàm trộn theo nguyên tắc: d0 được nhân với hàm trộn 0, d1 được nhân với hàm trộn 1. Tức là khi thu được tiếng vang ở bước 2, các tín hiệu này được đưa vào máy trộn riêng để cho ra tín hiệu trộn d'0 và d'1

Để thu được tín hiệu trộn d'0 và d'1 thì trong máy trộn sẽ tự động sinh ra tín hiệu sin khi tín hiệu muốn chuyển đổi được đưa vào. Kết quả tạo ra 2 tín hiệu trộn có dạng là các đường dốc, tín hiệu trộn 0 là phần bù của tín hiệu trộn 1.



Hình 2.61: Kết quả của hàm trộn

Bước 4: Kết hợp 2 tín hiệu trộn thu được tín hiệu mã hóa khi cộng 2 tín hiệu, những đoạn có giá trị bằng 1 là mã hóa bit 1, đoạn có giá trị 0 là mã hóa bit 0, những đoạn có giá trị nằm trong khoảng từ 0 đến 1 là đoạn chuyển tiếp giữa 2 đoạn mã hóa khác nhau (giữa 2 đoạn mã hóa khác nhau 0 và 1).

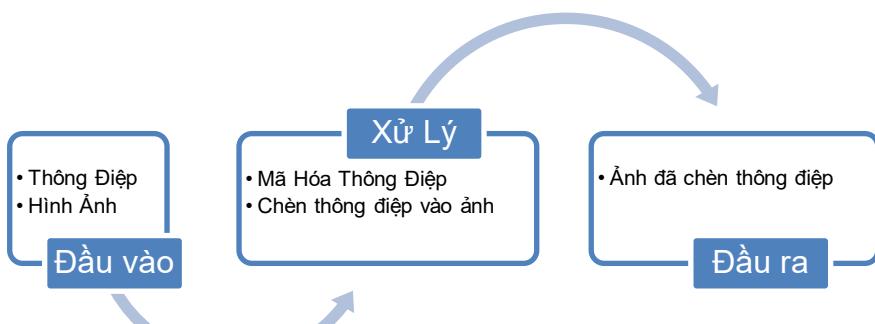
Lưu ý: Tổng giá trị của hai tín hiệu trộn luôn bằng 1, hai tín hiệu trộn này cộng lại với nhau bằng 1 nên có độ mịn chuyển đổi giữa các phần được mã hóa khác nhau và ngăn chặn thay đổi đột ngột trong cộng hưởng của tín hiệu cuối cùng.

PHẦN 3. THIẾT KẾ HỆ THỐNG

3.1. Tổng quan về hệ thống

Hệ thống steganography của nhóm em được thiết kế nhằm mục đích cung cấp một giải pháp bảo mật thông tin cao cấp. Cụ thể, trong tất cả các loại kỹ thuật giấu tin, nhóm em đã tìm hiểu và đưa ra quyết định chọn **hình ảnh** làm mục tiêu giấu tin với kỹ thuật LSB. Hệ thống sử dụng một thuật toán steganography tiên tiến, với kỹ thuật mã hóa số, để đảm bảo dữ liệu bí mật có thể được giấu kín mà không làm thay đổi đáng kể đến hình ảnh gốc.

Hệ thống được cấu trúc thành 3 thành phần chính: Mã hóa, Giấu tin, và Giải mã.



Hình 3.1: Sơ đồ hệ thống giấu tin trong ảnh

Tương tự, đối với việc giải mã thì ngược lại của quá trình trên. Trong đây, đầu vào chính là đầu ra của quá trình mã hóa.

3.2. Công nghệ và công cụ thiết kế

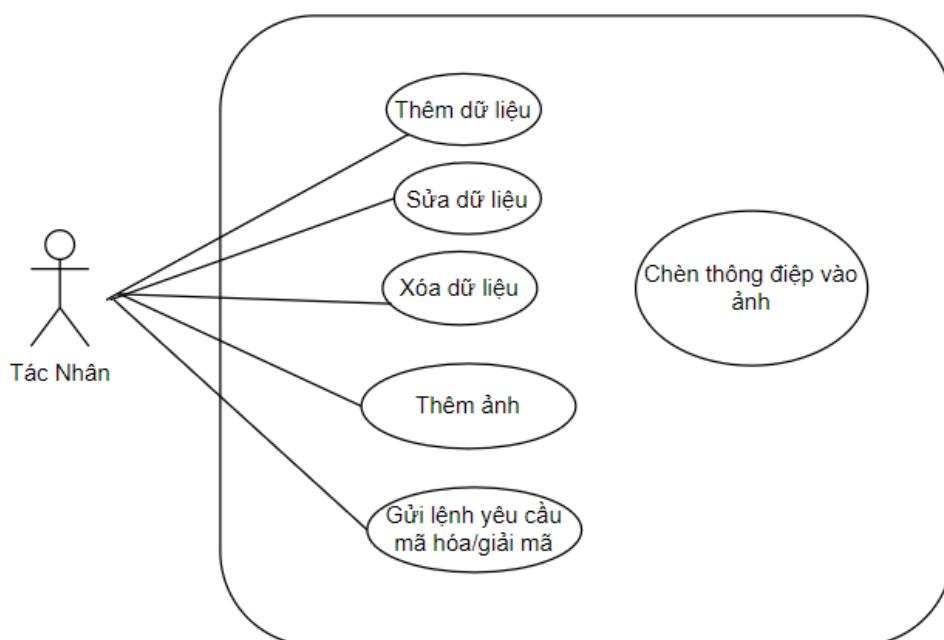
Trong hệ thống này, nhóm đã sử dụng:

- **Python:** Là ngôn ngữ lập trình chính cho toàn bộ hệ thống, thiết kế. Python là một ngôn ngữ có mã nguồn mở, và là ngôn ngữ lập trình hàng đầu nổi tiếng và phổ biến trên toàn thế giới, Python được hàng loạt doanh nghiệp lớn như Google, Apple và Microsoft ưa chuộng và sử dụng rộng rãi trong các hệ thống của mình. Sự ủng hộ mạnh mẽ từ cộng đồng lập trình viên và nhà phát triển khắp nơi trên thế giới đã

góp phần làm giàu thêm bộ thư viện phong phú của Python, giúp ngôn ngữ này càng trở nên mạnh mẽ và linh hoạt hơn.

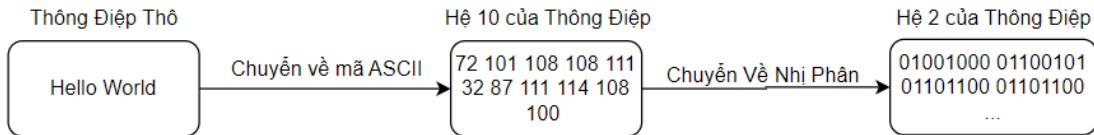
- **Pillow:** Thư viện hình ảnh Python là thư viện bổ sung mã nguồn mở và miễn phí dành cho ngôn ngữ lập trình Python, bổ sung hỗ trợ mở, thao tác và lưu nhiều định dạng tệp hình ảnh khác nhau.
- **Tkinter:** là thư viện GUI tiêu chuẩn cho Python. Tkinter trong Python cung cấp một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng cho bộ công cụ Tk GUI. Từ đó, giao diện người dùng có thể thao tác trên giao diện một cách dễ dàng hơn so với thao tác trên cửa sổ console
- **Visual Studio Code:** là trình soạn thảo, biên tập lập trình mã nguồn miễn phí được sử dụng trên 3 nền tảng đó là: Windows, macOS và Linux được xây dựng, phát triển bởi Microsoft. Visual Studio Code được các chuyên gia công nghệ thông tin đánh giá cao, nó là sự kết hợp hoàn hảo giữa IDE và CODE Editor. Nó hỗ trợ phần lớn đa số các ngôn ngữ lập trình hiện nay, dễ dùng, dễ cài đặt và cho phép tùy biến sâu trong ứng dụng. Hiệu quả công việc của nó là mang đến một hệ sinh thái mới vô cùng phong phú cho các ngôn ngữ lập trình.

3.3. Mô Tả Và Thiết Kế Hệ Thống



Hình 3.2: Biểu đồ use case tổng quát

- **Mã hóa:** Đầu tiên hệ thống cần nhận vào 2 đối tượng là **Thông Điệp** và **Hình Ảnh** cần giá trị tin (Thường là ảnh có .png). Hệ thống sẽ mã hóa thông điệp sang chuỗi nhị phân thông qua ký tự ASCII theo sơ đồ sau:



Hình 3.3: Sơ đồ biểu diễn mã hóa một chuỗi thành mã nhị phân tương ứng

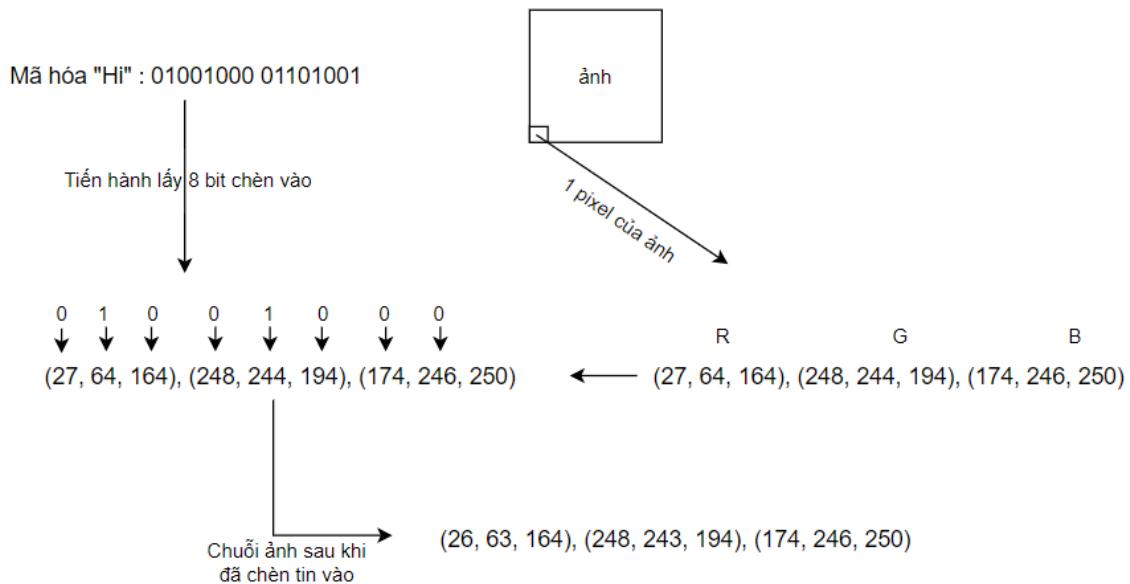
Hình ảnh được cấu tạo được cấu tạo từ hai thành phần chính là *header* và *data* (Ngoài ra còn nhiều hơn như Color Palette, ...) Trong đây, Header là phần đầu của một file ảnh chứa những thông tin của ảnh như kích thước ảnh, loại ảnh, metadata của ảnh. Và phần thứ hai là data của ảnh, nơi thực tế chứa dữ liệu của ảnh. Mỗi ảnh đều cấu tạo lên từ nhiều pixel, mỗi pixel gồm 3 màu (dùng RGB) theo chuỗi một nhóm gồm 3 chuỗi, mỗi chuỗi là biểu diễn từ 0 – 255 theo từng phần trong nhóm đó. Ví dụ trong RGB thì có 3 nhóm là R (red) – G (green) – B (blue). Biểu diễn từ 0 – 255 là độ đậm nhạt của của màu đó trong nhóm đó (vd: 255,0,0 – là màu đỏ rực) tất cả tạo nên 1 màu hòa trộn, nhiều pixel ghép với nhau tạo nên một bức ảnh hoàn chỉnh.

Ảnh nhận vào sẽ được trải dài các pixel của nó theo thứ tự, chuỗi chẵn khi đổi ra nhị phân sẽ có bit 0 ở cuối, tương tự như chuỗi lẻ thì cũng sẽ có bit 1 ở cuối (vd: 120,23,144 = 01111000 00010111 10010000)

Trong phần này, tác nhân có thể tác động trực tiếp vào hệ thống, cung cấp dữ liệu cho hệ thống cũng như tương tác với hệ thống, yêu cầu hệ thống cung cấp giải mã/mã hóa theo mong muốn

- **Giấu tin:** Như đã đề cập trước đó, nhóm đã sử dụng kỹ thuật LSB – Least Significant Bit cho hệ thống này. Kỹ thuật nhằm đến việc thay đổi các bit có trọng số nhỏ nhất, nghĩa là thay đổi các bit ở vị trí nào đó sao cho dữ liệu mới cho ra ít thay đổi nhất so với dữ liệu cũ. Vậy nên thay đổi chỉ tăng/giảm 1 đơn vị bit.

Cụ thể, chuỗi nhị phân đã mã hóa từ thông điệp của bước trên sẽ tiến hành cắt theo từng đoạn 8 bit. Mỗi đoạn 8 bit đó sẽ lần lượt giấu vào từng chuỗi trong ảnh.



Hình 3.4: Sơ đồ chèn bit từ thông điệp vào ảnh

Trong phần này, tác nhân không thể tác động vào hệ thống. Mọi hoạt động diễn ra đều tự động sử lý bên trong. Kết quả của bước này sẽ cho ra bức ảnh đã mã hóa với chất lượng tương đương ảnh gốc.

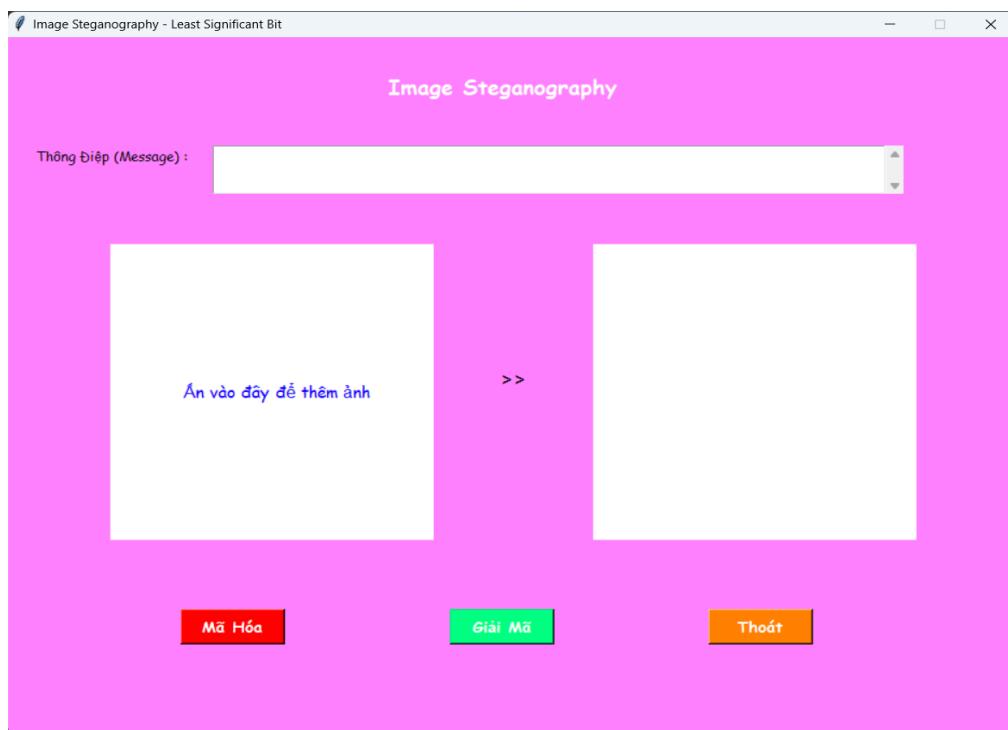
- **Giải mã:** Là quá trình ngược lại của mã hóa. Tại đây, hệ thống yêu cầu nhập ảnh có chứa thông điệp ẩn giấu để tiến hành trích xuất ra thông điệp.

Trong phần này, tác nhân cung cấp ảnh đã mã hóa cho hệ thống xử lý, ảnh được tự động trích xuất thông tin sau khi yêu cầu.

3.4. Xây Dựng Và Triển Khai Hệ Thống

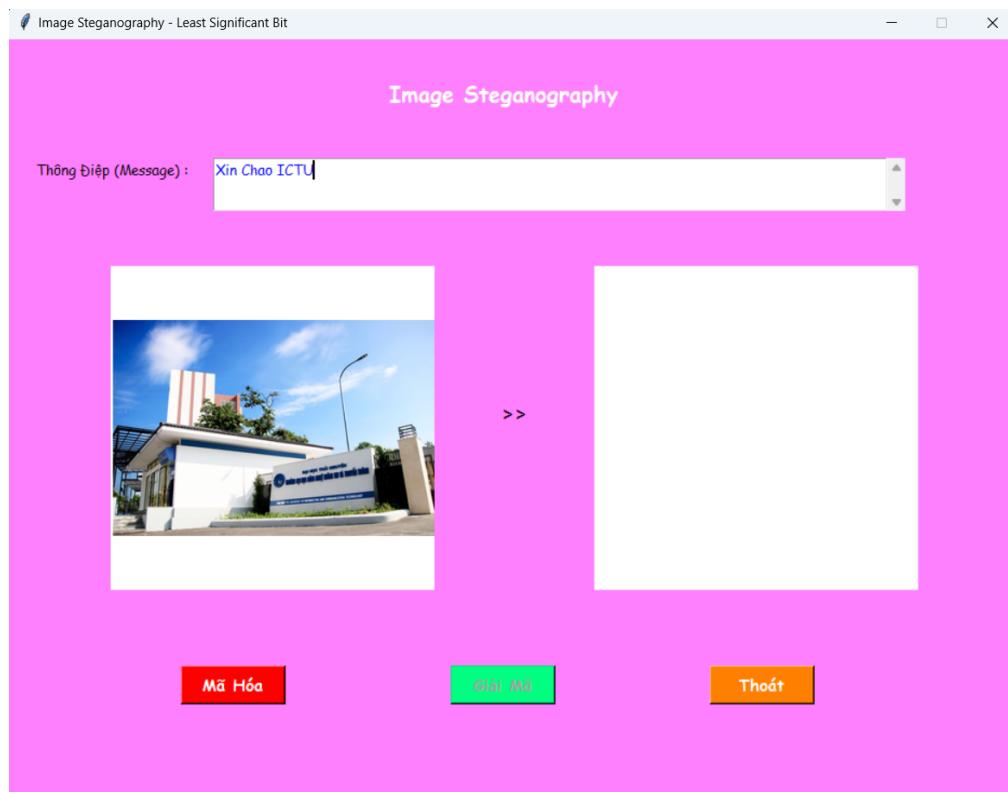
Sau khi nghiên cứu và chuẩn bị kỹ càng, nhóm đã bắt tay vào xây dựng chương trình. Sau đây là Demo sản phẩm:

MÃ HÓA



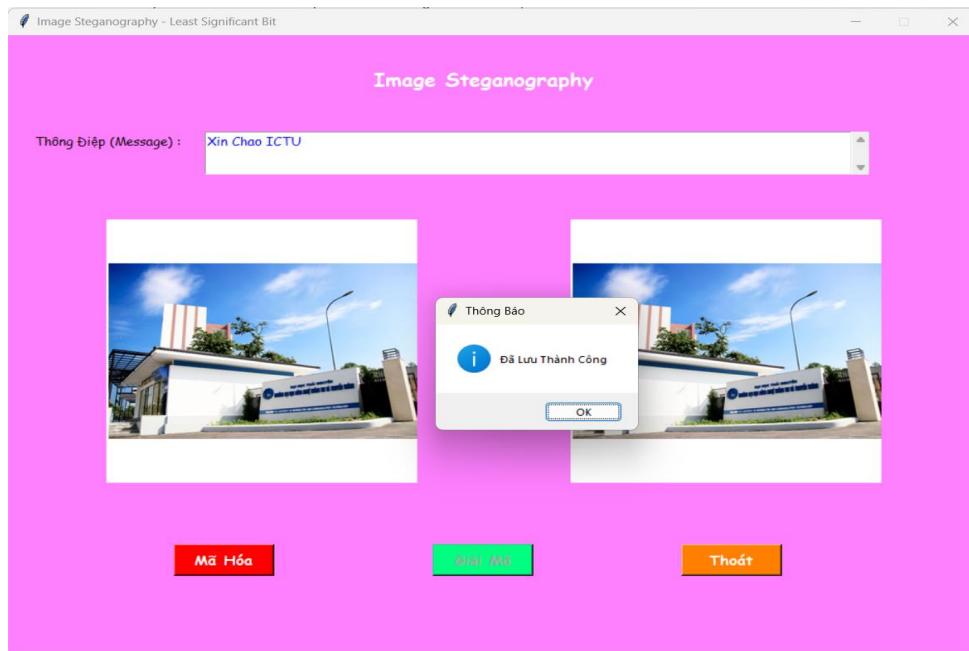
Hình 3.5: Giao diện chính

Thực hiện thêm ảnh và thông điệp



Hình 3.6: Demo chức năng thêm ảnh và thông điệp

Sau khi đã thêm, ấn “Mã Hóa” để mã hóa thông điệp vào ảnh, thực hiện chọn nơi lưu ảnh mới và đặt tên cho ảnh mới đó.

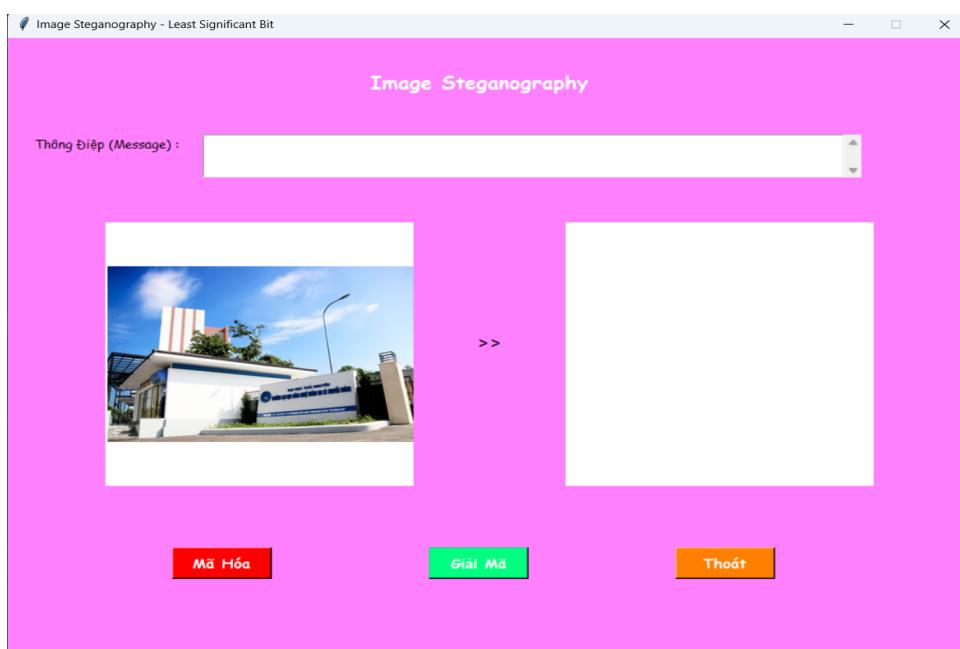


Hình 3.7: Demo mã hóa chèn thông điệp vào ảnh

Lúc này ảnh đã được lưu thành công vào đường dẫn mới và tên mới, đồng thời hiện ảnh đã mã hóa ở khung thứ hai bên tay phải. Kết thúc quá trình mã hóa.

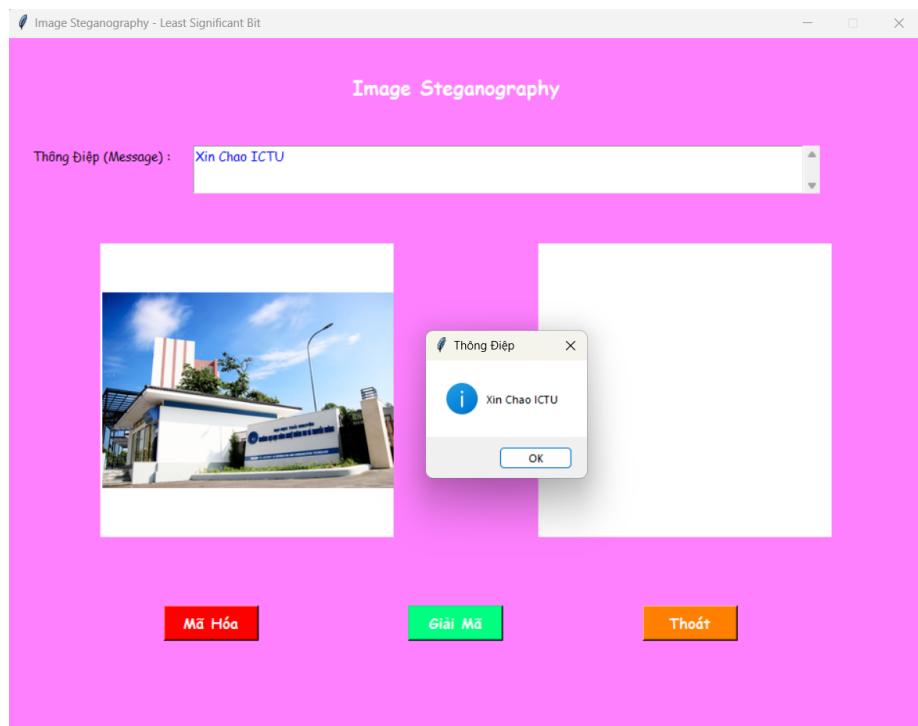
GIẢI MÃ

Thực hiện thêm ảnh cần được giải mã vào khung bên tay trái:



Hình 3.8: Demo thêm ảnh đã mã hóa

Ấn “Giải Mã” để tiến hành trích xuất thông điệp ẩn từ ảnh



Hình 3.9: Demo quá trình trích xuất thông điệp

Thông điệp lúc này hiện trên cửa sổ pop-up và đồng thời hiện lên thanh thông điệp. Kết thúc quá trình giải mã.

Chi tiết về source code và sản phẩm. Toàn bộ đính kèm theo đều chứa trong [này](#) hoặc quét QR CODE sau:



3.5. Kết Quả và Đánh Giá

Kết quả nhận được từ ảnh đã mã hóa và ảnh gốc:



Hình 3.10: *Ảnh gốc (bên trái) và ảnh đã chèn thông điệp (bên phải)*

Kết quả thử nghiệm cho thấy hệ thống có khả năng giữ tin hiệu quả trong hình ảnh với sự thay đổi tối thiểu đối với chất lượng hình ảnh nhìn bằng mắt thường. Cụ thể:

- ✓ Chất lượng hình ảnh: Trung bình, giá trị PSNR cho hình ảnh sau khi giữ tin là 48 dB, cho thấy sự thay đổi nhỏ về chất lượng hình ảnh. Giá trị SSIM trung bình là 0.98, chỉ ra rằng hình ảnh giữ được cấu trúc tương tự sau khi giữ tin.
- ✓ Khả năng chống phát hiện: Các kỹ thuật phân tích thông kê không thể phát hiện sự hiện diện của dữ liệu giấu trong hầu hết các trường hợp, chứng minh hiệu quả của phương pháp giấu tin.
- Các kết quả thu được từ hệ thống cho thấy một cân bằng tốt giữa việc bảo vệ thông tin và duy trì chất lượng hình ảnh. Tuy nhiên, cũng có một số hạn chế cần được cải thiện trong tương lai, bao gồm việc tăng dung lượng dữ liệu có thể giấu mà không làm giảm chất lượng hình ảnh và cải thiện khả năng chống lại các phương pháp phân tích nâng cao.

Tuy nhiên hệ thống vẫn còn nhiều thiếu sót, vẫn xảy ra một số lỗi đáng quan tâm. Cụ thể:

- Vẫn còn lỗi khi sử lý trên một số ảnh thuộc định dạng .jpg .jpeg .gif do header của nó không cố định, hoặc do ảnh quá chi tiết (thường có ở trên ảnh có 4 đoạn hex pixel với chiều thứ 4 là chiều sâu của ảnh)
- Đối với ảnh .png sẽ hoạt động tốt hơn tuy nhiên không phải tất cả
- Sản phẩm chưa thực sự ổn định, đặc biệt có một số lỗi vặt như (crash, infinity loop, sai kết quả, lỗi mã hóa/giải mã) nhưng khá hiếm
- Một số lỗi khác có thể do ảnh quá bé mà thông điệp lại quá dài gây vượt quá kích thước ảnh

3.6. Hướng Phát Triển Trong Tương Lai

Trong bối cảnh công nghệ và yêu cầu bảo mật thông tin ngày càng tăng, việc nâng cao hiệu suất và độ an toàn của hệ thống steganography là một nhiệm vụ quan trọng và liên tục. Dựa trên kết quả thu được và nhận định về hệ thống hiện tại, dưới đây là một số hướng phát triển chính để cải thiện và mở rộng khả năng của hệ thống:

- ✓ Khắc phục toàn bộ nhược điểm như đã đề cập ở trên của hệ thống.
- ✓ Tối Ưu Hóa Thuật Toán Giấu Tin, nâng cấp các thuật toán giấu tin mới có thể tăng cường khả năng chống phát hiện và tăng dung lượng dữ liệu giấu được mà không làm giảm chất lượng hình ảnh.
- ✓ Sử dụng AI để cung cấp gợi ý và tự động hóa quá trình giấu tin, làm cho hệ thống trở nên thông minh và linh hoạt hơn.
- ✓ Cải Thiện Độ Bảo Mật, kết hợp các phương pháp mã hóa mạnh mẽ hơn để bảo vệ dữ liệu giấu trước khi nó được chèn vào hình ảnh, đảm bảo an toàn thông tin ngay cả khi kỹ thuật steganography bị vượt qua
- ✓ Mở rộng khả năng hỗ trợ của hệ thống đối với nhiều định dạng hình ảnh khác nhau cũng như các thể loại khác như video, âm thanh,..., cũng như khám phá khả năng áp dụng trong video và âm thanh để tăng tính linh hoạt và ứng dụng rộng rãi.
- ✓ Cải thiện giao diện người dùng thân thiện và dễ sử dụng trên nhiều nền tảng, bao gồm web, điện thoại di động, và máy tính để bàn.