

**BÁO CÁO BÀI TẬP**

**MÔN:** CÁC KỸ THUẬT GIẤU TIN

*Chủ đề:* **Phương pháp giấu tin trong miền video nén dựa trên sự khác biệt năng lượng và phương pháp giấu tin trên miền nén của vieo chất lượng cao**

**Giảng viên:** TS. Đỗ Xuân Chợ

**Nhóm lớp**: 03

**Nhóm thực hiện: 01**

|  |  |
| --- | --- |
| **Họ và tên:** | **Mã sinh viên** |
| Nguyễn Tuấn Anh | B17DCAT009 |
| Lê Đức Thắng | B17DCAT164 |
| Lê Tiến Đạt | B17DCAT034 |
| Ngô Thùy Trang | B17DCAT190 |
| Bùi Đức Thắng | B17DCAT162 |
|  |  |

Hà Nội 2021

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 4](#_Toc73347493)

[DANH MỤC BẢNG BIỂU 5](#_Toc73347494)

[I. Tổng quan giấu tin trong video 6](#_Toc73347495)

[I.1. Giới thiệu về kỹ thuật giấu tin 6](#_Toc73347496)

[I.1.1. Sơ lược về kỹ thuật giấu tin 6](#_Toc73347497)

[I.1.2. Mô hình giấu tin 7](#_Toc73347498)

[I.1.3. Giấu tin trong dữ liệu đa phương tiện 9](#_Toc73347499)

[I.1.4. Một số ứng dụng cụ của kỹ thuật giấu tin 10](#_Toc73347500)

[I.2. Đặc điểm của giấu tin trong video 13](#_Toc73347501)

[I.3. Một số định dạng file Video 13](#_Toc73347502)

[I.4. Phân loại kỹ thuật giấu tin trong video 15](#_Toc73347503)

[I.4.1. Phân loại giấu tin trong video theo kỹ thuật giấu 15](#_Toc73347504)

[I.4.2. Phân loại giấu tin theo miền giấu 15](#_Toc73347505)

[I.4.3. Phân loại giấu tin theo mục đích 15](#_Toc73347506)

[II. Phương pháp giấu tin trong video 15](#_Toc73347507)

[II.1. Phương pháp giấu tin trong miền video nén dựa trên sự khác biệt năng lượng 15](#_Toc73347508)

[II.1.1. Tổng quan 15](#_Toc73347509)

[II.1.2. Quy trình giấu tin 16](#_Toc73347510)

[II.2. Phương pháp giấu tin trên miền nén của video chất lượng cao 27](#_Toc73347511)

[III. Cài đặt và kiểm nghiệm 33](#_Toc73347512)

[III.1. Môi trường cài đặt 33](#_Toc73347513)

[III.2. Sơ đồ thuật toán 34](#_Toc73347514)

[III.2.1. Thuật toán giấu tin 34](#_Toc73347515)

[III.2.2. Thuật toán lấy tin 35](#_Toc73347516)

[III.3. Cài đặt thuật toán 36](#_Toc73347517)

[III.3.1. Kiểm tra định dạng .avi 36](#_Toc73347518)

[III.3.2. Chuyển sang chunk tiếp theo 36](#_Toc73347519)

[III.3.3. Ghi dữ liệu vào chunk 37](#_Toc73347520)

[III.3.4. Đọc dữ liệu từ chunk 37](#_Toc73347521)

[III.3.5. Giấu tin 38](#_Toc73347522)

[III.3.6. Lấy tin 38](#_Toc73347523)

[III.4. Cài đặt giao diện 39](#_Toc73347524)

[III.4.1. Giao diện giấu tin 39](#_Toc73347525)

[III.4.2. Giao diện lấy tin 40](#_Toc73347526)

[III.4.3. Giao diện giấu tập tin 41](#_Toc73347527)

[III.4.4. Giao diện lấy tập tin 42](#_Toc73347528)

[III.4.5. Giao diện phân tích giấu tin 43](#_Toc73347529)

[III.5. Kiểm nghiệm, đánh giá và phân tích 43](#_Toc73347530)

[III.5.1. Kiểm nghiệm 43](#_Toc73347531)

[III.5.2. Đánh giá 43](#_Toc73347532)

[III.5.3. Phân tích 44](#_Toc73347533)

[IV. Kết luận 48](#_Toc73347534)

[Tài liệu tham khảo 49](#_Toc73347535)

# DANH MỤC HÌNH ẢNH

[Hình 1: Mô hình chung cho quá trình giấu và tách tin 8](#_Toc73347398)

[Hình 2: Sơ đồ phương pháp giấu tin trong miền video nén dựa trên DEW 16](#_Toc73347399)

[Hình 3: Ví dụ về việc chia khối lc 17](#_Toc73347400)

[Hình 4: Quá trình tính toán năng lượng trong vùng lc 18](#_Toc73347401)

[Hình 5: Cách tính toán và thay đổi năng lượng trên luông nén của một khối 8x8 DCT 19](#_Toc73347402)

[Hình 6: Năng lượng của khối DCT trong một số ví dụ cách chọn S(c) 20](#_Toc73347403)

[Hình 7: Quy trình nhúng bit 0 vào vùng Lc trong ví dụ 21](#_Toc73347404)

[Hình 8: Cách chiết xuất bit được nhúng trong vùng Lc trong ví dụ 22](#_Toc73347405)

[Hình 9: Ví dụ về sự phụ thuộc của khác biệt năng lượng dựa vào nội dung của video 23](#_Toc73347406)

[Hình 10: Khung hình sau khi được shuffle ngẫu nhiên bởi thuật toán trộn và khóa trộn cho trước 24](#_Toc73347407)

[Hình 11: Thuật toán nhúng bit 25](#_Toc73347408)

[Hình 12: Thuật toán chiết xuất bit 26](#_Toc73347409)

[Hình 13: Quy trình giấu tin trong nội dung video MPEG-2 27](#_Toc73347410)

[Hình 14: Quy trình mã hóa entropy thành phần hệ số DC 28](#_Toc73347411)

[Hình 15: Quy trình mã hóa entropy thành phần hệ số AC 29](#_Toc73347412)

# DANH MỤC BẢNG BIỂU

[Bảng 1: Phân loại bảng Huffman dành cho thành phần DC 22](#_heading=h.3as4poj)

[Bảng 2: Bảng Huffman các hệ số AC 23](#_heading=h.49x2ik5)

# Tổng quan giấu tin trong video

## Giới thiệu về kỹ thuật giấu tin

### Sơ lược về kỹ thuật giấu tin

Với hai mục đích là bảo mật cho những dữ liệu được giấu trong đối tượng mạng và bảo đảm an toàn cho chính các đối tượng chứa dữ liệu giấu trong đó (bảo vệ bản quyền) nên kỹ thuật giấu thông tin được phát triển thành hai lĩnh vực với những yêu cầu và tính chất khác nhau đó là giấu thông tin bí mật và thủy vân số.

Giấu thông tin bí mật (Steganography) có lịch sử hình thành và phát triển từ rất lâu đời, nó bắt nguồn từ Hi Lạp và được sử dụng cho tới ngày nay, chủ yếu phục vụ cho mục đích liên lạc bí mật.

Theo các tài liệu nghiên cứu ghi lại, kỹ thuật giấu tin cổ xưa nhất và cũng là đơn giản nhất được nhắc tới trong các tài liệu là khi vua Histiaeus (khoảng năm 440 TCN) cạo sạch tóc xăm thông điệp lên da đầu và chờ khi tóc mọc lại, người nô lệ đó chuyển thông tin tới người nhận. Sau đó, người ta sử dụng các vật liệu tự nhiên như bảng gỗ, sáp ong, hổ phách cho việc giấu thông tin.

Khi kỹ thuật phát triển hơn, con người sử dụng chữ viết với cỡ chữ nhỏ giấu trong các vật dụng hàng ngày (như các hộp, vali có hai đáy) để chuyển đi, hoặc dùng bồ câu để chuyển thông tin để che mắt các nhân viên an ninh, hải quan. Sang thế kỷ 17, người ta dùng cách đánh dấu vào các kí tự cần thiết trên một văn bản, một bài báo công khai nào đó rồi truyền tới tay người nhận. Sau đó là thời kì phát triển rực rỡ của công nghệ hoá học với sản phẩm là mực không màu - là các chất lỏng sản phẩm hữu cơ không màu và hiển thị màu khi gặp điều kiện hoá - lý thích hợp. Tới ngày nay với phương pháp kiểm tra độ ẩm bề mặt, mực không màu không còn tác dụng bảo mật nữa, nhưng nó vẫn còn được dùng như một dạng thuỷ vân để in các block nhỏ hay các chi tiết phát quang khi bị chiếu tia cực tím. Trong nửa cuối thế kỉ 19, các vi phim là bước phát triển kế tiếp, với sản phẩm hoàn hảo của các thợ ảnh chuyên nghiệp thì kích thước của mỗi thông điệp “chỉ nhỏ như một dấu chấm”.

Giấu thông tin trong ngôn ngữ, hành văn cũng được sử dụng từ lâu, với việc lựa chọn theo một quy luật xác định như ghép các từ đầu của mỗi câu hay nhặt các từ, các chữ cái theo một quy luật nào đó. Một hình thức cũng được người Trung Hoa sử dụng rất nhiều là sử dụng một tờ giấy mẫu có đục các lỗ làm một “mặt nạ” viết trước các nội dung cần giấu tại các vị trí đó, tiếp theo các nội dung “vô hại” khác lên sau, cách làm này tận dụng đặc điểm về chữ tượng hình và các từ đồng âm khác nghĩa của chữ Hán.

Kỹ thuật thuỷ vân bắt đầu được sử dụng muộn hơn, vào cuối thế kỷ 13 tại Ý, khi các nhà sản xuất giấy làm các hình mờ chìm trong giấy in để bảo vệ bản quyền nhà sản xuất. Watermark xuất phát từ đó, sau đó đã nhanh chóng phát triển ra toàn Châu Âu, áp dụng trong các lĩnh vực xuất bản khác cho tới ngày nay.Có nhiều cách để tiến hành phân loại các phương pháp giấu thông tin thông qua các tiêu chí khác nhau, như theo phương tiện chứa tin, các phương pháp tác động lên các phương tiện chứa tin, hay phân loại dựa theo các mục đích sử dụng....

1. Giấu thông tin bí mật

Đây là ứng dụng phổ biến nhất từ trước tới nay. Đối với giấu thông tin bí mật người ta quan tâm chủ yếu tới các mục tiêu sau:

* Độ an toàn của tin giấu (khả năng không bị phát hiện của tin giấu).
* Lượng thông tin tối đa có thể giấu trong một phương tiện chứa cụ thể mà vẫn có thể đảm bảo an toàn.
* Độ bảo mật của thông tin trong trường hợp giấu tin bị phát hiện.

Giấu thông tin bí mật không quan tâm nhiều tới các yêu cầu về khả năng bền vững của phương tiện chứa. Việc giải mã để nhận được thông tin cũng không cần phương tiện chứa gốc ban đầu. Các yêu cầu về khả năng chống tấn công không được quan tâm lắm, thay vào đó là thông tin giấu phải được bảo mật. Đối với các thuật toán giấu thông tin mật, người ta không chú trọng đến việc bảo vệ thông tin mật trước sự tấn công của các đối thủ mà thay vào đó quan tâm đến tính ẩn và tính an toàn đối với dữ liệu cần giấu.

1. Giấu thông tin thủy vân

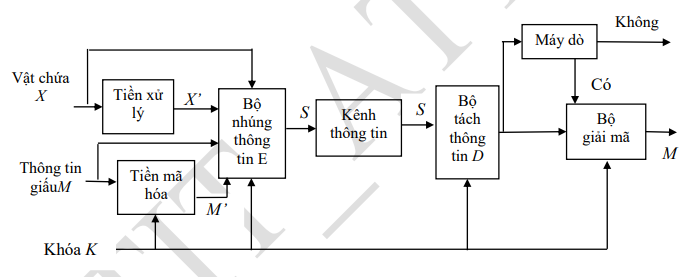
Khác với kỹ thuật giấu thông tin để giữ bí mật thông tin, giấu thông tin thủy vân có mục tiêu là bảo vệ bản quyền và xác thực thông tin. Vì vậy, kỹ thuật này không chống lại việc khai thác thông tin, mà quan trọng nhất đối với nó là đảm bảo tuyệt đối tính bền vững, nghĩa là không thể hủy bỏ được thông tin giấu nó trừ khi hủy chính sản phẩm chứa. Ngoài ra các thông tin nhúng cần có ảnh hưởng tối thiểu đối với phương tiện chứa, vì vậy thông tin cần giấu càng nhỏ càng tốt.

### Mô hình giấu tin

1. Khái niệm về kỹ thuật giấu tin

Kỹ thuật giấu thông tin là lĩnh vực chuyên nghiên cứu về các phương pháp, kỹ thuật, thuật toán nhằm nhúng thông tin vào một đối tượng dữ liệu khác. Cũng giống như mật mã học, kỹ thuật giấu tin bao gồm nhiều phương pháp, thuật toán và kỹ thuật khác nhau. Mỗi phương pháp, thuật toán và kỹ thuật có những yêu cầu về đầu vào và đầu ra khác nhau.

Thành phần cơ bản của kỹ thuật giấu thông tin gồm: Thuật toán giấu tin và bộ giải mã thông tin (tính đến cả khóa mật). Thuật toán giấu tin được dùng để giấu thông tin vào một phương tiện chứa bằng cách sử dụng một khoá bí mật được dùng chung bởi người mã và người giải mã, việc giải mã thông tin chỉ có thể thực hiện được khi có khoá. Bộ giải mã thực hiện quá trình giải mã trên phương tiện chứa đã chứa dữ liệu và trả lại thông điệp ẩn trong đó.



Hình 1: Mô hình chung cho quá trình giấu và tách tin

Các tham số chính trong mô hình trên là {X, M, S, K, E, D} trong đó:

* X là vật chứa.
* M là thông tin cần giấu.
* S là vật chứa đã chứa tin.
* K là khóa. - E là bộ nhúng thông tin.
* D là bộ tách thông tin.
* Quá trình nhúng thông tin là quá trình xử lý 𝐸: 𝑋 × 𝑀 × 𝐾 → 𝑆
* Quá trình tách thông tin là quá trình xử lý 𝐷: 𝑆 × 𝐾 → 𝑀, 𝑋

1. Các khái niệm trong mô hình giấu và tách tin

**Vật chứa X**: Là các đối tượng được dùng làm môi trường để giấu tin như ảnh, âm thanh, video, văn bản, …

**Thông tin cần giấu M**: chọn tùy theo mục đích của người sử dụng, nó có thể là thông tin (với các tin bí mật) hay các logo, hình ảnh bản quyền (thủy vân).

**Khóa K**: Trong quá trình giấu và tách tin có thể sử dụng nhiều hơn một khóa. Khóa là một chuỗi ngẫu nhiên được sinh ra bởi bộ sinh số mật mã an toàn (bộ sinh số này đáp ứng một số yêu cầu nhất định). Các số được sinh ra bởi bộ sinh số này có thể xác định vị trí các mẫu đã sửa đổi. Thông tin M sẽ được giấu một cách phù hợp với khóa trong các mẫu này do đó vật chứa sẽ ít bị biến dạng.

**Vật chứa tin S**: là vật chứa đã chứa tin, về cơ bản là vật chứa X và thông tin cần giấu M. Về mặt chất lượng S không được khác biệt nhiều so với vật chứa X.

**Bộ tiền mã hóa**: là một thiết bị được thiết kế để chuyển đổi thông tin cần giấu M sang một hình thức thuận tiện để giấu vào vật chứa. Trước khi giấu thông tin M vào vật chứa X, cần phải chuyển đổi M sang một dạng phù hợp. Ví dụ: Với X là 1 file ảnh thì M thường phải được biểu diễn dưới dạng mảng bit 2 chiều. Để tăng tính ổn định (tính chống biến dạng) của M thì M phải được mã hóa chống nhiễu hoặc sử dụng tín hiệu băng thông rộng. Sử dụng khóa K để tăng tính bí mật cho M. Đầu ra của bộ tiền mã hóa là thông tin đã mã hóa M’.

**Bộ tiền xử lý**: xác định các đặc thù của hệ thống nhận thức của con người từ đó xác định các vị trí ít quan trọng hoặc khó bị phát hiện trong vật chứa X giúp cho việc nhúng M vào X hiệu quả và ít bị phát hiện hơn.

**Bộ nhúng thông tin**: Thông tin sẽ được giấu vào trong vật chứa nhờ một bộ nhúng. Bộ nhúng là những chương trình thực hiện các thuật toán để giấu tin.

**Bộ tách thông tin**: Quá trình tách tin được thực hiện thông qua một bộ tách tin tương ứng với bộ nhúng thông tin của quá trình nhúng. Bộ tách triển khai các thuật toán tách tin tương ứng với các thuật toán giấu tin. Trong kỹ thuật giấu tin thì bộ tách thông tin cũng quan trọng không kém so với bộ nhúng thông tin. Bộ tách thông tin cũng sử dụng các phương pháp, thuật toán, kỹ thuật nhằm tìm kiếm và trích xuất thông tin. Thông thường thì mỗi kỹ thuật giấu tin thì sẽ có kỹ thuật tách tin tương ứng.

**Máy dò**: Dùng để phát hiện có thông tin được giấu trong vật chứa hay không hoặc thông tin giấu còn nguyên vẹn hay không (thông tin giấu có thể bị sửa đổi do các lỗi trong kênh thông tin, các lỗi trong hoạt động xử lý tín hiệu hoặc do các vụ tấn công cố ý). Máy dò sử dụng các biện pháp như khoảng cách Hamming hoặc tương quan chéo giữa vật chứa hiện tại và bản gốc (trong trường hợp có sự hiện diện của nó).

**Bộ giải mã**: dùng để phục hồi thông tin được giấu M. Nút này có thể bị bỏ qua

### Giấu tin trong dữ liệu đa phương tiện

Dữ liệu đa phương tiện là dạng dữ liệu đa dạng, phong phú hiện đang tồn tại trong mọi lĩnh vực ứng dụng của công nghệ thông tin. Chính vì vậy, dữ liệu đa phương tiện cũng được sử dụng làm phương tiện chứa trong quá trình giấu tin. Để lựa chọn dữ liệu làm phương tiện chứa cần phải tìm hiểu rõ về cấu trúc của loại dữ liệu đó. Một số dữ liệu đa phương tiện thường được sử dụng làm môi trường để giấu tin như: text, audio, video, ảnh…

1. Giấu tin trong ảnh số

Giấu thông tin trong ảnh số là một phần của khái niệm giấu thông tin với việc sử dụng ảnh số làm phương tiện chứa. Hiện nay, giấu thông tin trong ảnh là một phương pháp chiếm tỉ lệ lớn nhất trong các chương trình ứng dụng. Các phần mềm hệ thống giấu tin trong đa phương tiện bởi lượng thông tin được trao đổi trong hình ảnh là rất lớn và hơn nữa giấu thông tin trong ảnh cũng đóng vai trò hết sức quan trọng trong hầu hết các ứng dụng bảo vệ an toàn thông tin như: xác thực thông tin, xác định làm thay đổi thông tin, bảo vệ bản quyền tác giả, điều khiển truy cập, giấu thông tin mật.... Chính vì thế mà vấn đề này đã nhận được sự quan tâm rất lớn của các cá nhân, tổ chưc, trường đại học và viện nghiên cứu trên thế giới.

Ngày nay, khi ảnh số đã được sử dụng khá phổ biến, thì giấu thông tin trong ảnh đã đem lại nhiều ứng dụng quan trọng trên các lĩnh vực của đời sống xã hội. Ví dụ như đối với các nước phát triển, chữ ký tay đã được số hoá và lưu trữ sử dụng như là hồ sơ cá nhân của các dịch vụ ngân hàng và tài chính. Nó được dùng để xác thực trong các thẻ tín dụng của người tiêu dùng.

Trong một số ứng dụng về nhận diện thẻ chứng minh, thẻ căn cước, hộ chiếu..., người ta có thể giấu thông tin trên các ảnh thẻ để xác định thông tin thực. Một đặc điểm của giấu thông tin trong ảnh đó là thông tin được giấu một cách “vô hình”. Nó như là cách thức truyền thông tin mật cho nhau mà người khác không thể biết được, bởi sau khi giấu thông tin thì chất lượng ảnh gần như không thay đổi, đặc biệt đối với ảnh màu hay ảnh xám.

1. Giấu tin trong audio

Là các phương pháp nhằm giấu thông tin vào môi trường âm thanh. Giấu tin trong âm thanh thường áp dụng các biện pháp và kỹ thuật nhằm giấu thông tin vào trong các vùng âm thanh sao cho ngưỡng nghe của tai người không phát hiện ra những bất thường hoặc nhiễu do các thuật toán giấu tin gây ra. Hiện nay, giấu tin trong âm thanh cũng đang được quan tâm và ứng dụng trong thực tế.

1. Giấu tin trong video

Là các phương pháp nhằm giấu thông tin vào môi trường âm thanh hoặc hình ảnh. Giấu tin trong video cũng được quan tâm và được phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, nhận thực thông tin và bảo vệ bản quyền tác giả. Ví dụ như các hệ thống chương trình trả tiền xem theo video clip (pay per view application), ...

### Một số ứng dụng cụ của kỹ thuật giấu tin

1. Lấy dấu vân tay

Lấy dấu vây tay là quá trình thêm dấu vân tay vào một đối tượng hoặc xác định dấu vân tay có sẵn của một đối tượng. Dấu vân tay là đặc điểm phân biệt một đối tượng với các đối tượng khác. Các kỹ thuật gán dấu vân tay không có tác dụng phòng chống giả mạo và do đó không ngăn người dùng sao chép dữ liệu trái phép. Kỹ thuật này chỉ cho phép chủ sở hữu tìm ra được người dùng đã phân phối chúng bất hợp pháp. Ví dụ: Trong truyền hình vệ tinh được mã hóa, người dùng có thể được cấp một bộ khóa để giải mã các luồng video. Đài truyền hình có thể chèn dấu vân tay vào từng gói dữ liệu để phát hiện các sử dụng trái phép. Nếu một người dùng cung cấp khóa giải mã của họ cho những người khác và những người này giải mã và xem video trái phép, thì đài truyền hình có thể truy tìm thủ phạm phát tán video trái phép

* *Ví dụ: Lấy dấu vân tay bất đối xứng trong mua bán hình ảnh số*

Các dấu vân tay ngăn cản mọi người sao chép dữ liệu số bất hợp pháp bằng cách: khi có bản sao được phân phối lại bất hợp pháp, thì người bán có thể xác định người mua ban đầu của bản sao này. Thông thường, các chương trình lấy dấu vân tay là đối xứng, nghĩa là: Cả người mua và người bán đều biết bản sao này có dấu vân tay. Do đó, khi có bản sao được phân phối lại bất hợp pháp, thì có 2 nghi phạm trong trường hợp này: người mua ban đầu hoặc chính người bán. Điều đó dẫn đến việc người bán cố tình vu khống cho người mua hoặc người mua có thể dễ dàng chối bỏ hành vi sao chép của mình. Để giải quyết vấn đề, các chuyên gia sử dụng chương trình lấy dấu vân tay bất đối xứng, trong đó chỉ có người mua biết bản sao có dấu vân tay. Nếu sau đó người bán tìm thấy nó ở đâu đó, người bán có thể xác định người mua và chứng minh sự kiện này cho các bên thứ ba. Chương trình này bao gồm bốn giao thức: sinh khóa, nhúng dấu vân tay, xác định và tranh chấp.

1. Xác thực nội dung

Cách tiếp cận phổ biến cho vấn đề này là tạo một chữ ký số, chữ ký này sẽ được gắn liền với nội dung cần xác minh. Tuy nhiên chữ ký này dễ dàng bị mất đi trong quá trình truyền dữ liệu. Ví dụ: Trường hợp chữ ký được gắn vào một hình ảnh dạng JPEG. Nếu hình ảnh này được chuyển đổi sang định dạng tệp khác, ảnh mới không có khoảng trống cho chữ ký trong tiêu đề, chữ ký sẽ bị mất và hình ảnh không còn được xác minh. Một giải pháp cho vấn đề này là nhúng trực tiếp chữ ký vào nội dung bằng kỹ thuật thủy vân số. Lúc này, chữ ký số sẽ được coi là một dấu hiệu xác thực. Dấu hiệu xác thực này được thiết kế sao cho khi nội dung bị sửa đổi (dù là sửa đổi nhỏ nhất), thì dấu hiệu này sẽ trở nên không hợp lệ (thủy vân dễ vỡ). Để thực hiện được nhiệm vụ tạo chữ ký và nhúng chữ ký này vào vật chứa có thể tiến hành tách nội dung làm hai phần: một phần để tính chữ ký, một phần để nhúng chữ ký. Ví dụ: tính toán một chữ ký từ các bit cao của hình ảnh và nhúng chữ ký vào trong các bit thấp của hình ảnh này.

Một ý tưởng mở rộng hơn là việc xác thực cục bộ: nếu một hình ảnh được chia thành các khối và mỗi khối có dấu hiệu xác thực riêng được nhúng trong nó. Người quản sẽ nhận biết những phần nào của hình ảnh đã được xác thực và những phần nào đã được sửa đổi. Ví dụ về một cuộc điều tra của cảnh sát về một tội phạm. Hãy tưởng tượng cảnh sát nhận được một video giám sát đã bị giả mạo. Nếu video được xác thực bằng chữ ký truyền thống, cảnh sát sẽ biết là video đó không chính xác và không thể tin được. Tuy nhiên, nếu video này đã được xác thực cục bộ, họ có thể phát hiện ra rằng mỗi khung video đều đáng tin cậy ngoại trừ biển số xe hơi. Đây sẽ là bằng chứng mạnh mẽ cho thấy danh tính của một người có liên quan đến tội phạm đã bị xoá khỏi video.

1. Kiểm soát sao chép

Cách tiếp cận cơ bản nhất là nhúng thủy vân không bao giờ sao chép (never-copy watermark) vào dữ liệu và gắn sẵn các thiết bị phát hiện thuỷ vân vào trong các hệ thống đọc ghi. Mỗi khi có dữ liệu đi qua hệ thống đọc ghi, hệ thống này sẽ kiểm tra:

* Nếu dữ liệu không có thủy vân thì thiết bị đọc ghi cho phép sao chép dữ liệu.
* Nếu dữ liệu có thủy vân thì thiết bị đọc ghi cấm sao chép dữ liệu.

Tuy nhiên cách tiếp cận này có hạn chế là không phải tất cả hệ thống đọc ghi đều có gắn thiết bị phát hiện thủy vân do nhà sản xuất phải mất thêm chi phí lắp đặt và khách hàng thì thích thiết bị có khả năng tạo bản sao trái phép.

1. Bảo vệ quyền tác giả

Đây là ứng dụng cơ bản nhất của kỹ thuật thuỷ vân số. Một thông tin nào đó mang ý nghĩa quyền sở hữu tác giả sẽ được nhúng vào trong các sản phẩm. Thủy vân đó chỉ một mình người chủ sở hữu hợp pháp các sản phẩm đó có và được dùng làm minh chứng cho bản quyền sản phẩm. Giả sử có một sản phẩm dữ liệu số như ảnh, âm thanh, video được lưu thông trên mạng. Để bảo vệ các sản phẩm chống lại hành vi lấy cắp hoặc làm giả cần phải có một kỹ thuật để “dán tem bản quyền” vào sản phẩm này. Việc dán tem hay chính là việc nhúng thuỷ vân cần phải đảm bảo không để lại một ảnh hưởng lớn nào đến việc cảm nhận sản phẩm. Yêu cầu kỹ thuật đối với ứng dụng này là thuỷ vân phải tồn tại bền vững cùng với sản phẩm muốn bỏ thuỷ vân này mà không được phép của người chủ sở hữu thì chỉ còn cách là phá huỷ sản phẩm đó.

1. Một số ứng dụng khác

**Truyền thông tin mật**: Liên lạc bí mật giữa hai bên tham gia truyền thông mà không bị bên thứ ba phát hiện. Xây dựng kênh truyền thông bí mật.

**Ứng dụng xấu**: Tội phạm mạng sử dụng các kỹ thuật giấu tin mật để tạo Malware (phần mềm độc hại). Một trong cách phổ biến nhất là sử dụng các tệp đa phương tiện làm môi trường để giấu các mã độc. Một trong các kỹ thuật phổ biến nhất là sử dụng ảnh kỹ thuật số để che giấu cài đặt phần mềm độc hại hoặc một tập tin cấu hình hay lưu trữ trực tiếp toàn bộ mã độc hại. Ví dụ:

* Vào năm 2015, phần mềm độc hại của Vawtrak/ Neverquest bắt đầu sử dụng giấu tin để ẩn các cài đặt trong favicons. Phần mềm độc hại này chiết xuất các bit quan trọng nhất từ mỗi pixel của hình ảnh để tái tạo lại một URL được nhúng trước đó để tải tập tin cấu hình của nó.
* Các phần mềm độc hại như Ramsomware giấu tệp tin thực thi mã hóa trong các hình ảnh. Khi người dùng tải hình ảnh về, tệp tin độc hại này được thực thi, kết quả là toàn bộ dữ liệu trong máy người dùng bị mã hóa.
* Stegobot là mạng máy tính mà các máy trong mạng bị nhiễm mã độc thông qua giấu tin mật. Kẻ tấn công có thể điều khiển các máy trong mạng này từ xa và thực hiện những hành động mà chủ sở hữu của các máy này không hề hay biết. Một mạng Stegobot có thể có tới hàng trăm nghìn, thậm chí là hàng triệu máy tính.

## Đặc điểm của giấu tin trong video

Các file video được được đặc trưng các khung hình và khung âm thanh. Mỗi khung hình là một ảnh kỹ thuật số trực giao bitmap bao gồm một raster các điểm ảnh (pixel). Pixels chỉ có một thuộc tính màu sắc của chúng. Màu sắc của một điểm ảnh được biểu diễn bởi một giá trị cố định các bit.

Kỹ thuật giấu tin trong video trước kia chỉ tập trung vào việc giấu tin vào các khung hình của file video do các kỹ thuật này thường mang lại hiệu quả cao. Tuy nhiên, trong một thời gian gần đây kỹ thuật giấu tin trong khung âm thanh của video cũng đang được nghiên cứu và áp dụng nhiều trong thực tế. Bên cạnh đó, do đặc tính của file video là tập hợp các khung hình và âm thanh nên trong quá trình thực hiện giấu tin trong video cần phải lưu ý đến một số tham số đặc trung của video. Các tham số gồm: Pixel mỗi khung hình, Bit trên mỗi khung hình; kích thước video…Để làm rõ hơn về các tham số này, hãy xét ví dụ sau đây: Giả sử video có thể có thời gian (T) 1 giờ (3600 giây), kích thước khung hình 640 x 480 (R x C) ở độ sâu màu 24 bit và tỷ lệ khung hình 25 fps (số khung hình được hiển thị trong 1s). Các đặc trung của một số tham số được tính như sau:

* Pixel mỗi khung hình = 640 \* 480 = 307.200
* Bit trên mỗi khung hình = 307.200 \* 24 = 7.372.800 = 7, 37 Mbits
* Tỷ lệ bit (BR) = 7.37 \* 25 = 184, 25 Mbits / sec
* Kích thước video (VS) = 184 Mbits / sec \* 3600 giây = 662.400 Mbits = 82.800 MB

## Một số định dạng file Video

1. MPEG

Moving Picture Experts Group (MPEG) – “Nhóm các chuyên gia hình ảnh động” là một nhóm các quy tắc hoạt động được thành lập bởi ISO và IEC để thiết lập các tiêu chuẩn cho việc truyền tải âm thanh và video. Công nghệ mới cho phép có nhiều cách để nén dữ liệu video mà vẫn đảm bảo được chất lượng hình ảnh đạt yêu cầu. Một số chuẩn nén MPEG phổ biến như: MJPEG; MPEG-2; MPEG-4; Chuẩn H.264. Trong đó chuẩn MPEG-4 là chuẩn cho các ứng dụng Multimedia. MPEG-4 là một tiêu chuẩn cho nén ảnh kỹ thuật truyền hình số, các ứng dụng về đồ họa và video tương tác hai chiều (Games, Video conference) và các ứng dụng Multimedia tương tác hai chiều (World Wide Web hoặc các ứng dụng nhằm phân phát dữ liệu Video như truyền hình cáp, Internet Video...). MPEG-4 đã trở thành một tiêu chuẩn công nghệ trong quá trình sản xuất, phân phối và truy cập vào các hệ thống Video. Chuẩn H.264 là một chuẩn mã hóa/giải mã video và định dạng tệp video đang được sử dụng rộng rãi nhất hiện này vì khả năng ghi, nén và chia sẻ video phân giải cao. Tệp này có dung lượng thấp nhưng mang lại chất lượng rất cao. H.264 cũng cho chất lượng hình ảnh tốt nhất, kích thước file nhỏ nhất, hỗ trợ DVD và truyền với tốc độ cao so với các chuẩn trước đó.

1. AVI

Định dạng AVI (Audio Video Interle) là một định dạng số đa phương tiện do Microsoft giới thiệu vào tháng khoảng 11/1992 như một chuẩn video dành cho Windows. Tệp AVI có thể chứa cả dữ liệu âm thanh và video trong một tệp, cho phép đồng bộ với phát lại audio – video. Đặc điểm của tệp AVI là dạng video không nén.

1. FLV

Định dạng FLV (Flash video) là một dạng file nén từ các file video khác để tải lên trang web với dung lượng nhỏ, tuy nhiên chất lượng của hình ảnh không bằng được file gốc (MP4, WAV…). Tệp FLV được lựa chọn cho việc nhúng video trong web, đây là định dạng hay được sử dụng bởi ứng dụng trên web như: Youtube, Google Video, Yahoo! Video…

1. H.263

H.263 được sử dụng rộng rãi trên internet như tệp FLV, hay sử dụng trong hội nghị, truyền hình, điện thoại video, giám sát và theo dõi.

1. WMV

Định dạng WMV (Windows Media Video) là một định dạng video chứa video được mã hóa theo bộ code Windows Media Video và âm thanh được mã hóa theo codec Windows Media Audio codec.

1. MP4

Định dạng MP4 là định dạng thường được sử dụng để lưu trữ video và âm thanh, nhưng cũng có thể được sử dụng để lưu trữ dữ liệu khác như phụ đề và hình ảnh. MP4 cho phép truyền tải trên Internet.

1. MOV

Định dạng MOV là một định dạng được Apple phát triển. Đây là một định dạng đa phương tiện phổ biến, thường được dùng trên Internet do ưu điểm tiết kiệm dung lượng của nó.

1. H265

Định dạng H.265 hay còn gọi là HEVC (High Efficiency Video Coding – code video hiệu suất cao) là một định dạng video mang lại khả năng nén cao gần gấp đôi so với H.264/AVC. Định dạng H.265 giúp giảm băng thông cần thiết để truyền tải phim, đặc biệt là trên các thiết bị di động.

## Phân loại kỹ thuật giấu tin trong video

Do sự phong phú và đa dạng về định dạng file video cũng như khả năng có thể giấu tin trong cấu trúc của một file video nên hiện nay đang có rất nhiều kỹ thuật giấu tin được nghiên cứu và áp dụng vào môi trường video. Theo thống kê từ các tài liệu thì có thể phân loại các kỹ thuật giấu tin trong video thành một số kỹ thuật sau:

### Phân loại giấu tin trong video theo kỹ thuật giấu

* Giấu thông tin trong miền hệ số.
* Giấu thông tin trong mặt phẳng bit.
* Giấu thông tin vào sự thay đổi khung cảnh.
* Giấu thông tin vào hệ số khác biệt năng lượng.
* Giấu thông tin trong video chuẩn H.264.
* Giấu thông tin trong video chuẩn H.265

### Phân loại giấu tin theo miền giấu

* Giấu thông tin trên miền hình ảnh của video.
* Giấu thông tin trên miền âm thanh của video.

### Phân loại giấu tin theo mục đích

Thủy vân số:

* Giấu thông tin vào sự thay đổi khung cảnh
* Giấu thông tin trong video chuẩn H.264, H265…
* Giấu thông tin trong miền hệ số.

Giấu tin mật: 90

* Giấu thông tin trong mặt phẳng bit.

Dựa vào thống kê về những nghiên cứu trong lĩnh vực giấu tin trong video cũng như những phương pháp giấu tin trong video đang được áp dụng và triển khai trong thực tế hiện nay thì rõ ràng ứng dụng và tiềm năng của các kỹ thuật giấu tin trong video là rất lớn. Tiếp theo, bài giảng sẽ trình bày chi tiết về một số kỹ thuật giấu tin đã được nghiên cứu và áp dụng hiện nay.

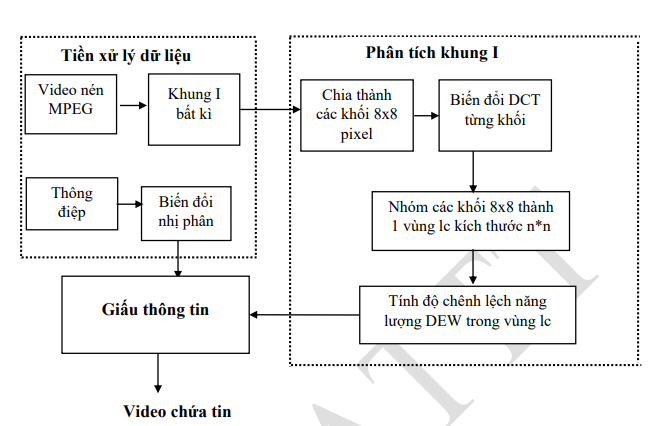
# Phương pháp giấu tin trong video

## Phương pháp giấu tin trong miền video nén dựa trên sự khác biệt năng lượng

### Tổng quan

Các kỹ thuật nhúng thủy vân dựa trên mối tương quan có lợi thế là có thể lấy thủy vân ra được từ giải mã các luồng video hoặc mã hóa lại chúng. Tuy nhiên để nhúng hoặc phát hiện một thủy vân 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 vân 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.

### Quy trình giấu tin



Hình 2: Sơ đồ phương pháp giấu tin trong miền video nén dựa trên DEW

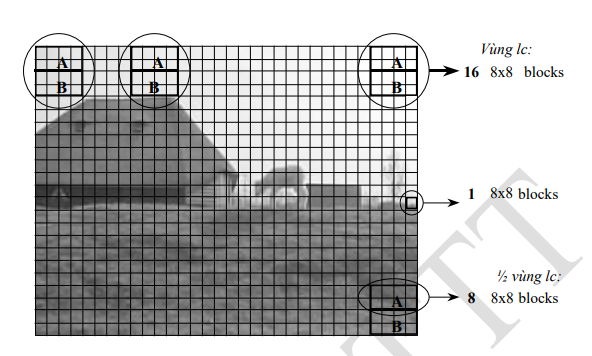
Từ sơ đồ giấu tin trong video theo phương pháp DEW bao gồm các bước sau:

**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 giấu thông tin mật. Đối với phương pháp DEW nên chọn khung I.

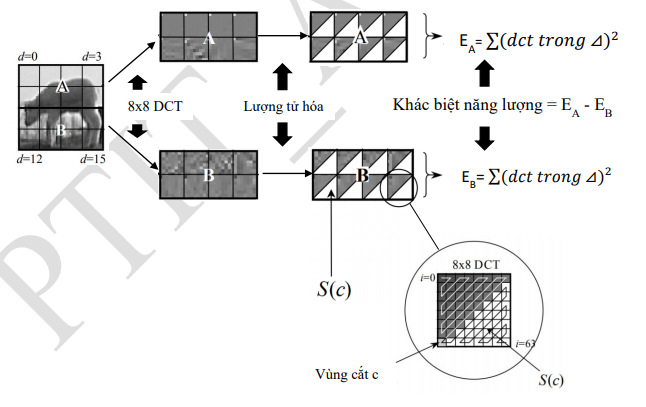
**Bước 2**. Phân tích khung hình.

* Ảnh được chia thành các khối 8x8 pixel. Rồi từ đó đưa về hệ số DCT (các khối 8x8 hệ số DCT).
* Nhóm các khối 8x8 thành một vùng lc kích thước n\*n: Lưu ý: Trong trường hợp trên với n = 16 khối 8x8 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 khối 8x8 DCT. Hình dưới đây mà mô tả về việc chia khối lc



Hình 3: Ví dụ về việc chia khối lc

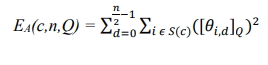
* Tính độ chênh lệch năng lượng DEW trong vùng lc:



Hình 4: Quá trình tính toán năng lượng trong vùng lc

*Giải thích các thành phần trong sơ đồ*

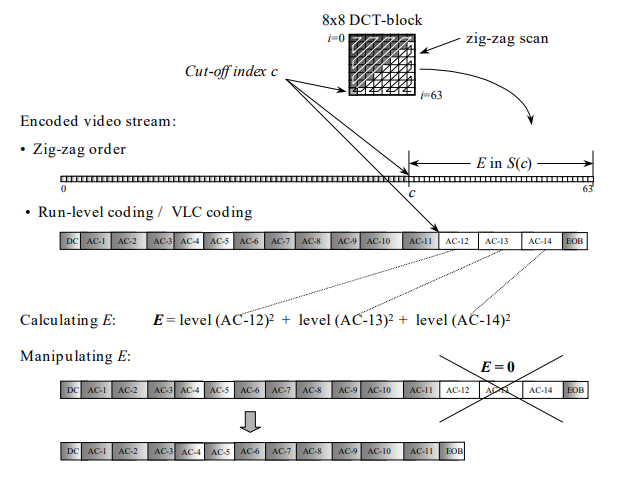
* EA năng lượng nửa trên: Năng lượng trong một vùng EA bằng tổng bình phương của một tập con cụ thể của các hệ số DCT trong vùng EA này.
* EB năng lượng nửa dưới: tính tương tự như EA + D là sự khác biệt năng lượng. Sự khác biệt được định nghĩa theo công thức: D = EA – EB;
* Tập con này biểu diễn bởi S(c) (hình tam giác trắng). Công thức tính năng lượng tại một vùng như sau:



*Trong đó:*

* EA 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.
* 𝜃𝑖,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)

Có một số lý do để tính toán sự chênh lệch năng lượng này theo khối tam giác trong 1 Block DCT. Lý do quan trọng nhất là tính toán sự chênh lệch năng lượng và thay đổi EA và EB có thể dễ dàng được thực hiện trên luồng nén. Tất cả các hệ số DCT cần thiết cho việc tính toán EA hoặc EB được đặt ở vị trí thuận tiện ở cuối nén 8x8 khối DCT sau khi đặt hàng zig-zag. Các hệ số có thể được buộc bằng 0 để điều chỉnh năng lượng mà không cần mã hóa lại luồng bằng cách dịch chuyển điểm cuối của điểm đánh dấu khối (EOB) đối với hệ số DC. Hình 5 minh họa bằng đồ thị quy trình tính toán E trong một khối DCT nén duy nhất và thay đổi E bằng cách loại bỏ hệ số DCT nằm ở cuối quá trình quét zig-zag (tức là hệ số DCT tần số cao).

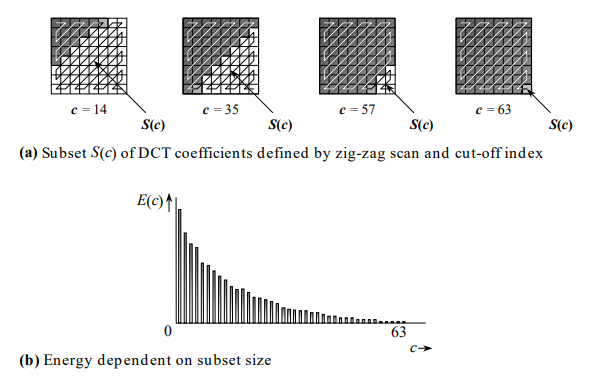


Hình 5: Cách tính toán và thay đổi năng lượng trên luông nén của một khối 8x8 DCT

Năng lượng hiện diện trong các tiểu vùng lc A và B được xác định bởi Công thức đóng vai trò trung tâm trong quá trình nhúng và chiết xuất hình mờ. Giá trị của EA và EB được xác định bởi 4 yếu tố:

* Nội dung không gian của các vùng LC(A) và B
* Số khối n trên mỗi vùng lc
* Hệ số chất lượng JPEG trước hoặc lượng tử hóa lại Qjpeg
* Kích thước của tập con S(c) (tức là diện tích hình tam giác)

Kích thước của tập con S(c) (Công thức 4.2.2) được xác định bằng cách quét zig-zag tiêu chuẩn và chỉ số cắt c. Nếu hệ số DCT được quét theo đường zig-zag được đánh số từ 0 đến 63, trong đó hệ số có chỉ số 0 đại diện cho thành phần DC và hệ số có chỉ số 63 thành phần tần số cao nhất, tập hợp con này bao gồm các hệ số DCT với chỉ số c… 63 (c> 0). Trong Hình 4.3.1, một số ví dụ được hiển thị về các tập hợp con được xác định bởi tăng các chỉ số giới hạn. Các năng lượng xác định bằng thực nghiệm tương ứng được vẽ dưới đây. Hình này cho thấy rằng việc tăng chỉ số giới hạn sẽ làm giảm năng lượng.



Hình 6: Năng lượng của khối DCT trong một số ví dụ cách chọn S(c)

**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 = EA – EB = EA – 0 = +EA

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

D = EA – EB = 0 – EB = – EB

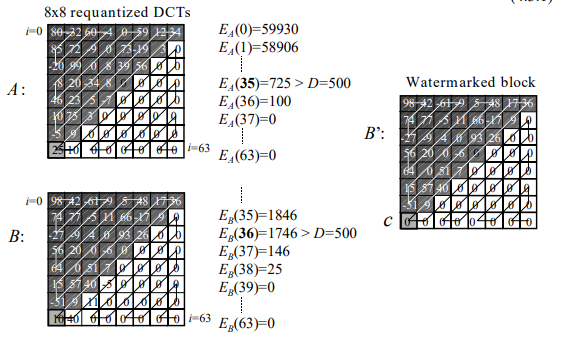
*Ví dụ dưới đây sẽ thể hiện rõ hơn quy trình giấu thông tin:*

* Cần giấu bit b0 = 0. Xét một vùng lc với n = 2 (tương ứng với 2 khối block DCT).
* Năng lượng khác biệt D = 500.

Chỉ số c sẽ được xác định bằng công thức:



Từ đó ta xác định được vị trí c có mức năng lượng vượt quá 500 trong khối A là A(35), trong khối B là B(36). Và bit nhúng là b0 = 0. Suy ra ta sẽ phải loại bỏ năng lượng của vùng S(c) trong khối B. Tức sẽ đưa các hệ số DCT của các block trong vùng S(c) của khối B về 0. Từ đó ta thu được khối B’



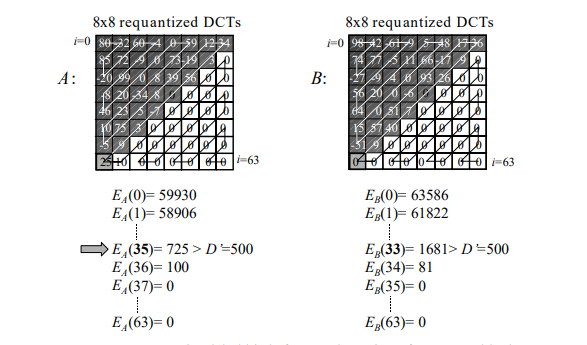
Hình 7: Quy trình nhúng bit 0 vào vùng Lc trong ví dụ

Để triết xuất được bit đã giấu từ vùng Lc đã sử dụng cho việc nhúng bit bên trên, ta sẽ làm như sau:

* Xác định chỉ số c triết xuất bằng công thức

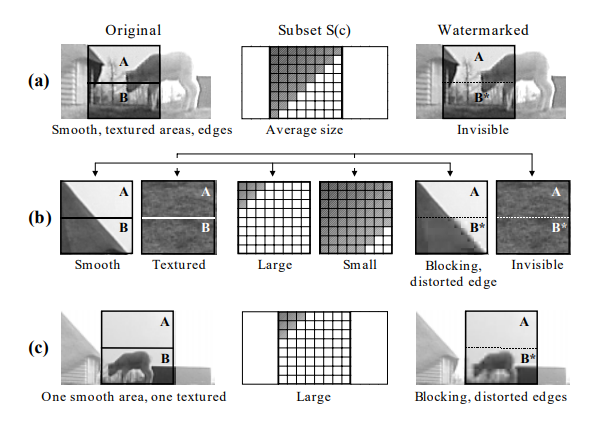


* Trong hình 8 đưa ra một ví dụ về việc trích xuất bit nhãn b0 từ vùng lc bao gồm n = 2 khối DCT đã được nhúng . Để khai thác, D’= D = 500 được sử dụng. Chỉ số giới hạn tối đa mà EA năng lượng vượt quá D’= 500 là 35, đối với EB chỉ số giới hạn này là 33. Điều này có nghĩa là thuật toán nhúng watermark đã sử dụng chỉ số giới hạn là 35. Sự chênh lệch năng lượng EA(35)-EB (35) = 725. Do chênh lệch năng lượng là dương, giá trị 0 được gán cho bit nhãn b0. Từ đó ta triết xuất được bit 0 đã được nhúng trong video



Hình 8: Cách chiết xuất bit được nhúng trong vùng Lc trong ví dụ

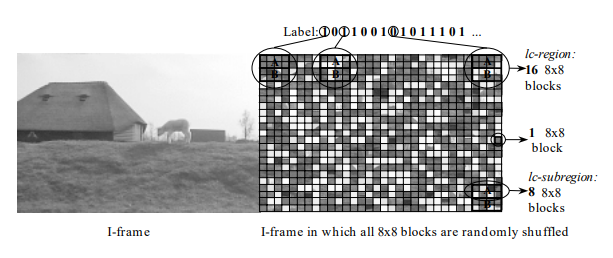
Thuật toán áp dụng trong hình thức này phụ thuộc nhiều vào nội dung video. Hình 9 hiển thị một số ví dụ về sự phụ thuộc nội dung này. Trong hình a, một vùng lc là được mô tả trong đó các tiểu vùng lc A và B đều chứa các vùng nhẵn, vùng có rang cưa hay vùng có nhiều kết cấu. Đây là những ví dụ điển hình về các vùng có năng lượng trung bình trong Hệ số DCT AC. Trong trường hợp này, quy trình nhúng hình mờ sẽ chọn một tập hợp con S(c) với chỉ số cắt ở đâu đó giữa phạm vi 1… 63. Điều này có nghĩa là một số hệ số ở dải tần số cao nhất và dải tần số trung bình bị loại bỏ. Nếu số lượng năng lượng bị loại bỏ trong các dải tần này bị hạn chế, bit nhãn sẽ không đáng chú ý. Vì lượng tử hóa lại bằng cách mã hóa lại ở tốc độ bit thấp hơn sẽ không ảnh hưởng đến chênh lệch năng lượng trong dải tần số trung bình một cách nghiêm trọng, bit nhãn sẽ tồn tại sau một cuộc tấn công re-encoding.



Hình 9: Ví dụ về sự phụ thuộc của khác biệt năng lượng dựa vào nội dung của video

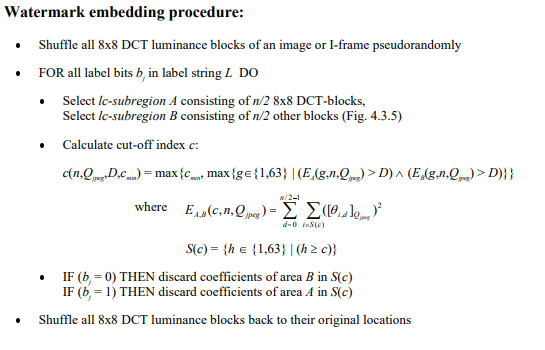
* Trong hình a, vùng Lc chứa nhiều những khu vực mịn, răng cưa và kết cấu, từ đó ta tính được chỉ số c ở mức trung bình, suy ra vùng S(c) cũng ở mức trung bình. Khi giấu bit vào vùng đó thì chất lượng video sẽ ít bị thay đổi và khó nhân ra.
* Trong hình b1 chỉ chứa các vùng mịn, từ đó sẽ xác định được 1 vùng S(c) rất lớn trong block DCT, khi nhúng bit vào thì sẽ làm thay đổi chất lượng video và sẽ dễ nhận thấy bằng mắt thường khi 2 vùng liền nhau sẽ có 1 vùng mịn và 1 vùng răng cưa.
* Trong hình b2, tuy sau khi mã hóa thì chất lượng video sẽ rất ít có thay đổi và khó nhận thấy bằng mắt thường, nhưng với việc vùng S(c) quá nhỏ yêu cầu bằng cách mã hóa lại ở tốc độ bit thấp hơn sẽ ảnh hưởng đến sự chênh lệch năng lượng trong nghiêm trọng là các dải tần số cao nhất, bit nhãn sẽ không tồn tại sau một cuộc tấn công re-encoding.
* Tình huống xấu nhất được mô tả trong Hình c, trong đó một tiểu vùng lc là hoàn toàn mịn, trong khi phần còn lại có kết cấu và chứa các cạnh sắc nét. Nếu tích cực chênh lệch năng lượng D = EA - EB phải được tạo ra trong vùng lc này, tất cả các hệ số AC DCT trong lc-tiểu vùng B phải được loại bỏ bằng cách chọn một tập hợp con cực lớn S (c) để tạo EA> EB. Sự hiện diện của bit nhãn hiển nhiên trở nên rõ ràng trong tiểu vùng lc B.

Để tránh việc giấu bit nhãn vào những vùng Lc không mong muốn, và có nhiều sự lựa chọn hơn khi tiến hành nhúng để tăng hiệu quả của phương pháp, trước khi nhúng chúng ta sẽ tiến hành trộn ngẫu nhiên các block trong khung hình đã chọn để tạo ra them nhiều vùng lc có chứa cả những vùng mịn, răng cưa và vùng kết cấu.

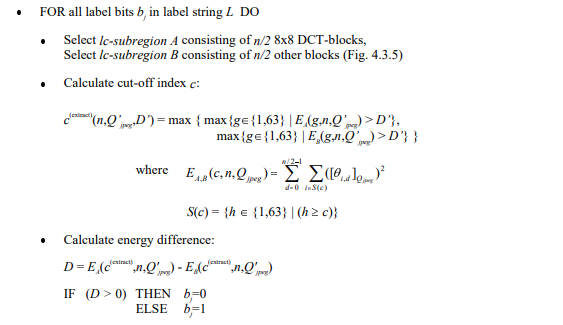
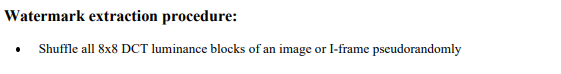


Hình 10: Khung hình sau khi được shuffle ngẫu nhiên bởi thuật toán trộn và khóa trộn cho trước

Với tất cả những tìm hiểu bên trên, ta có thể tổng hợp lại thuật toán nhúng và chiết xuất bit nhãn với phương pháp nhúng bit sử dụng sự chênh lệch năng lượng như sau:

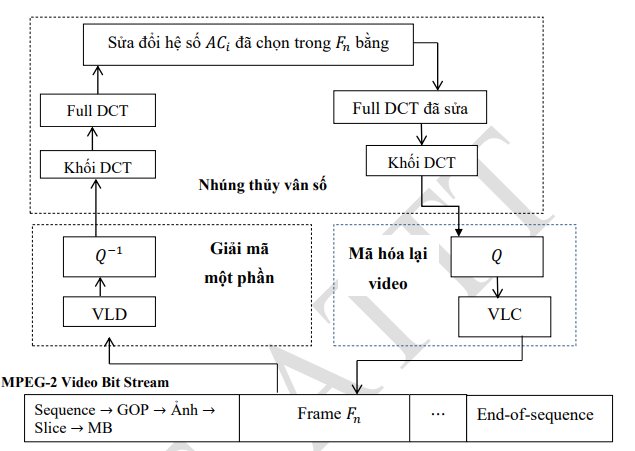
* Thuật toán nhúng bit: 

Hình 11: Thuật toán nhúng bit

* Thuật toán chiết xuất bit: 

Hình 12: Thuật toán chiết xuất bit

## Phương pháp giấu tin trên miền nén của video chất lượng cao



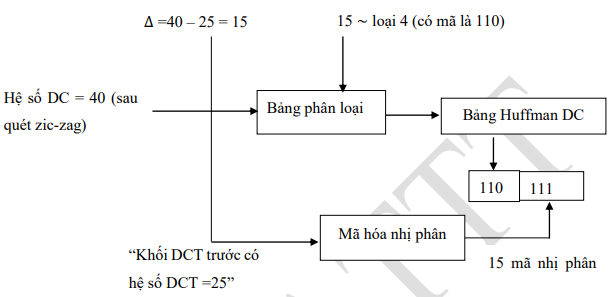
Hình 13: Quy trình giấu tin trong nội dung video MPEG-2

Từ quy trình giấu tin trong video MPEG-2, có thể thấy được các bước chính trong kỹ thuật giấu tin trên miền nén của video chất lượng cao như sau:

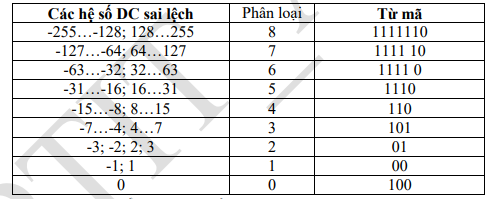
**Bước 1**. Lựa chọn khung: Chọn một khung bất kỳ để tiến hành nhúng, nên chọn khung I vì khung I là khung cơ sở và có thể coi là ảnh gốc, với khung này khi giải mã thì không cần lấy thông tin từ khung khác.

**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 hoá bằng các từ mã dài, quá trình này được gọi là phương pháp mã hoá 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:

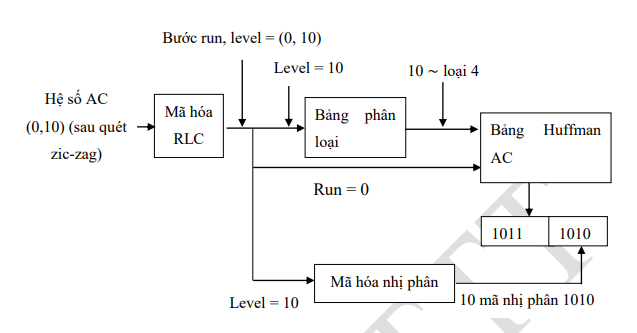


Hình 14: Quy trình mã hóa entropy thành phần hệ số DC

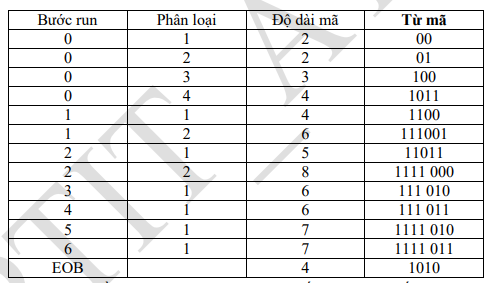


*Bảng 1: Phân loại bảng Huffman dành cho thành phần DC*

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



Hình 15: Quy trình mã hóa entropy thành phần hệ số AC

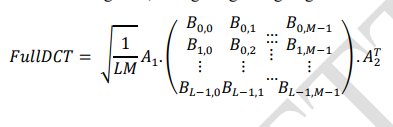


*Bảng 2: Bảng Huffman các hệ số AC*

* Lượng tử hóa: Đầu vào ở bước này là 64 hệ số DCT của khối 8x8 sẽ được lượng tử hoá dựa trên một bảng lượng tử gồm 64 phần tử Q(u, v) với 0≤ u, v≤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). Trong chương 2 bài giảng đã trình bày chi tiết về bước lượng tử hóa trong kỹ thuật biến đổi DCT. 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.

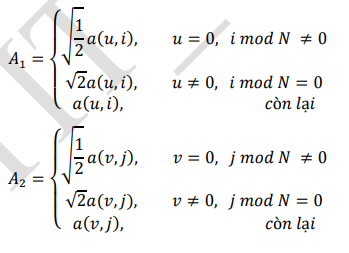
**Bước 3**. Giấu thông tin. Để có thể giấu được thông tin vào khung hình vừa lựa chọn. Tại đây người giấu tin cần thực hiện các biện pháp tiền xử lý như sau:

* 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,𝑗 là N. L và M số hàng và cột trong hàng tương ứng

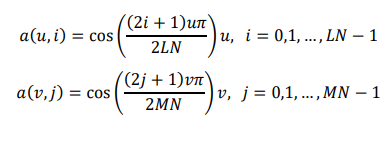


*Trong đó:*

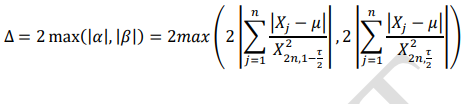
* 𝐿𝑁 × 𝑀𝑁: kích thước khung hình;
* L và M số hàng và cột trong hàng tương ứng;
* N kích thước của một khối 𝐵𝑖, 𝑗. 𝐵𝑖, 𝑗 là ma trận với N × N yếu tố và đại diện cho tập hợp các hệ số DCT cho khoanh vùng.
* 𝐴1và 𝐴2 là các ma trận vuông với LN × LN và MN × MN kích thước tương ứng và được định nghĩa theo công thức:



*Trong đó:*



* 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:



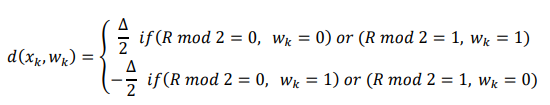
*Trong đó:*

* 𝛼, 𝛽: khoảng tin cậy
* 𝜇: tham số vị trí (là giá trị trung bình của biểu đồ)
* 𝜏: tỷ lệ bit lỗi BER
* X: chuỗi các biểu đồ khác biệt
* 𝑋2n2: biểu thị định lượng pth của phân bố X2 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 = {w1, w2, …, wn}, trong đó 𝑤𝑘 ∈ {0,1} và n có nghĩa là độ dài của thông tin cần giấu. 𝑥 = {x1, x2, …, xn} được chọn các hệ số DCT toàn khung của một khung và 𝑦 = {y1, y2, …, y3} 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 và giá trị được sửa đổi:

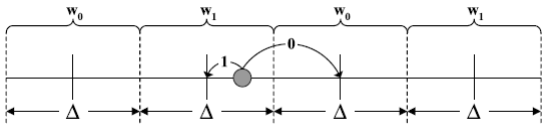


*Trong đó:*

* 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:



*Trong đó R viết tắt cho 𝑟𝑜𝑢𝑛𝑑 ()*



*Hình 8: Thay thế giá trị cho thông tin cần giấu trong QIM*

* Giai đoạn 5: Biến đổi DCT ngược. Sau khi đã giấu 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 giấu 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.

# Cài đặt và kiểm nghiệm

## Môi trường cài đặt

Chương trình giấu tin trong video được thực hiện trên môi trường Java phiên bản 8 của Oracle Corporation với nhiều tính năng liên quan đến đọc và lưu dữ liệu cũng như chuyển đổi giữa các kiểu dữ liệu, giúp cài đăt chương trình một cách thuận tiện.

Hệ điều hành tối thiểu bao gồm: Win 7, Window Vista hoặc WinNT.

Hệ điều hành khuyên dùng: Win 10.

Phần mềm yêu cầu: Java SE Development Kit 8; NetBean IDE hoặc Eclipse IDE.

## Sơ đồ thuật toán

### Thuật toán giấu tin

Có thể ghi tiếp

Kiểm tra số chunk khả dụng

Hết

Chuyển sang số chunk tiếp theo

Ghi hết 1 chunk

vẫn còn dữ liệu

Đọc tệp tin

Kiểm tra định dạng .avi

Báo lỗi

Không hợp lệ

Ghi dữ liệu vào chunk

Hợp lệ

Kiểm tra độ dài dữ liệu

Kết thúc

Đã ghi hết

Các bước thực hiện:

* Đọc thông điệp/tập tin (nối chuỗi với một chuỗi thông điệp kết thúc để sử dụng trong quá trình lấy tin) và tập tin video đầu vào (phải có định dạng .avi) và chuyển thành mảng các byte (byte[]).
* Từ mảng byte của tập tin video, duyệt các bộ 4 byte liên tiếp để tìm vị trí của thẻ “movi” (có dạng 01101101 01101111 01110110 01101001) của video (nếu không tìm được có nghĩa là tập tin video không đúng định dạng => báo lỗi).
* Nếu tìm được, sử dụng phương pháp LSB để thay đổi các bit có trọng số nhỏ nhất của các byte luồng dữ liệu (nằm ngay sau thẻ “movi”) bằng các bit chứa thông tin của thông điệp/tập tin cần giấu.
* Nếu đã ghi đè hết các bit của chunk mà chưa ghi hết được dữ liệu của thông điệp cần ẩn giấu, tìm đến chunk tiếp theo và tiếp tục sử sử dụng LSB cho đến khi giấu hết được toàn bộ thông điệp (nếu đã ghi đè lên tất cả các chunk mà vẫn chưa giấu hết được thông điệp có nghĩa là thông điệp quá dài hoặc tập tin video chứa có dung lượng quá ngắn không đủ yêu cầu => báo lỗi)
* Từ mảng byte[] đã được ghi đè, viết vào một tập tin video mới có định dạng .avi, thu được tập tin video có chứa thông điệp/tập tin cần ẩn giấu.

### Thuật toán lấy tin

Có thể đọc tiếp

Tìm chuỗi dấu hiệu kết thúc

Hết

Đọc dữ liệu từ chunk

Không tìm thấy

Đọc tệp tin

Kiểm tra định dạng .avi

Báo lỗi

Không hợp lệ

Đọc dữ liệu từ chunk

Hợp lệ

Tìm chuỗi dấu hiệu kết thúc

Kết thúc

Tìm thấy

Các bước thực hiện:

* Đọc tập tin video chứa thông điệp/tập tin ẩn (phải có định dạng .avi) và chuyển thảnh mảng các byte (byte[]).
* Từ mảng byte của tập tin video, duyệt các bộ 4 byte liên tiếp để tìm vị trí của thẻ “movi” (có dạng 01101101 01101111 01110110 01101001) của video (nếu không tìm được có nghĩa là tập tin video không đúng định dạng => báo lỗi).
* Nếu tìm được, đọc các bit có trọng số nhỏ nhất của các byte luồng dữ liệu (nằm ngay sau thẻ “movi”) và lưu vào mảng byte[].
* Nếu đã ghi đè hết các bit của chunk mà chưa thấy thông điệp kết thúc, tìm đến chunk tiếp theo và tiếp tục đọc các bit cho đến khi tìm thấy hết. Thông điệp kết thúc.
* Từ mảng byte[] đọc được, chuyển sang dạng chuỗi ký tự để lấy ra thông điệp/tập tin ẩn giấu.

## Cài đặt thuật toán

### Kiểm tra định dạng .avi

private final String VIDEO\_START\_SEQUENCE = "movi";

public int getVideoStreamPos(byte[] content) throws ArrayIndexOutOfBoundsException {

int bytePos = -1;

String valueFound = "";

// duyệt hết phần header của file

while (!valueFound.equals(this.VIDEO\_START\_SEQUENCE)) {

// lấy 4 bytes để kiểm tra định dạng file

bytePos++;

valueFound = "";

for (int i = bytePos; i < bytePos + 4; i++) {

valueFound += (char) content[i];

}

}

// trả về vị trí vị trí đầu tiên của phần data

return bytePos;

}

### Chuyển sang chunk tiếp theo

public int getNextChunkSize(byte[] content, int chunkPos) {

int chunkSize = 0;

int currPos = chunkPos + 4;

byte[] sizeBytes = new byte[4];

// lấy lần lượt 4 bytes vào 1 chunk

for (int i = sizeBytes.length - 1; i >= 0; i--) {

sizeBytes[i] = content[currPos];

currPos++;

}

ByteBuffer sizeBytesWrapper = ByteBuffer.wrap(sizeBytes);

chunkSize = sizeBytesWrapper.getInt();

// trả về số bytes thực tế lấy được của chunk

return chunkSize;

}

### Ghi dữ liệu vào chunk

public String saveMessageIntoChunk(String message, byte[] content, int pos, int chunkSize) {

String remainingMessage = null;

int messageIndex = 0;

int charIndex = 0;

int currPos = pos;

int overhead = chunkSize % 8;

for (int i = 0; i < (chunkSize - overhead); i++) {

if (messageIndex < message.length()) {

char messageChar = message.charAt(messageIndex);

int bit\_ch = (messageChar >> charIndex) & 1;

// chèn dữ liệu vào bit có trọng số thấp nhất của từng byte

content[currPos] = (byte) ((content[currPos] & (byte) 0xFE) | (byte) bit\_ch);

currPos++;

if (charIndex >= 7) {

charIndex = 0;

messageIndex++;

} else {

charIndex++;

}

} else {

break;

}

}

if (messageIndex < message.length()) {

remainingMessage = message.substring(messageIndex);

}

return remainingMessage;

}

### Đọc dữ liệu từ chunk

public String getMessageFromChunk(String message, byte[] content, int pos, int chunkSize) {

String resultMessage = message;

int currPos = pos;

int overhead = chunkSize % 8;

int charIndex = 0;

char messageChar = 0;

for (int i = 0; i < (chunkSize - overhead); i++) {

// lấy bit có trọng số thấp nhất trên từng byte dữ liệu

int bit\_byte = (content[currPos] & (byte) 0x01) << charIndex;

messageChar = (char) (messageChar | (char) bit\_byte);

currPos++;

if (charIndex >= 7) {

resultMessage += messageChar;

messageChar = 0;

charIndex = 0;

if (resultMessage.length() >= 4) {

if (resultMessage.substring(resultMessage.length() - END\_SEQUENCE.length()).equals(END\_SEQUENCE)) {

break;

}

}

} else {

charIndex++;

}

}

return resultMessage;

}

### Giấu tin

private final String END\_SEQUENCE = "\u0000";

public byte[] hideMessage(String message, byte[] fileContent) throws ArrayIndexOutOfBoundsException {

int currentPos = 0;

//kiểm tra định dạng file và lấy vị trí byte đầu tiên của phần data của file video

currentPos = getVideoStreamPos(fileContent);

currentPos += 4;

String messageToHide = message + END\_SEQUENCE;

int chunkSize;

while ((chunkSize = getNextChunkSize(fileContent, currentPos)) > 0 && messageToHide != null) {

// lấy từng chunk

currentPos += 8;

// giấu dữ liệu vào chunk

messageToHide = saveMessageIntoChunk(messageToHide, fileContent, currentPos, chunkSize);

currentPos += chunkSize;

}

return fileContent;

}

### Lấy tin

public String findMessage(byte[] fileContent) throws ArrayIndexOutOfBoundsException {

String message = "";

int currentPos = 0;

// kiểm tra định dạng file và lấy vị trí byte đầu tiên phần data file video

currentPos = getVideoStreamPos(fileContent);

currentPos += 4;

int chunkSize;

while ((chunkSize = getNextChunkSize(fileContent, currentPos)) > 0) {

// duyệt từng chunk

currentPos += 8;

// lấy dữ liệu từ chunk

message = getMessageFromChunk(message, fileContent, currentPos, chunkSize);

currentPos += chunkSize;

if (message.substring(message.length() - END\_SEQUENCE.length()).equals(END\_SEQUENCE)) {

break;

}

}

return message.substring(0, message.length() - END\_SEQUENCE.length());

}

## Cài đặt giao diện

### Giao diện giấu tin

A screenshot of a cell phone

Description automatically generated

### Giao diện lấy tin

A screenshot of a cell phone

Description automatically generated

### Giao diện giấu tập tin

A screenshot of a cell phone

Description automatically generated

### Giao diện lấy tập tin

A screenshot of a cell phone

Description automatically generated

### Giao diện phân tích giấu tin

A screenshot of a cell phone

Description automatically generated

## Kiểm nghiệm, đánh giá và phân tích

### Kiểm nghiệm

Thử nghiệm trên 10 tập tin video có định dạng .avi với độ dài khác nhau và các thông điệp ẩn giấu khác nhau.

### Đánh giá

Đối chiếu kết quả của các tập tin ban đầu và tập tin đã được nhúng thông điệp ẩn giấu:

A screenshot of a cell phone

Description automatically generated

* Dung lượng cua tập tin không bị thay đổi.
* File kết quả vẫn hoạt động bình thường, không có hiện tượng vỡ hình, vỡ tiếng.
* Bằng mắt thường không thể quan sát được sự khác biệt

### Phân tích

Sau khi giấu tin file sẽ có sự thay đổi các giá trị trong file. Để phân tích sự khác nhau giữa hai file để tiến tới phân tích phát hiện giấu tin có thể sử dụng các phương pháp:

* So sánh giá trị băm của hai file.

A screenshot of a cell phone

Description automatically generated

* Kiểm tra sự khác nhau các byte theo từng vị trí:

A screenshot of a social media post

Description automatically generated

* Biểu số giá trị tương ứng từng byte giữa 2 file: Giá trị các bit khác nhau không quá 1.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

# Kết luận

Có thể nói, từ khi ra đời, công nghệ giấu tin đã và đang làm rất tốt vai trò của nó trong nhiều lĩnh vực như bảo vệ an toàn thông tin trong quá trình trao đổi, bảo vệ quyền tác giả trong quá trình phân phối, ... Tuy nhiên bên cạnh đó, có những trường hợp lợi dụng kỹ thuật giấu tin để thực hiện những hành vi bất hợp pháp, gây nguy hiểm cho người khác. Chính vì vậy, việc nghiên cứu, tìm hiểu về các cách giấu tin cũng như các phương pháp phát hiện ngày càng đóng vai trò quan trọng trong ngành an toàn bảo mật thông tin. Qua bài tập lớp này, chúng em mong rằng mình đã trình bày được những kiến thực khái quát nhất về công nghệ giấu tin nói chung và kỹ thuật giấu tin trong video nói riwnwg, góp phần làm tiền đề cho những nghiên cứu sau này.

# Tài liệu tham khảo

* Bài giảng Các kỹ thuật giấu tin TS. Đỗ Xuân chợ

