**BAN CƠ YẾU CHÍNH PHỦ**

**HỌC VIỆN KỸ THUẬT MẬT MÃ**

**---------\*\*\*--------**

****

**BÁO CÁO BÀI TẬP**

**MÔN KỸ THUẬT LẬP TRÌNH**

**Đề Tài: Ẩn Thông Điệp Bên Trong Tệp Âm Thanh**

*Giảng viên hướng dẫn:* ***Đặng Xuân Bảo***

Khoa An toàn thông tin – Học viện Kỹ thuật mật mã

**Nhóm 8:**

**Nguyễn Đức Mạnh - AT170432**

**Nguyễn Bá Trung - AT170452**

**Nguyễn Duy Cương - AT170406**

**Nguyễn Đức Thắng - AT170445**

**Lê Trung Hiếu - AT170416**

**Hà Nội, tháng 3 năm 2023**

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 4](#_Toc108684670)

[**CHƯƠNG 1. TỔNG QUAN VỀ STEGANOGRAPH** 5](#_Toc108684671)

[1.1 Steganography là gì? 5](#_Toc108684672)

[1.2 Phân biệt Steganography và Cryptography. 5](#_Toc108684673)

[1.3 Lịch sử giấu tin 6](#_Toc108684674)

[1.4 Các thành phần của hệ giấu tin 6](#_Toc108684675)

[1.5 Các yêu cầu của hệ giấu tin 8](#_Toc108684676)

[1.5.1 Tính “vô hình” 8](#_Toc108684677)

[1.5.2 Khả năng chống giả mạo 8](#_Toc108684678)

[1.5.3 Tính bền vững 9](#_Toc108684679)

[1.5.4 Dung lượng tin giấu 9](#_Toc108684680)

[1.5.5 Độ phức tạp tính toán 9](#_Toc108684681)

[1.6 Phân loại giấu tin](#_Toc108684682) 9

[1.7 Tấn công hệ thống giấu tin](#_Toc108684683) 10

[1.8 Các ứng dụng của giấu tin 1](#_Toc108684684)1

[1.8.1 Liên lạc bí mật 1](#_Toc108684685)1

[1.8.2 Bảo vệ bản quyền (copyright protection) 1](#_Toc108684686)1

[1.8.3 Nhận thực hay phát hiện xuyên tạc thông tin (authentication and tamper detection) 1](#_Toc108684687)2

[1.8.4 Ghi dấu vân tay (fingerprinting) 1](#_Toc108684688)2

[1.8.5 Gán nhãn (labeling) 1](#_Toc108684689)2

[1.8.6 Điều khiển sao chép (copy control) 1](#_Toc108684690)2

[**CHƯƠNG 2. GIẤU THÔNG ĐIỆP TRONG FILE ÂM THANH** 1](#_Toc108684691)3

[2.1 Giấu tin trong âm thanh số 1](#_Toc108684692)3

[2.1.1 Ngưỡng nghe 1](#_Toc108684693)3

[2.1.2 Hiện tượng che khuất 1](#_Toc108684694)4

[2.1.3 Âm thanh và các đặc tính của âm thanh 1](#_Toc108684695)5

[2.1.4 Biểu diễn âm thanh số 1](#_Toc108684696)8

[2.1.5 Các định dạng âm thanh phổ biến](#_Toc108684697) 19

[2.1.6 Một số chương trình giấu tin trên âm thanh 2](#_Toc108684698)0

[2.2 Các kỹ thuật giấu tin phổ biến 2](#_Toc108684699)1

[2.2.1 Các kỹ thuật bổ trợ cho giấu tin 2](#_Toc108684700)1

[2.2.2 Các phép biến đổi từ miền thời gian sang miền tần số 2](#_Toc108684701)1

[2.2.3 Phép biến đổi Fourier rời rạc 2](#_Toc108684702)1

[2.2.4 Phép biến đổi wavelet (DWT) 2](#_Toc108684703)2

[2.2.5 Xáo trộn dữ liệu mật 2](#_Toc108684704)3

[2.2.6 Xáo trộn dữ liệu theo chuỗi hoán vị 2](#_Toc108684705)4

[2.2.7 Xáo trộn dữ liệu bằng phép biến đổi Arnold 2](#_Toc108684706)4

[2.2.8 Sinh chuỗi giả ngẫu nhiên 2](#_Toc108684707)5

[2.2.9 Bộ sinh đồng dư (congruential generator) 2](#_Toc108684708)6

[2.2.10 Bộ sinh Fibonacci 2](#_Toc108684709)6

[2.3 Đánh giá các phương pháp giấu tin trong âm thanh 2](#_Toc108684710)6

[2.3.1 Đánh giá bằng các độ đo 2](#_Toc108684711)6

[2.3.2 Đánh giá bằng các phầm mềm phát hiện tin 2](#_Toc108684711)7

[2.3.3 Đánh giá bằng bảng đánh giá ODG (Object Difference Grade) 27](#_Toc108684713)

[2.4 Phương pháp giấu tin trong âm thanh 29](#_Toc108684714)

[2.4.1 Phương pháp điều chỉnh bit ít quan trọng nhất (LSB coding) 29](#_Toc108684715)

[2.4.2 Phương pháp chẵn lẻ (parity coding) 3](#_Toc108684716)0

[2.4.3 Phương pháp mã hoá pha (phase coding) 3](#_Toc108684717)1

[2.4.4 Phương pháp mã hoá tiếng vọng (echo coding) 3](#_Toc108684718)3

[2.4.5 Phương pháp trải phổ 3](#_Toc108684719)5

[2.4.6 Phương pháp điều chỉnh biên độ 37](#_Toc108684720)

[2.4.7 Phương pháp lượng tử hoá (quantization) 39](#_Toc108684721)

[2.4.8 Phương pháp điều chế tỉ lệ thời gian 40](#_Toc108684722)

[2.4.9 Phương pháp giấu dựa vào Patchwork 4](#_Toc108684723)0

[2.4.10 Phương pháp dựa vào các đặc trưng nổi bật 4](#_Toc108684723)3

[**CHƯƠNG 3. THỰC NGHIỆM** 44](#_Toc108684724)

[3.1 Đặt Vấn Đề. 44](#_Toc108684725)

[3.2 Mã Hóa 45](#_Toc108684726)

[3.2.1 Nguyên lý hoạt động 45](#_Toc108684727)

[3.2.2 Source Code 46](#_Toc108684728)

[3.3 Giải Mã 48](#_Toc108684729)

[3.3.1 Nguyên lý hoạt động 48](#_Toc108684730)

[2.3.2. Code 48](#_Toc108684731)

[3.4 Demo 49](#_Toc108684732)

[**TỔNG KẾT** 50](#_Toc108684733)

# **LỜI MỞ ĐẦU**

Với sự hoàn hảo của các tiện ích đa phương tiện số và những tiến bộ của công nghệ, Steganography ngày càng phát triển và được ứng dụng khắp nơi. Trong lĩnh vực an toàn thông tin, nó được kết hợp với các phương pháp trong Cryptography để có thể chuyển các thông báo mật đi dưới một bức tranh, một bản nhạc vô hại, nhờ thế làm tăng thêm vòng bảo mật của chúng.

Trong lĩnh vực sở hữu trí tuệ, nó được xem như một cơ chế tuyệt hảo để chống gian lận, giả mạo và lạm dụng bản quyền. Các nhà công nghiệp đã đầu tư hàng triệu đô la vào các kỹ thuật che giấu các logo của các công ty và các dấu hiệu bản quyền khác trong các hình ảnh số, các bản ghi video và âm nhạc… Do đó, nhóm đã quyết định chọn và làm đề tài về giấu tin trong file âm thanh, từ đó chỉ ra được những ưu, nhược điểm và tính ứng dụng trong đời sống.

Nhóm xin chân thành cảm ơn sự giúp đỡ của thầy Đặng Xuân Bảo đã hướng dẫn và hỗ trợ nhóm trong quá trình làm đề tài!

1. **TỔNG QUAN VỀ STEGANOGRAPH**
   1. **Steganography là gì?**

Steganography là một kỹ thuật ẩn tin trong đó thông tin được ẩn trong một đối tượng khác mà không dễ dàng bị phát hiện. Steganography không giống như mật mã hóa, trong đó thông tin được mã hóa thành một dạng khác để tránh bị đánh cắp hoặc thay đổi, mà thay vào đó, thông tin được ẩn trong một hình ảnh, âm thanh, văn bản hoặc bất kỳ định dạng file nào khác mà không làm thay đổi sự xuất hiện của đối tượng đó.

**Ví dụ cụ thể**: Bạn có thể sử dụng kỹ thuật giấu thông điệp trong tệp văn bản. Bạn có thể chèn một thông điệp bí mật vào các khoảng trắng giữa các từ hoặc các dòng trống trong văn bản. Khi văn bản được đọc bình thường, người đọc sẽ không nhận ra sự thay đổi này.

Bạn có thể giấu một tin nhắn bí mật trong một hình ảnh. Để làm điều này, bạn có thể thay đổi một số điểm ảnh trong hình ảnh bằng cách tăng hoặc giảm độ sáng hoặc màu sắc một chút. Khi xem hình ảnh bình thường, không ai sẽ nhận ra sự thay đổi này. Tuy nhiên, nếu bạn biết cách, bạn có thể phân tích hình ảnh và khôi phục tin nhắn bí mật.

* 1. **Phân biệt Steganography và Cryptography.**

Steganography và Cryptography đều là các kỹ thuật được sử dụng để bảo vệ thông tin nhạy cảm, tuy nhiên, chúng khác nhau về cách thức hoạt động.

Steganography: là kỹ thuật giấu thông tin trong một tập tin khác mà không làm thay đổi sự xuất hiện của nó. Thông tin được ẩn trong một đối tượng khác, chẳng hạn như hình ảnh, văn bản, âm thanh hoặc video. Mục đích của Steganography là ẩn thông tin để người khác không nhìn thấy hoặc phát hiện được, nhưng khi thông tin được tìm thấy, nó không còn được bảo mật.

Cryptography: là kỹ thuật mã hóa thông tin thành một dạng khác để tránh bị đánh cắp hoặc thay đổi. Thông tin được mã hóa với một mã khóa duy nhất mà chỉ người có mã khóa mới có thể giải mã thông tin đó. Mục đích của Cryptography là bảo vệ thông tin bằng cách chuyển đổi nó thành một dạng không thể đọc được để ngăn người khác đọc được thông tin đó.

Ví dụ: Nếu bạn muốn truyền tin nhắn giữa hai người, bạn có thể sử dụng cryptography để mã hóa tin nhắn để người khác không thể đọc được nó. Trong khi đó, nếu bạn muốn giữ cho tin nhắn bí mật và không muốn ai biết rằng có tin nhắn được truyền đi, bạn có thể sử dụng Steganography để giấu tin nhắn trong một tập tin hình ảnh hoặc âm thanh để người khác không phát hiện được.

* 1. **Lịch sử giấu tin**

Lịch sử của giấu tin hoặc steganography rất lâu đời, có thể theo dõi ngược lại đến thời kỳ cổ đại. Những người xưa sử dụng nhiều phương pháp khác nhau để giấu tin nhắn như viết trên bề mặt da động vật, tráo đổi vị trí các chữ cái trong văn bản để tạo ra một tin nhắn mới, hoặc sử dụng các dấu hiệu như đốt lửa hoặc tiếng trống để gửi tin nhắn cho những người khác.

Trong thời kỳ Trung cổ, các kỹ thuật giấu tin đã được phát triển và sử dụng rộng rãi. Một trong những kỹ thuật được sử dụng nhiều trong thời kỳ này là viết bằng mực dưới đáy giấy, sau đó bôi mực lên bề mặt giấy để che giấu tin nhắn. Ngoài ra, các nhà kinh tế của thời kỳ này cũng sử dụng giấu tin để bảo vệ các giao dịch thương mại quan trọng.

Trong thế kỷ 20, kỹ thuật giấu tin đã tiến bộ nhanh chóng với sự phát triển của công nghệ số. Trong thời đại này, các kỹ thuật giấu tin đã được sử dụng rộng rãi trong lĩnh vực quân sự để giữ bí mật các thông tin quân sự quan trọng. Các kỹ thuật mới như giấu tin trong hình ảnh và âm thanh đã được phát triển và sử dụng rộng rãi trong các cuộc điều tra tình báo.

Ngày nay, steganography được sử dụng rộng rãi trong các lĩnh vực khác nhau, bao gồm an ninh mạng, tài chính, truyền thông, giải trí và nhiều lĩnh vực khác nữa.

* 1. **Các thành phần của hệ giấu tin**

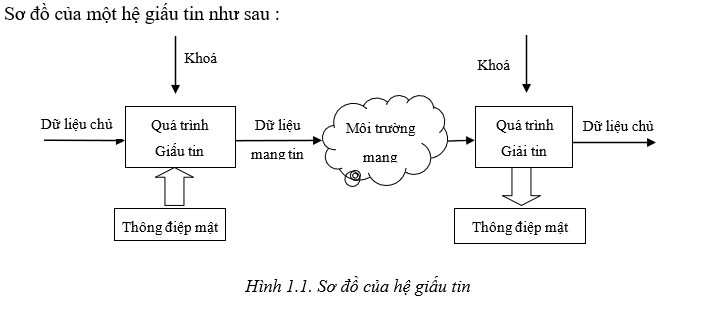
Một hệ giấu tin gồm có các thành phần chính sau đây:

+ Thông điệp mật (secret message): có thể là văn bản, hình ảnh, âm thanh. Trong quá trình giấu tin, chúng được chuyển thành chuỗi các bit.

+ Dữ liệu chủ hay môi trường sẽ chứa tin mật (host signal): dữ liệu có thể chọn làm dữ liệu chủ là tệp ảnh, tệp âm thanh, tệp video hay tệp cơ sở dữ liệu, văn bản.

+ Khoá K: là khóa tham gia vào quá trình giấu tin để tăng tính bảo mật.

+ Dữ liệu mang tin hay môi trường đã chứa tin mật (stego signal)



**1.5 Các yêu cầu của hệ giấu tin**

Có nhiều phương pháp giấu tin cho những kiểu dữ liệu khác nhau. Để đánh giá chất lượng của một phương pháp giấu tin, người ta dựa vào một số tiêu chí sau:

**1.5.1 Tính vô hình:**

Tính vô hình là một trong những đặc điểm quan trọng của hệ giấu tin, cho phép thông tin được giấu đi một cách tự nhiên và không bị phát hiện bởi những người không được cho phép truy cập thông tin đó. Các hệ thống giấu tin vô hình được thiết kế để giấu thông tin trong các đối tượng khác, như hình ảnh, văn bản hoặc âm thanh, mà không làm thay đổi sự xuất hiện của đối tượng đó. Những người không có kiến thức về việc giấu tin sẽ không thể nhận ra sự hiện diện của thông tin giấu trong đối tượng đó.

Các hệ thống giấu tin vô hình phụ thuộc vào các thuật toán và kỹ thuật tinh vi để chèn thông tin vào các đối tượng một cách bí mật và hiệu quả. Ví dụ, để giấu tin trong hình ảnh, các pixel trong hình ảnh sẽ được thay đổi một cách nhỏ để tạo ra một mẫu mới và đặc biệt, điều này sẽ không ảnh hưởng đến chất lượng của hình ảnh ban đầu.

**1.5.2** **Khả năng chống giả mạo**

- Khả năng chống giả mạo của hệ giấu tin phụ thuộc vào cách thức mà thông tin được giấu và kiểm tra tính toàn vẹn của thông tin. Nếu thông tin được giấu một cách an toàn và đáng tin cậy, thì khả năng chống giả mạo của hệ giấu tin cũng sẽ tăng lên.

-Một số kỹ thuật giấu tin được thiết kế để kiểm tra tính toàn vẹn của thông tin, ví dụ như sử dụng mã hóa bảo vệ hoặc mã băm (hash) để đảm bảo rằng thông tin không bị thay đổi sau khi được giấu. Khi thông tin được giải mã, hệ thống sẽ kiểm tra tính toàn vẹn của thông tin bằng cách so sánh với giá trị mã hóa hoặc mã băm được tính toán trước đó.

**1.5.3 Tính bền vững**

Sau khi tin được giấu vào đối tượng chứa, các tệp mang tin được truyền qua các kênh công cộng và có thể phải qua hay chịu các biến đổi khác nhau, chẳng hạn như phép lọc, thêm nhiễu, cắt bớt dữ liệu hay lấy mẫu lại. Tính bền vững thể hiện ở sự nguyên vẹn, không thay đổi (hay thay đổi không đáng kể) của thông tin được giấu sau những biến đổi như vậy.

* + 1. **Dung lượng tin giấu**

Dung lượng tin giấu được tính bằng tỷ lệ của lượng tin cần giấu so với kích thước đối tượng gốc. Các phương pháp giấu tin mật đều cố gắng giấu được càng nhiều tin trong dữ liệu gốc càng tốt nhưng vẫn giữ được bí mật.

* + 1. **độ phức tạp tính toán**

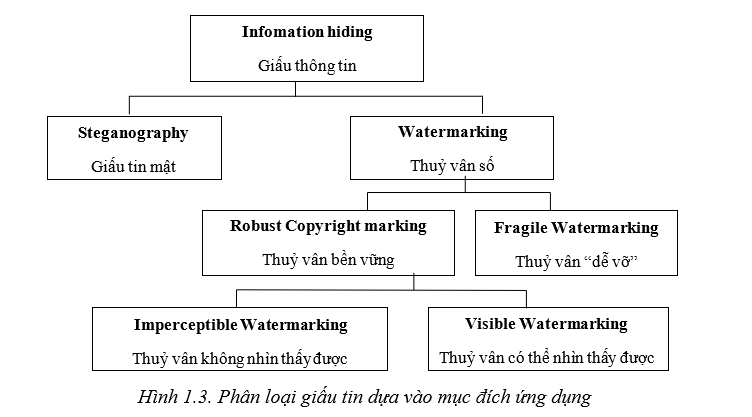
Độ phức tạp tính toán của hệ giấu tin phụ thuộc vào nhiều yếu tố như số lượng và độ phức tạp của các thành phần trong hệ thống, cách thức xử lý dữ liệu, số lượng người dùng sử dụng cùng lúc và tần suất truy cập, độ tin cậy và độ an toàn của hệ thống.

Trong hệ giấu tin, các phép tính thường được thực hiện trên các bit dữ liệu và được thực hiện trên các thiết bị tính toán đặc biệt như các vi xử lý và FPGA. Do đó, độ phức tạp tính toán của hệ giấu tin thường được đo bằng số lượng thao tác logic và các lần truy cập bộ nhớ.

Ngoài ra, các thuật toán mã hóa được sử dụng trong hệ giấu tin cũng có ảnh hưởng đến độ phức tạp tính toán. Các thuật toán mã hóa phức tạp hơn yêu cầu nhiều thời gian và tài nguyên tính toán hơn để thực hiện.

* 1. **Phân loại giấu tin**

Giấu tin có thể phân loại theo đặc trưng, tính chất và ứng dụng. Ta có thể phân loại các kỹ thuật giấu tin theo kiểu tài liệu dùng làm dữ liệu chứa (văn bản, ảnh, âm thanh, video..), theo miền xử lý (miền dữ liệu trực tiếp hay miền biến đổi), hay theo mục đích giấu tin.



**1.7 Tấn công hệ thống giấu tin**

Tấn công hệ thống giấu tin là các kỹ thuật tấn công được sử dụng để phá vỡ hoặc tìm ra thông tin được giấu trong hệ thống giấu tin. Các tấn công này có thể được phân loại thành hai loại chính:

Tấn công không phá hủy: Các kỹ thuật này nhằm vào việc truy xuất thông tin được giấu mà không ảnh hưởng đến nội dung của dữ liệu gốc. Ví dụ như kỹ thuật "steganalysis" được sử dụng để phát hiện và trích xuất thông tin giấu tin trong hình ảnh, âm thanh hoặc văn bản. Kỹ thuật này sử dụng các phương pháp phân tích thống kê và xác suất để tìm ra các đặc trưng và mẫu khác biệt giữa dữ liệu gốc và dữ liệu giấu tin.

Tấn công phá hủy: Các kỹ thuật này nhằm vào việc phá vỡ hệ thống giấu tin và phá hủy dữ liệu gốc. Ví dụ như kỹ thuật "steganography cracking" được sử dụng để tìm ra vị trí và nội dung của dữ liệu giấu tin, từ đó phá vỡ hệ thống giấu tin và truy cập được vào dữ liệu gốc. Kỹ thuật này sử dụng các phương pháp tấn công vét cạn (brute-force attack), tấn công đánh đố (guessing attack), tấn công phá mã (cryptography attack) hoặc tấn công mạng (network attack) để tìm ra thông tin giấu tin và phá hủy dữ liệu gốc.

Để bảo vệ hệ thống giấu tin khỏi các tấn công này, cần sử dụng các kỹ thuật bảo mật và phát triển các hệ thống giấu tin mới với tính bảo mật cao hơn, sử dụng các phương pháp giấu tin phức tạp hơn và khó bị phát hiện. Các kỹ thuật giám sát và phát hiện tấn công cũng cần được áp dụng để giám sát và phát hiện các hành vi tấn công đối với hệ thống giấu tin.

* 1. **Các ứng dụng của giấu tin**
     1. Chart, histogram

        Description automatically generated**Liên lạc bí mật**

Giấu tin có thể được sử dụng để liên lạc bí mật giữa người gửi và người nhận. Việc sử dụng mã hóa để giữ thông tin bí mật có thể gây chú ý cho người thám mã. Vì vậy, người ta sử dụng các phương pháp ẩn tin để truyền thông tin bí mật trên mạng máy tính. Cụ thể, mật khẩu sẽ được nén, mã hóa và ẩn mã vào môi trường để tránh sự chú ý của người thám mã. Việc kết hợp nhiều giải pháp giúp tăng tính bảo mật và mức độ an toàn trong quá trình truyền thông tin mật trên mạng công khai.

* + 1. **Bảo vệ bản quyền (copyright protection)**

Đây là ứng dụng cơ bản nhất của kỹ thuật thuỷ vân số. Theo đó, một thông tin nào đó sẽ được đưa vào trong sản phẩm để chứng minh quyền sở hữu của tác giả. Tuy nhiên, việc nhúng này cần phải đảm bảo không ảnh hưởng lớn đến sản phẩm. Điều kiện kỹ thuật đối với ứng dụng này là thủy vân phải tồn tại bền vững cùng với sản phẩm. Nếu muốn loại bỏ thủy vân mà không được phép của chủ sở hữu, thì sản phẩm sẽ phải bị phá hủy.

* + 1. **Nhận thực hay phát hiện xuyên tạc thông tin (authentication and tamper detection)**

Đối với ứng dụng này, một tập hợp các thông tin sẽ bị ẩn trong dữ liệu chứa và sau đó được sử dụng để biết rằng dữ liệu chứa có bị thay đổi hay không. Yêu cầu đối với ứng dụng này là có khả năng ẩn nhiều thông tin và thủy vân không cần phải tồn tại lâu dài trước khi thực hiện các phép xử lý trên các đối tượng đã được ẩn.

* + 1. **Ghi dấu vân tay (fingerprinting)**

Mục tiêu của dấu vân tay là để chuyển thông tin về người nhận sản phẩm phương tiện tiện ích xác định đây là bản sao duy nhất của sản phẩm. Để thực hiện điều này, mỗi sản phẩm sẽ có một thủy vân riêng được nhúng vào các bản sao khác nhau của thông tin gốc trước khi chuyển cho nhiều người. Yêu cầu đối với ứng dụng này là đảm bảo mức độ an toàn cao cho các thủy vân, tránh sự cố xóa dấu vết khi phân phối. Ghi dấu vân tay tương tự như số xê ri của phần mềm.

* + 1. **Gán nhãn (labeling)**

Kỹ thuật này cho phép nhúng các thông tin như tiêu đề, chú giải, nhãn thời gian và các họa tiết minh hoạ khác vào trong các đối tượng chứa, ví dụ như đính kèm tên người lên ảnh hoặc đính kèm tên vùng địa phương lên bản đồ. Khi sao chép các đối tượng này, các thông tin nhúng trong nó cũng sẽ được sao chép. Chỉ có chủ sở hữu của tác phẩm mới có thể tách ra và xem các giải pháp này. Ngoài ra, việc gán nhãn cũng được sử dụng để nhúng các từ khóa vào trong các đối tượng chứa để các công cụ tìm kiếm có thể tìm kiếm nhanh các đối tượng đó.

* + 1. **Điều khiển sao chép (copy control)**

Sao chép điều khiển (điều khiển sao chép) là một kỹ thuật được sử dụng để giới hạn hoặc ngăn chặn việc sao chép trái cho phép dữ liệu đa phương tiện. Kỹ thuật này có thể sử dụng các giá trị thủy vân để xác định trạng thái sao chép như "không bao giờ sao chép" hoặc "sao chép một lần", đồng thời có thể sử dụng các thiết bị đọc ghi để phát và ghi lại các giá trị thủy vân để đảm bảo tính toàn vẹn của dữ liệu. Tuy nhiên, để đảm bảo hiệu quả tính toán của kỹ thuật này, cần phải chắc chắn rằng thủy vân được bảo vệ và sử dụng các phương pháp phát hiện thủy vân đã ẩn để tránh bị đánh cắp hoặc giả mạo.

**CHƯƠNG 2. GIẤU THÔNG ĐIỆP TRONG FILE ÂM THANH**

**2.1 Giấu tin trong âm thanh số**

Ngay từ những ngày đầu phát triển, các thuật toán giấu tin hầu hết chỉ tập trung nghiên cứu nhiều lĩnh vực ảnh. Khác với giấu tin trong ảnh là khai thác hệ thống thị giác của con người, giấu tin trên âm thanh tập trung khai thác khả năng cảm nhận của hệ thính giác của con người (Human Auditory System).

Diagram

Description automatically generated

*Hình 1.4. Hệ thống thính giác của con người*

Hình 1.4 mô tả cấu trúc hệ thống thính giác của con người. Các tín hiệu âm thanh từ môi trường bên ngoài sẽ được thu nhận bởi tai ngoài, sau đó các tín hiệu này chuyển đến màng nhĩ và chuyển đến các tế bào thần kinh thông qua các cấu trúc bên trong của tai.

Theo các nghiên cứu về sinh học, hệ thính giác người nhạy cảm hơn nhiều so với hệ thống thị giác và các tấn công trên âm thanh số cũng đa dạng hơn nên giấu tin trên âm thanh không phải là lĩnh vực dễ thực hiện. Giấu tin trong âm thanh dựa vào ngưỡng nghe của người và hiện tượng che khuất (masking) âm thanh để giấu tin.

### **Ngưỡng nghe**

Phạm vi nghe của tai người trong khoảng từ 20Hz đến 20 kHz nhưng nghe rõ nhất đối với âm thanh trong phạm vi từ 1kHz đến 4kHz. Ngưỡng nghe của con người thay đổi theo độ tuổi và có sự khác nhau giữa nam và nữ. Hình 1.5 thể hiện dải tần của âm thanh. Những âm thanh trong khoảng từ 0 - 20Hz gọi là hạ âm, những âm thanh trên 20 KHz gọi là siêu âm. Con người không nghe được hạ âm và siêu âm.

### **Hiện tượng che khuất**

Mặc dù tai người có thể nghe ở dải tần rộng nhưng những gì tai người có thể nghe phụ thuộc vào môi trường nơi đang nghe. Nếu như ở trong môi trường nhiễu mạnh (ví dụ nhiễu trắng), nhiều âm thanh nhỏ sẽ bị che khuất (masked), vì thế ta không thể nghe được chúng.

Hiện tượng che khuất tín hiệu âm thanh (auditory masking) xảy ra khi một âm thanh này ảnh hưởng đến sự cảm nhận một âm thanh khác. Âm thanh bị che gọi là maskee, âm thanh che là masker. Khoảng cường độ khác nhau giữa maskee và masker gọi là mức độ che.

Diagram, schematic

Description automatically generated

*Hình 1.6 Ngưỡng che của tín hiệu âm thanh*

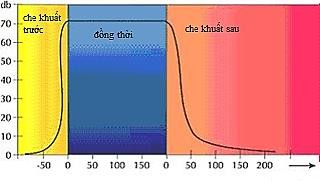
Ví dụ ban đầu ta có thể nghe âm thanh ở mức 10 dB. Khi có sự xuất hiện đồng thời một âm thanh khác thì ta không thể nghe được âm thanh ban đầu trừ khi ta điều chỉnh âm thanh ban đầu lên 26 dB. Như vậy khoảng bị che (masked threshold) của âm thanh ban đầu so với âm thanh che là 16 dB. Hình 1.6 minh hoạ cho ngưỡng che của tín hiệu âm thanh.

Diagram

Description automatically generated

*Hình 1.7. Che khuất âm thanh trên miền tần số*

Hiện tượng che khuất âm thanh xảy ra trên cả miền tần số và miền thời gian. Hình 1.7 minh họa cho che khuất âm thanh trong miền tần số (simultaneous masking) và hình 1.8 minh họa cho che khuất âm thanh trong miền thời gian (temporal masking).



*Hình 1.8. Che khuất âm thanh trên miền thời gian*

### **Âm thanh và các đặc tính của âm thanh**

Âm thanh là các [dao động](https://vi.wikipedia.org/wiki/Dao_%C4%91%E1%BB%99ng) [cơ học](https://vi.wikipedia.org/wiki/C%C6%A1_h%E1%BB%8Dc) của các [phân tử](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%E1%BB%AD), [nguyên tử](https://vi.wikipedia.org/wiki/Nguy%C3%AAn_t%E1%BB%AD) hay các hạt làm nên [vật chất](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_ch%E1%BA%A5t) và lan truyền trong vật chất như các [sóng](https://vi.wikipedia.org/wiki/S%C3%B3ng). Âm thanh, giống như nhiều sóng, được đặc trưng bởi [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91), [bước sóng](https://vi.wikipedia.org/wiki/B%C6%B0%E1%BB%9Bc_s%C3%B3ng), [chu kỳ](https://vi.wikipedia.org/wiki/Chu_k%E1%BB%B3), [biên độ](https://vi.wikipedia.org/wiki/Bi%C3%AAn_%C4%91%E1%BB%99) và [vận tốc](https://vi.wikipedia.org/wiki/V%E1%BA%ADn_t%E1%BB%91c) lan truyền. Ba đặc trưng sinh lý của âm thanh là âm lượng, độ cao, âm sắc. Âm lượng đại diện cho độ lớn (loudness) của tín hiệu âm thanh, tương quan với biên độ của tín hiệu, cũng được xem là năng lượng hoặc cường độ của tín hiệu âm thanh. Độ cao đại diện cho tỷ lệ rung của tín hiệu âm thanh, có thể biểu diễn bởi tần số cơ bản hoặc tương đương. Âm sắc đại diện cho nội dung ngữ nghĩa của tín hiệu âm thanh, giúp ta phân biệt âm do các nguồn khác nhau phát ra.

Đơn vị để đo âm thanh là decibel (dB). Decibel được dùng để so sánh độ mạnh của 2 âm thanh. Giá trị của này thể hiện tỉ lệ (ratio) chứ không phải là một giá trị cụ thể.



Trong đó PRL là cường độ (strength) của tín hiệu cần so sánh và P1 là cường độ của tín hiệu được so sánh.

Ví dụ: P1=3.1\*PRL thì:

dB =20\*log10(3.1PRL/PRL)

=20\*log10(3.1) =20\*0.49993 =9.827 ~ 10.

Các đặc tính của âm thanh:

* Âm thanh có thể dùng máy biến năng chuyển âm thanh từ dạng này sang dạng năng lượng khác để lưu trữ.
* Cường độ bức xạ của âm thanh giảm dần tỉ lệ với bình phương khoảng cách với nguồn.

Khi làm việc trong môi trường âm thanh, ngoài thông tin số thể hiện âm thanh chúng ta còn phải chú ý đến phương tiện truyền dẫn tín hiệu âm thanh đó. Phương tiện truyền dẫn một tín hiệu âm thanh là môi trường mà trong đó tín hiệu phải đi qua để đến được đích. Có thể chia môi trường truyền dẫn ra thành 4 nhóm như sau :

* Môi trường số end – to – end: trong đó các tệp âm thanh được copytrực tiếp từ máy này sang máy khác.
* Môi trường lấy mẫu lại theo tỉ lệ khác: trong đó tín hiệu được lấy mẫu lại theo tỉ lệ lấy mẫu cao hoặc thấp hơn.
* Môi trường truyền dẫn analog và lấy mẫu lại : trong đó tín hiệu được chuyển sang analog sau đó lấy mẫu lại.
* Môi trường "over the air": tín hiệu được truyền qua không khí và được lấy mẫu lại bằng micro.

Tùy vào môi trường truyền tín hiệu âm thanh mà có các kỹ thuật thích hợp để nhúng dữ liệu vào tệp âm thanh này.

Chất lượng tín hiệu có thể giảm sút khi truyền trên đường truyền. Ở dạng tương tự hay dạng số, chúng ta đều có những cách để làm giảm sự hao hụt tín hiệu trên đường truyền. Hình 1.9 minh hoạ cho các kỹ thuật làm giảm sự hao hụt tín hiệu trong trường hợp tín hiệu tương tự và tín hiệu số.

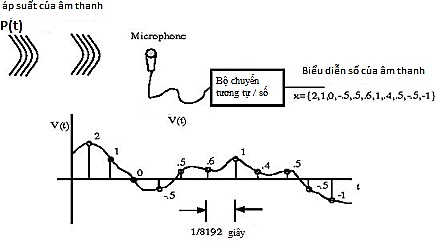
Diagram

Description automatically generated

*Hình 1.9. Kỹ thuật làm giảm hao hụt tín hiệu trên đường truyền*

### **Biểu diễn âm thanh số**

Để lưu trữ và xử lý âm thanh bằng máy tính, các tín hiệu âm thanh cần phải được chuyển từ dạng tương tự sang dạng số. Thông tin số đại diện của một tệp âm thanh bao gồm các mẫu tín hiệu được lấy theo phương pháp lượng tử hóa và lấy mẫu theo thời gian. Âm thanh thường được lượng tử hóa với độ sâu số (bit depth) 8, 16 hoặc 24 bit. Đối với âm thanh chất lượng cao, mức lượng tử thường được sử dụng là 16 bit. Tỷ lệ lấy mẫu theo thời gian quyết định biên vùng tần số. Tỉ lệ lấy mẫu thông thường nằm trong khoảng 8 kHz đến 44.1 kHz



*Hình 1.10. Chuyển âm thanh dạng tương sang dạng số*

Hình 1.11 thể hiện chi tiết quá trình lấy mẫu, lượng tử hoá khi chuyển tín hiệu ở dạng tương tự sang dạng số. Lỗi lượng tử (quantization error) có thể giảm xuống bằng cách tăng tỉ lệ lấy mẫu (sampling rate) và dùng nhiều bit hơn để biểu diễn giá trị một mẫu (quantization level).

Chart, histogram

Description automatically generated

*Hình 1.11. Lượng tử hoá và biểu diễn dạng số tín hiệu tương tự*

Một mẫu dữ liệu âm thanh có thể biểu diễn bằng 8 bit, 16 bit hay 24 bit và tệp âm thanh có thể gồm 1 kênh (mono) hay 2 kênh (stereo). Số byte dùng để lưu 1 giây âm thanh được xác định theo công thức (1.2):

Số byte ≈ (tần số lấy mẫu ∗ số kênh ∗ số bit biểu diễn 1 mẫu)/8 (1.2)

Ví dụ với âm thanh stereo, dùng 16 bit để lưu 1 mẫu, tần số lấy mẫu 44 kHz, mã hoá PCM, thì số byte cần dùng để lưu 3 phút sẽ là:

≈ 3min\*60sec/min\*44000samples/sec\*16bits/sample\*2(channel)/8bits/byte

≈ 31.68 Mb.

**2.1.5: Các định dạng âm thanh phổ biến**

Do các thuật toán giấu tin phụ thuộc vào định dạng của tệp âm thanh, nên để giấu tin vào tệp âm thanh, ta cần hiểu rõ định dạng tệp âm thanh. Có thể phân các định dạng thành âm thanh thành ba nhóm, đó là nhóm định dạng không nén, nhóm định dạng nén không mất mát thông tin và nhóm định dạng nén mất mát thông tin.

1.MP3: là định dạng âm thanh kỹ thuật số phổ biến nhất, được sử dụng để lưu trữ và chia sẻ âm nhạc trên internet.

2.WAV: định dạng âm thanh không nén dữ liệu, thường được sử dụng trong các dự án âm nhạc chuyên nghiệp.

3.FLAC: định dạng âm thanh nén không mất mát, được sử dụng để lưu trữ âm nhạc với chất lượng cao nhưng kích thước file vẫn nhỏ.

4.AAC: định dạng âm thanh kỹ thuật số được sử dụng phổ biến trong các thiết bị di động và trên internet.

5.OGG: định dạng âm thanh được sử dụng trong các trình phát đa phương tiện miễn phí như VLC Media Player.

6.AIFF: định dạng âm thanh không nén, thường được sử dụng trong các dự án âm nhạc chuyên nghiệp trên hệ điều hành macOS.

**2.1.6:Một số chương trình giấu tin trên âm thanh**

Đã có nhiều phần mềm giấu tin trong âm thanh được phát triển và ứng dụng trong thực tế. Bảng 1.2 thể hiện một số chương trình giấu hiện tại trong khảo sát của Pedram Hayati và các cộng sự. Ngoài giấu tin trên âm thanh, một số phần mềm có thể giấu trên nhiều loại dữ liệu và chạy trên nhiều hệ điều hành khác nhau. Ví dụ như OpenPuff có thể giấu trên ảnh, video, âm thanh và tệp pdf.

*Bảng 1.2. Bảng một số phần mềm giấu tin trong âm thanh*

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm giấu tin** | **Định dạng hỗ trợ** | **Dạng bản quyền** |
| Info Stego | mp3 | Shareware |
| ScramDisk | wav | Shareware |
| MP3Stego | mp3 | Mã nguồn mở |
| StegoWav | wav | Mã nguồn mở |
| Hide4PGP | mp3, voc | Mã nguồn mở |
| Steghide | wav, au | Mã nguồn mở |
| S-Tool | wav | Mã nguồn mở |
| Invisible Secrets | wav | Thương mại |
| Steganos | wav, voc | Thương mại |

**2.2 Các kỹ thuật giấu tin phổ biến**

**2.2.1: Các kỹ thuật bổ trợ cho giấu tin**

Để giấu tin hiệu quả, thông thường ta cần dùng các kỹ thuật bổ trợ. Các kỹ thuật này bao gồm chuyển dữ liệu âm thanh từ miền thời gian sang miền tần số, kỹ thuật sinh các chuỗi số ngẫu nhiên và các kỹ thuật thay đổi thứ tự giấu các bit trong chuỗi bit mật.

**2.2.2: Các phép biến đổi từ miền thời gian sang miền tần số**

Thay vì xử lý trực tiếp giá trị các mẫu âm thanh trên miền thời gian, chúng ta có thể sử dụng các kỹ thuật biến đổi để chuyển đổi sang các miền khác với các biến số mới.Việc sử dụng phương pháp khảo sát gián tiếp này giúp đơn giản hóa nhiều công việc trong quá trình xử lý âm thanh, so với việc sử dụng phương pháp khảo sát trực tiếp trong miền biến số độc lập.Các biến đổi được sử dụng để chuyển đổi miền biến số độc lập sang các miền khác có các biến số mới, và có nhiều cách biến đổi khác nhau. Có nhiều cách biến đổi, trong đó cách biến đổi hay được sử dụng là biến đổi Fourier, biến đổi wavelet…

**2.2.3: Phép biến đổi Fourier rời rạc**

Phép biến đổi Fourier rời rạc (DFT) là một phương pháp phân tích tín hiệu, được sử dụng để chuyển đổi một chuỗi các số thực hoặc số phức từ miền thời gian sang miền tần số. Điều này có ý nghĩa là nó có thể phân tích một tín hiệu âm thanh thành các thành phần tần số khác nhau. Biến đổi này là một công cụ quan trọng trong xử lý tín hiệu và được sử dụng rộng rãi trong các ứng dụng âm thanh và hình ảnh.

Tuy nhiên, tính toán trực tiếp DFT cho các chuỗi số lớn có thể rất tốn kém và không hiệu quả. Để giải quyết vấn đề này, thuật toán biến đổi Fourier nhanh (FFT) được phát triển. FFT là một thuật toán tính toán DFT một cách hiệu quả, với độ phức tạp tính toán là O(nlogn) thay vì O(n2) như tính toán DFT trực tiếp.

**2.2.4: Phép biến đổi wavelet(DWT)**

Phép biến đổi Fourier là một công cụ quan trọng trong việc xử lý tín hiệu số để tách và phân tích các tín hiệu thành các tần số khác nhau. Tuy nhiên, phép biến đổi này chỉ cho chúng ta thông tin toàn cục về tín hiệu và không thể xử lý các tín hiệu không tuần hoàn hoặc có các đột biến không dự báo được. Để khắc phục hạn chế này, người ta đã phát triển phép biến đổi wavelet.

Phép biến đổi wavelet được tính dựa trên phân tích đa phân giải, trong đó người dùng sử dụng các kỹ thuật lọc số để phân tích tín hiệu. Mỗi một tín hiệu được phân tích thành hai thành phần: thành phần xấp xỉ A (Approximation) tương ứng với thành phần tần số thấp và thành phần chi tiết D (Detail) tương ứng với thành phần tần số cao, thông qua hai bộ lọc thông thấp và thông cao, trong đó, bộ lọc thông cao sử dụng hàm wavelet và bộ lọc thông thấp sử dụng hàm tỉ lệ.

Graphical user interface

Description automatically generated with medium confidence

*Hình 2.1. Phân giải tín hiệu thành 2 thành phần xấp xỉ và chi tiết*

Hình 2.1 minh họa cho việc phân tích tín hiệu âm thanh thành hai thành phần xấp xỉ và chi tiết. Chuỗi tín hiệu gốc là dữ liệu của một tệp âm thanh gồm 1422720 mẫu sẽ được phân tích thành hai dãy A và D có độ dài gồm 711363 mẫu.

Trong hai thành phần A và D, thành phần D không quan trọng. Những thay đổi trên thành phần này ít ảnh hưởng đến tín hiệu khi ta khôi phục lại. Ví dụ nếu thay đổi thành phần D=0 (loại bỏ thành phần D), thực hiện biến đổi ngược lại, ta sẽ có dãy tín hiệu như hình 2.2.

A picture containing text, device

Description automatically generated

*Hình 2.2. Tín hiệu gốc và tín hiệu đã loại bỏ thành phần D*

**2.2.5: Xóa trộn dữ liệu mật**

Giả sử có chuỗi M gồm n phần tử, là thông điệp mật cần giấu vào trong tệp âm thanh. Thay vì giấu tuần tự từng phần tử từ M1 đến Mn, ta thực hiện xáo trộn chuỗi M rồi mới đem giấu vào tệp âm thanh. Mục đích của việc này là để gây khó khăn cho người thám tin khi thực hiện tấn công rút trích tin. Nếu người thám tin biết thuật toán giấu thì rất có thể dò ra được chuỗi tin giấu. Xáo trộn dữ liệu đòi hỏi người thám tin phải sắp xếp lại chuỗi tin nhận được theo trật tự đúng mới biết được tin, nếu không, chuỗi tin nhận được cũng không có ý nghĩa. Dưới đây là hai thuật toán dùng để xáo trộn dữ liệu được sử dụng trong các chương sau của luận án.

**2.2.6: Xáo trộn dữ liệu theo chuỗi hoán vị**

Nếu chuỗi tin mật cần giấu có độ dài n (đánh số từ 1 đến n) thì sẽ có n! hoán vị của nó. Ta có thể hoán vị chuỗi mật này trước khi giấu. Có nhiều kỹ thuật tạo hoán vị của tập {1,2,..., n}. Một trong các cách hiệu quả là dùng phương pháp sinh theo kiểu thứ tự từ điển.

Ví dụ ta có chuỗi cần giấu là “axnd”. Ta gán cho các ký tự chuỗi giá trị theo đúng thứ tự xuất hiện của chúng là {1, 2, 3, 4}.

Chuỗi này sẽ có 24 hoán vị là {1, 2, 3, 4}, {1, 2, 4, 3},{1, 3, 2, 4},

{1, 3, 4,2},{1, 4, 2, 3},{1, 4, 3, 2},{2, 1, 3, 4},{2, 1, 4, 3},{2, 3, 1, 4},

{2, 3, 4, 1},{2, 4, 1,3},{ 2, 4, 3, 1},{3, 1, 2, 4},{3, 1, 4, 2},{3, 2, 1, 4},

{3, 2, 4, 1},{3, 4, 1, 2},{3, 4, 2,1},{4, 1, 2, 3},{4, 1, 3, 2, 4},{4, 2, 1, 3},

{4, 2, 3, 1},{4, 3, 1, 2},{4, 3, 2, 1}.

Khi chọn k là 1 giá trị nào đó trong n! giá trị thì ta sẽ chọn hoán vị thứ k của chuỗi ban đầu và đem chuỗi đó đi giấu thay vì chuỗi nguyên gốc (k=1). Trong trường hợp giá trị k lớn hơn n! thì k sẽ được gán bằng (k mod n!) + 1.

Ví dụ k= 20 thì chuỗi giấu là: “danx” tương ứng với hoán vị {4, 1, 3, 2}.

Khi n lớn thì việc sinh chuỗi hoán vị thứ k (lớn) có thể tốn thời gian. Có nhiều cách để sinh ra hoán các hoán vị này. Một thuật toán hiệu quả để sinh ra hoán vị thứ k của tập n phần tử được trình bày trong.

**2.2.7:Xóa trộn dữ liệu bằng phép biến đổi Arnold**

Biến đổi Arnold là kỹ thuật biến đổi dùng để xáo trộn các điểm ảnh. Gọi I(x, y) là điểm ảnh ban đầu ở toạ độ (x, y). Sau khi thực hiện phép chuyển đổi, điểm ảnh sẽ được chuyển đến toạ độ mới là (x’, y’). Công thức xác định (x’, y’) như sau:



trong đó x, y ∈ {0,1, ..., N-1} và x’, y’ ∈ {0,1, ..., N-1}.

*Bảng 2.1. Chu kỳ lặp lại của phép biến đổi tương ứng với kích thước N*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 2 | 3 | 6 | 10 | 16 | 20 | 32 | 64 | 80 | 100 |
| Chu kỳ | 3 | 4 | 12 | 30 | 12 | 30 | 24 | 36 | 60 | 150 |

Nếu thực hiện một số lần biến đổi, ảnh biến đổi sẽ trở lại ảnh ban đầu. Chu kỳ của phép biến đổi tương ứng với kích thước khối ảnh N được thể hiện trong bảng 2.1.

|  |  |  |
| --- | --- | --- |
|  |  |  |



*Hình 2.3. Ảnh gốc và sau khi thực hiện chuyển đổi.*

Hình 2.3 là minh hoạ cho ảnh gốc là logo Đại học Công nghệ, Đại học Quốc gia Hà Nội, kích thước 80\*80 và hình ảnh xáo trộn ở các bước 1, 2, 6, 12 (dòng trên, từ trái qua phải) và 20, 30, 40, 58, 59 (dòng dưới, từ trái qua phải) sử dụng phép biến đổi trong .

Ảnh có kích thước n=80, chu kỳ là 60. Nếu lấy ảnh ở lần chuyển 59 và thực hiện thêm 1 lần nữa thì sẽ được ảnh ban đầu.

**2.2.8: Sinh chuỗi ngẫu nhiên**

Các phương pháp giấu tin trong audio thường dùng kết hợp với các chuỗi ngẫu nhiên để thực hiện giấu tin. Chuỗi ngẫu nhiên này có thể dùng để xác định ví trí mẫu được chọn để giấu hoặc vị trí bit giấu. Các số ngẫu nhiên có thể được sinh trực tiếp bằng các bộ sinh trong máy tính (C++, PHP và Matlab sử dụng bộ sinh Mersenne Twister) hoặc do người lập trình tự cài đặt.

**2.2.9: Bộ sinh số đồng dư (congruential generator)**

Công thức cho bộ sinh số đồng dư có dạng:

𝑥𝑘 = (𝑎 ∗ 𝑥𝑘−1 + 𝑏) 𝑚𝑜𝑑 𝑀 (2.2)

Trong đó a và b là hai số nguyên cho trước, trị x0 ban đầu được gọi là “hạt giống” (seed) và số nguyên M là số xấp xỉ (hoặc bằng) với số nguyên lớn nhất trên máy tính. Bộ sinh đồng dư cải tiến sử dụng công thức như sau :

𝑥𝑘 = (𝑎 ∗ 𝑥𝑘−1 + 2 ∗ 𝑎) 𝑚𝑜𝑑 𝑛 (2.3)

Trong đó a là một số nguyên lớn và thường chọn là một số nguyên tố, số mồi là x0.

**2.2.10: Bộ sinh Fibonacci**

Bộ sinh này sẽ tạo ra giá trị mới như là một hiệu, một tổng hoặc một tích của các giá trị trước. Một ví dụ tiêu biểu là bộ sinh hiệu:

𝑥𝑘 = (𝑥𝑘−17 − 𝑥𝑘−5) (2.4)

Chúng ta nói rằng, bộ sinh này có độ trễ 17 và 5. Bộ sinh hiệu này có thể tạo ra một kết quả âm. Một ưu điểm của bộ sinh Fibonacci so với bộ sinh đồng dư là bộ sinh Fibonacci có thời gian (lặp lại) dài.

**2.3 Đánh giá các phương pháp giấu tin trong âm thanh**

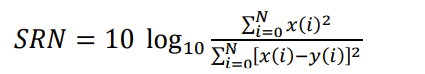
Để đánh giá hiệu năng của các kỹ thuật giấu tin trong âm thanh, chúng ta thường dựa vào tỉ lệ có thể cảm nhận và tỉ lệ có thể phát hiện.

**2.3.1. Đánh giá bằng các độ đo**

**2.3.1.1. Độ đo SNR**

Để đánh giá khả năng cảm nhận ta dựa vào tham số là segSNR (segmental signal – to – noise ratio). Giá trị của segSNR chỉ số lượng thay đổi trên dữ liệu chứa do chèn âm dữ liệu mật vào và được tính theo thang độ decibel. Trong dữ liệu âm thanh, giá trị SNR dưới 20 dB thường được xem là nhiễu trong khi giá trị SNR từ 20 dB hàm ý rằng chất lượng âm thanh được đảm bảo.

Độ đo SNR được tính như sau:

(2.5)

Trong công thức trên, x (n) thể hiện âm thanh gốc và y(n) thể hiện âm thanh chứa tin giấu.

**2.3.1.2. Độ đo NCC (Normalized Cross Correlation)**

NCC dùng để đo độ tương quan giữa hai chuỗi số, có thể dùng để đo độ tương quan giữa âm thanh gốc và âm thanh có chứa tin, hoặc chuỗi bit được giấu và chuỗi bit nhận được. Công thức để tính NCC như sau:



trong đó 𝑥(𝑖) là chuỗi số gốc và y(𝑖) là chuỗi nhận được có các thay đổi trên chuỗi dữ liệu gốc. Giá trị của NCC nằm trong khoảng từ 0 đến 1. Nếu giá trị này càng gần 1 chứng tỏ hai chuỗi càng giống nhau.

**2.3.2: Đánh giá bằng các phần mềm phát hiện tin**

Ba phần mềm tiêu biểu phát hiện tin trong âm thanh là HITIT, Stego-Suite và Stegsecret. HITIT được phát triển bởi Yavanoglu U và các cộng sự, thực hiện phát hiện tin giấu trong tệp wav. Stego-Suite là sản phẩm thương mại của công ty Allien và Stegsecret là sản phẩm mã nguồn mở được phát triển bởi Alfonso Muñoz. Nếu như kỹ thuật giấu của chúng ta không vượt qua được một phần mềm phát hiện tin giấu thì không thể nói kỹ thuật giấu là an toàn.

**2.3.3: Đánh giá bằng bảng đánh giá ODG(Object Difference Grade)**

Bảng đánh giá ODG được đề xuất bởi hiệp hội viễn thông quốc tế ITU (International Telecommunication Union). Đây là kỹ thuật chủ quan, dựa vào cảm nhận của con người. Bảng ODG gồm có 5 tiêu chí tương ứng với 5 mức điểm khác nhau. Chi tiết các tiêu chí trong bảng 2.2.

*Bảng 2.2. Các tiêu chí đánh giá theo thang ODG*

|  |  |
| --- | --- |
| **Mô tả sự thay đổi trên dữ liệu gốc** | **ODG** |
| Không cảm nhận được | 0.0 |
| Cảm nhận được, nhưng không gây khó chịu | −1.0 |
| Hơi gây khó chịu | −2.0 |
| Gây khó chịu | −3.0 |
| Rất khó chịu | −4.0 |

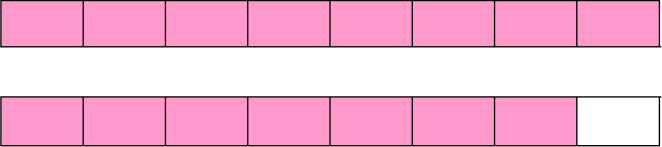
**2.4 phương pháp giấu tin trong âm thanh**

Các phương pháp giấu tin trong âm thanh có thể thực hiện trên miền thời gian hoặc miền tần số, hoặc kết hợp trên cả hai miền. Một số phương pháp có thể cần sử dụng tín hiệu gốc trong quá trình rút trích thông tin. Mục này sẽ trình bày một số phương pháp giấu tin cơ bản và các nghiên cứu liên quan đến phương pháp giấu đó.

### **2.4.1** **Phương pháp điều chỉnh bit ít quan trọng nhất (LSB coding)**

Đây là một trong những kỹ thuật được nghiên cứu và ứng dụng sớm nhất trong lĩnh vực giấu tin trên âm thanh cũng như trên các định dạng dữ liệu khác. Ý tưởng chính của kỹ thuật này là thay thế các bit ít quan trọng của mẫu dữ liệu gốc bằng các bit của tin giấu. Các bit được gọi là bit ít quan trọng khi ta thay đổi giá trị của bit đó từ 0 sang 1 hay từ 1 sang 0 thì sự thay đổi giá trị của mẫu dữ liệu không lớn và nó không gây ra sự khác biệt nào đối với hệ thống tri giác của con người. Ví dụ ta có mẫu 8 bit có giá trị bằng 94 và cần giấu bit 1 vào trong mẫu này. Khi đó ta sẽ điều chỉnh giá trị bit cuối cùng mẫu từ 0 thành 1 và giá trị mới của mẫu là 95. Hình 2.4 minh hoạ cách điều chỉnh bit trên mẫu có giá trị 94 để giấu bit 1. Trong trường hợp cần giấu bit 0 vào mẫu này thì ta không cần điều chỉnh giá trị của mẫu.

Giá trị mẫu dữ liệu gốc



0

1

0

1

1

1

1 0

Giá trị mẫu sau khi điều chỉnh để giấu bit 1:

0 1 0 1 1 1 1 **1**

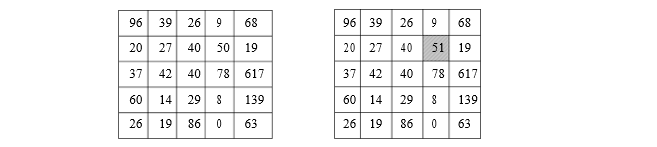
*Hình 2.4. Điều chỉnh bit thấp nhất của mẫu để giấu bit 1*

Khi giấu 1 bit mật vào mẫu dữ liệu thì độ lệch giá trị giữa mẫu dữ liệu gốc và dữ liệu mang tin tối đa là 1.

Để tăng độ bền vững của kỹ thuật giấu này, ta có thể giấu nhiều lần chuỗi thông điệp mật hoặc điều chỉnh vị trí các bit cao hơn trong mỗi mẫu để giấu tin. Các kỹ thuật giấu dùng phương pháp điều chỉnh LSB thường chọn các bit từ vị trí 1 đến 3, tính từ bên phải sang trái. Ưu điểm của phương pháp LSB là tỉ lệ dữ liệu lớn, bình quân 8 mẫu dữ liệu có thể giấu một byte dữ liệu mật, vì vậy tỉ lệ dữ liệu có thể bằng số mẫu dữ liệu trong tệp. Do không phải thực hiện nhiều phép toán phức tạp nên phương pháp này có thời gian thực hiện rất nhanh. Đối với phương pháp điều chỉnh LSB, ta có thể tăng thêm tỉ lệ giấu bằng cách dùng hai bit LSB, nghĩa là trong mỗi mẫu ta sẽ giấu 2 bit mật. Tuy nhiên cách này cũng làm tăng nhiễu trên đối tượng chứa dẫn đến đối phương dễ phát hiện và thực hiện các tấn công.

### **Phương pháp chẵn lẻ (parity coding)**

Phương pháp chẵn lẻ cũng tương tự như phương pháp mã hoá LSB, nhưng thay vì dùng một mẫu dữ liệu sẽ dùng một nhóm các mẫu. Dãy tín hiệu chứa sẽ được chia thành các đoạn (khối) có cùng kích thước. Tính chẵn lẻ của đoạn được xác định bằng tổng giá trị các mẫu trong đoạn đó chẵn hay lẻ. Nếu như đoạn có tính chẵn nhưng cần giấu bit 1 thì sẽ điều chỉnh (lật) một bit của một mẫu nào đó trong đoạn để đoạn có tính lẻ. Nếu như cần giấu bit 0 vào đoạn có tính chẵn thì không cần phải làm gì. Tương tự như vậy cho trường hợp giấu bit 1.



*Hình 2.5. Điều chỉnh mẫu để giấu bit 1 trong khối theo phương pháp chẵn lẻ*

Hình 2.5 minh họa cho việc giấu bit 1 vào khối. Do tổng giá trị của khối là số chẵn nên ta sẽ điều chỉnh giá trị 1 mẫu để đảm bảo tổng của khối có giá trị lẻ.

Trong quá trình giải tin ta sẽ dựa vào tính chẵn lẻ của đoạn để rút trích các bit.

Nếu như đoạn có tính lẻ thì rút trích bit 1, ngược lại rút trích bit 0.

Tính chẵn lẻ của đoạn cũng có thể được xác định theo một tiêu chí khác nào đó.

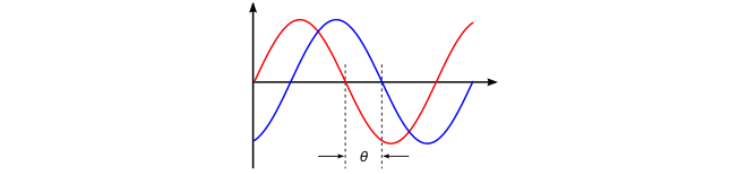
Ví dụ có thể xác định tính chẵn lẻ theo tổng số bit 1 của các mẫu trong đoạn.

So với phương pháp mã hoá LSB, phương pháp mã hoá chẵn lẻ có tỉ lệ dữ liệu thấp hơn, do phải dùng nhiều mẫu hơn để giấu 1 bit. Có thể nâng độ an toàn của phương pháp này bằng cách chọn các đoạn có kích thước không cố định và điều chỉnh hàm xác định tính chẵn lẻ của đoạn.

Kỹ thuật chẵn lẻ được dùng nhiều trong giấu tin trong ảnh

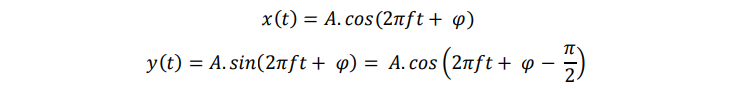
**2.4.3. Phương pháp mã hoá pha (phase coding)**

Mã hóa pha là một phương pháp khác dùng để giấu tin trong âm thanh. Phương pháp mã hoá pha dựa vào đặc tính tai người không phân biệt được sự khác nhau về pha của hai tín hiệu âm thanh. Việc giấu tin được thực hiện thông qua việc thay thế pha của một đoạn (segment) âm thanh ban đầu bằng một pha tham chiếu (referency phase) thể hiện dữ liệu. Pha của các đoạn tiếp theo sẽ được điều chỉnh sao cho duy trì mối quan hệ giữa các đoạn.



*Hình 2.6. Sự dịch chuyển pha của tín hiệu*

Giả sử ta có hai chuỗi x(t) và y(t) như sau:



Trong đó A là biên độ, f là tần số và 𝜑 là pha. Thuật ngữ pha được hiểu theo nghĩa đó là tham chiếu đến một tín hiệu nào khác. Ví dụ nếu tham chiếu đến tín hiệu 𝐴. cos(2𝜋𝑓𝑡) thì tín hiệu x(t) có pha là 𝜑 và tín hiệu y(t) có pha là 𝜑 – 𝜋/ 2 .

Thuật toán giấu tin sử dụng phương pháp mã hoá pha như sau :

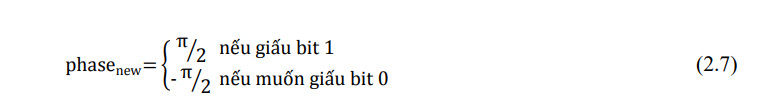
Bước 1) Chuỗi âm thanh gốc được chia thành dãy N đoạn (segment) ngắn và biến đổi Fourier cho từng đoạn.

Bước 2) Tính sự chênh lệch về pha giữa các đoạn kề nhau.

Bước 3) Điều chỉnh pha để giấu tin:

- Đối với pha đầu tiên, tạo pha P0 tùy ý.

- Đối với tất cả các đoạn khác, tạo ra các pha mới để giấu tin theo công thức (2.7).



- Kết hợp pha mới này với biên độ gốc để tạo ra đoạn mới, Sn.

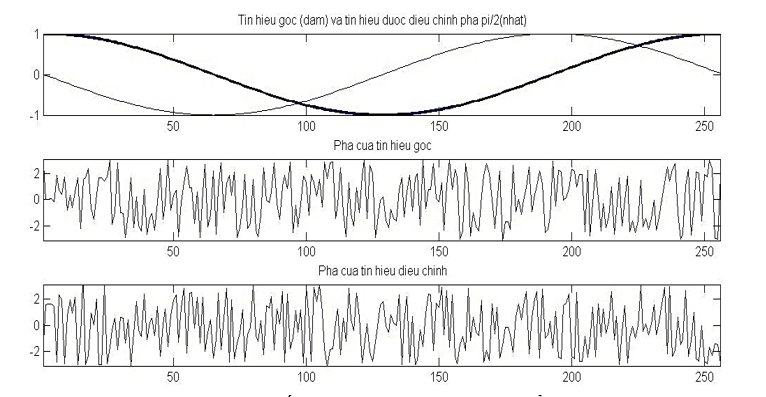
Bước 4) Thực hiện biến đổi Fourier ngược các đoạn và kết nối các đoạn lại để tạo ra tín hiệu âm thanh mới

Để giải tin được giấu theo phương pháp này cần phải biết chiều dài của các đoạn, thực hiện theo chiều ngược lại để lấy lần lượt các bit giấu.

Ví dụ có tín hiệu phức X gồm 256 (t=256) mẫu, được sinh theo công thức: X=complex (cos (2\*pi\*2\*t), sin (2\*pi\*2\*t))

Y là kết quả khi dịch pha X một khoảng 𝜋⁄2.

Minh hoạ cho tín hiệu X, Y và pha của chúng trong hình 2.7.



*Hình 2.7. Tín hiệu gốc và tín hiệu sau khi dịch chuyển pha* 𝜋/2

Mã hoá pha có ưu điểm là không gây nhiễu cho tín hiệu gốc. Lý do là phương pháp mã hoá pha chỉ thay đổi pha chứ không thay đổi biên độ và tần số của tín hiệu gốc. Hạn chế của phương pháp mã hoá pha là có tỉ lệ dữ liệu thấp và thời gian xử lý để giấu tin lâu.

Tỉ lệ dữ liệu giấu của phương pháp này thấp vì dữ liệu chỉ được giấu trong đoạn đầu nên tối đa chỉ bằng kích thước đoạn đầu (cũng bằng kích thước mỗi đoạn).

Ta không thể tăng kích thước của mỗi đoạn lên quá lớn vì nếu như vậy sẽ gây ra hiện tượng không liên tục về pha trong tín hiệu (phase discontinuous) và người thám tin dựa vào đặc trưng này phát hiện tệp có chứa tin giấu. Thông thường ta dùng các kỹ thuật làm phẳng pha để kiểm tra sau khi giấu tin xong. Hình 2.8 minh họa sự không liên tục về pha trong tín hiệu.

A picture containing histogram

Description automatically generated

**2.4.4. Phương pháp mã hoá tiếng vọng (echo coding)**

Phương pháp mã hoá tiếng vọng nhúng tin vào âm thanh bằng cách điều chỉnh độ trễ thêm tiếng vọng vào tín hiệu gốc. Phương pháp này dựa vào đặc trưng của hệ thống thính giác con người là không phân biệt được hai âm thanh nếu chúng xảy ra gần như đồng thời, có độ lệch trong khoảng từ 1 đến 40 mili giây. Gọi S(n) là tín hiệu gốc và X(n) là tín hiệu được thêm tiếng vọng. Ta có công thức giấu như sau:

𝑋(𝑛) = 𝑆(𝑛) + 𝛼. 𝑆(𝑛 − 𝑑) (2.8)

Diagram

Description automatically generated

*Hình 2.9. Các thành phần trong tiếng vọng của tín hiệu*

Dữ liệu nhúng được giấu bằng cách thay đổi 3 tham số của tiếng vọng là biên độ ban đầu, tỉ lệ phân rã và độ trễ. Các tham số này thể hiện như trong hình 2.9.

Nếu độ lệch thời gian giữa tín hiệu gốc và tiếng vọng thấp thì hai tín hiệu có thể trộn lẫn mà hệ thống thính giác của con người rất khó nhận biết được. Giá trị độ lệch này không thể xác định chính xác vì tuỳ thuộc vào chất lượng âm thanh và kiểu âm thanh. Giá trị được chọn trong các phương pháp giấu tin sử dụng kỹ thuật điều chỉnh tiếng vọng là dưới 20 mili giây.

Bằng cách dùng thời gian trễ khác nhau giữa tín hiệu gốc và tiếng vọng để thể hiện tương ứng giá trị nhị phân 1 hoặc 0, theo cách đó dữ liệu được giấu vào tệp âm thanh. Hai độ trễ có thể chọn cố định hoặc tuỳ biến tuỳ thuộc vào khoá.

Khi đó công thức (2.8) sẽ được điều chỉnh để giấu bit 1 và bit 0 trên một đoạn âm thanh sẽ là:

𝑋(𝑛) = 𝑆(𝑛) + 𝛼. 𝑆(𝑛 − 𝑑1)

𝑋(𝑛) = 𝑆(𝑛) + 𝛼. 𝑆(𝑛 − 𝑑0) (2.9)

Trong một số bài toán, ta chỉ cần thêm một tiếng vọng vào tín hiệu gốc như công thức (2.9) để giấu tin. Tuy nhiên, trong các phương pháp điều chỉnh tiếng vọng cải tiến, ta có thể thêm nhiều tiếng vọng, có thể là vọng trước và vọng sau so với tín hiệu gốc để giấu tin. Ví dụ trong đề xuất phương pháp thêm tiếng vọng cả trước và sau so với tín hiệu gốc như công thức (2.10).

𝑋(𝑛) = 𝑆(𝑛) + 𝛼. 𝑆(𝑛 + 𝑑) + 𝛼. 𝑆(𝑛 − 𝑑) (2.10) Bằng cách phân tích cepstrum của 𝑋(𝑛) ta sẽ biết chính xác vị trí thêm tiếng vọng . Để rút trích tin giấu, ta sẽ phân tích cepstrum của từng đoạn, dò tìm ra độ lệch 𝑑0 , 𝑑1 từ đó sẽ rút trích bit 0 hoặc bit 1. Chi tiết về cepstrum của tín hiệu được

trình bày ở mục 3.5.

Việc phân tích cepstrum để dò ra các giá trị 𝑑0, 𝑑1 trong quá trình giải tin tốn nhiều thời gian, và đây cũng chính là khuyết điểm của phương pháp mã hoá tiếng vọng. Ngoài ra, phương pháp giấu tin này chỉ cần biết kích thước mỗi đoạn giấu mà không cần dùng tín hiệu gốc trong quá trình giải tin nên người thám tin cũng có thể tự phân tích cepstrum để lấy tin giấu.

### **Phương pháp trải phổ**

Phương pháp trải rộng phổ biến là một trong những kỹ thuật được sử dụng trong kỹ thuật giấu tin trong tệp âm thanh. Phương pháp này tận dụng tính chất của phép biến đổi Fourier để ẩn thông tin vào tần số phổ biến của tín hiệu âm thanh gốc

Có nhiều phương pháp khác nhau dùng để trải dữ liệu nhúng vào phổ tần số :

Kỹ thuật DSSS (Direct Sequency Spread Spectrum) trải tín hiệu bằng cách nhân tín hiệu với một dãy giả ngẫu nhiên nào đó được gọi là chip. Tuy nhiên, nhược điểm của phương pháp này có thể thêm vào những nhiễu ngẫu nhiên mà người nghe có thể phát hiện ra .

Đối với phương pháp FHSS (Frenquency Hopped Spread Spectrum), tín hiệu âm thanh gốc được chia thành các mảnh nhỏ hơn và mỗi mảnh mang một tần số duy nhất. Lợi điểm cơ bản của kỹ thuật trải phổ là có thể chống lại các thay đổi. Bởi vì dữ liệu nhúng được trải đều trên dữ liệu chứa nên rất khó có thể thay đổi thông tin nhúng mà không làm thay đổi dữ liệu chứa. Đây là một đặc điểm đáng chú ý để thực hiện thủy vân trong âm thanh.

#### **Thuật toán giấu tin theo phương pháp trải phổ**

Đầu vào: - Tín hiệu gốc s(n);

- Chuỗi bit cần giấu b(n);

- Khoá k dùng để sinh chuỗi ngẫu nhiên.

Đầu ra: - Tín hiệu x(n) chứa tin mật.

Bước 1) Chuyển chuỗi bit nhị phân cần giấu 𝑣(𝑛) sang dạng lưỡng cực 𝑏(𝑛)

tương ứng, trong đó 𝑏(𝑖) = 1 𝑛ế𝑢 𝑣(𝑖) = 1 𝑣à 𝑏(𝑖) = −1 𝑛ế𝑢 𝑏(𝑖) = 0.

Bước 2) Sinh chuỗi ngẫu nhiên 𝑟(𝑛) dựa vào khoá.

Bước 3) Xây dựng chuỗi tin giấu 𝑤(𝑛)=𝑏. 𝑟(𝑛).

Bước 4) Trải chuỗi tin giấu vào chuỗi tín hiệu gốc dùng công thức

𝑥(𝑛) = 𝑠(𝑛) + 𝛼. 𝑤(𝑛) (2.11)

Trong công thức (2.11) trên, 𝛼 là giá trị dùng để điều chỉnh tính không cảm nhận được của dữ liệu trước và sau khi giấu và/hoặc tính bền vững của tin giấu.

#### **Thuật toán giải tin theo phương pháp trải phổ**

Quá trình giải tin dựa vào sự tương quan tuyến tính (linear correlation) giữa hai chuỗi x(n) và r(n), trong đó r(n) là chuỗi ngẫu nhiên được sinh ra từ khoá mật. Trong quá trình giải tin cũng cần sử dụng các phương pháp xử lý tín hiệu số như lọc thông cao hay các phương phương pháp lọc khác.

Diagram

Description automatically generated

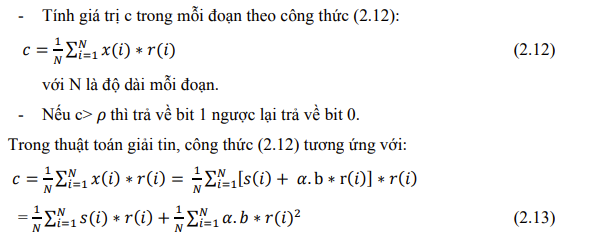
*Hình 2.10. Xử lý (lọc) mỗi khối để giải tin trong kỹ thuật trải phổ*

Đầu vào: -Tín hiệu chứa tin mật x(n);

* Khoá k dùng để sinh chuỗi ngẫu nhiên;
* Ngưỡng 𝜌 để phân biệt bit 0 hoặc bit 1. Đầu ra : - Chuỗi bit b(n), là chuỗi thông tin mật.

Bước 1) Sinh chuỗi ngẫu nhiên 𝑟(𝑛) dựa vào khoá.

Bước 2) Lấy từng bit trong mỗi đoạn



Để lấy được chuỗi tin đã giấu thì ta cần điều chỉnh sao cho phần thứ nhất của công thức (2.13) nhỏ hơn so với phần thứ 2. Nếu không thoả mãn yêu cầu này thì tin nhận được khi giải sẽ bị sai. Giải pháp để đảm bảo yêu cầu phần thứ 2 trong công thức (2.13) lớn hơn là sử dụng phương pháp tiền xử lý trên tín hiệu x(n), chẳng hạn lọc để loại bỏ s(n) ra khỏi x(n).

### **Phương pháp điều chỉnh biên độ**

Trong phương pháp này, thông tin được nhúng bằng cách thay đổi năng lượng của 2 hay 3 khối (block). Năng lượng của mỗi block kích thước N được xác định như công thức (2.14):

𝑁

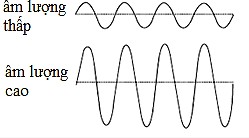
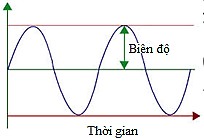
𝐸 = ∑

𝑖=1

𝑎𝑏𝑠(𝑆(𝑖))

(2.14)

Nếu các tín hiệu trong khối có biên độ lớn thì năng lượng của khối sẽ cao.



*Hình 2.11. Biên độ và năng lượng của tín hiệu*

Giả sử có hai block liền nhau được dùng để nhúng thông tin. Ta có thể làm cho năng lượng của hai block A và B giống hay khác nhau bằng cách chỉnh sửa biên độ của từng block. Gọi EA và EB lần lượt là năng lượng của block A và block B.

* Nếu E A ≥ EB – ℓ, ta xác định bit giá trị của thông điệp mật m = 0.
* Nếu EA < EB -ℓ, ta xác định bit giá trị thông điệp mật m = 1.

Phương pháp này có hạn chế khi độ lệch hai khối lớn. Giả sử block A có năng lượng cao hơn nhiều so với block B. Khi bit thông điệp mật m nhúng vào là 0 sẽ không có vấn đề gì. Ngược lại, nếu giấu bit 1 ta phải điều chỉnh sao cho năng lượng EB lớn hơn EA. Vì khoảng năng lượng giữa hai block chênh lệch nhau rất lớn, do đó, sau khi chỉnh sửa, kết quả đạt được không còn tự nhiên như lúc đầu mà rất dễ bị phát hiện. Vấn đề khó khăn này có thể được giải quyết bằng cách sử dụng đến ba block thay vì sử dụng hai block, hoặc có thể sử dụng nhiều hơn.

* + 1. **Phương pháp lượng tử hoá (quantization)**

Phương pháp lượng tử hoá sẽ thay thế giá trị một mẫu x thành y để giấu tin.

Công thức điều chỉnh x thành y như sau:

A picture containing graphical user interface

Description automatically generated

Trong công thức (2.15), 𝑞(. ) là hàm lượng tử và 𝐷 là bước lượng tử. Hàm lượng tử 𝑞(𝑥, 𝐷) như công thức (2.16):

𝑞(𝑥, 𝐷) = [𝑥/𝐷] ∗ 𝐷 (2.16)

Trong đó [𝑥] là hàm lấy số nguyên gần x nhất. Hình 2.12 minh hoạ cho phương pháp lượng tử.

A picture containing text, antenna

Description automatically generatedGiả sử một mẫu dữ liệu x được lượng tử đến vị trí nốt đen 𝑞(𝑥, 𝐷). Nếu giá trị bit cần giấu là 0 thì sẽ điều chỉnh để di chuyển đến vị trí nốt trắng, ngược lại sẽ được di chuyển đến nốt chéo.

*Hình 2.12. Sơ đồ điều chỉnh giá trị*

A picture containing text

Description automatically generatedVí dụ x=81, D = 8, khi đó 𝑞(𝑥, 𝐷) =𝑞(81, 8) = 80. Nếu cần giấu bit 1 thì y=82, ngược lại y sẽ bằng 78 nếu giấu bit 0.

Một cải tiến của phương pháp này được đưa ra bằng cách dùng thêm một vector phụ (vector dither). Công thức điều chỉnh như sau:

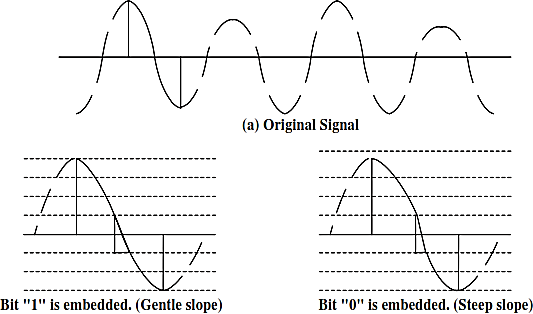
𝑦𝑚 = 𝑞(𝑥 + 𝑑𝑚, 𝐷) – 𝑑𝑚

Trong đó m là chỉ số và 𝑑𝑚 là phần tử thứ m của vector phụ. Ví dụ cho 𝑑1=2,

𝑑2=0, x=8 và D = 4 thì 𝑦1= 10 và 𝑦2= 8.

### **Phương pháp điều chế tỉ lệ thời gian**

Phương pháp điều chỉnh tỉ lệ thời gian được thực hiện bằng cách kéo dài hoặc rút ngắn tỉ lệ thời gian âm thanh. Ý tưởng cơ bản của phương pháp này là thay đổi tỉ lệ thời gian giữa hai cực (tức là hai giá trị cực đại và cực tiểu trong đoạn xét). Khoảng giữa hai cực được chia làm N phân đoạn bằng nhau. Ta thay đổi độ dốc của tín hiệu, tuỳ thuộc vào bit muốn nhúng. Ví dụ dốc thoải để giấu bit 1, dốc đứng để giấu bit 0.



Tín hiệu gốc

Giấu bit 1

Giấu bit 0

*Hình 2.13. Điều chỉnh độ dốc để giấu thông tin*

Trong trình bày thuật toán giấu dùng phương pháp điều chỉnh tỉ lệ thời gian nâng cao cải tiến khả năng chống tấn công chỉnh sửa tỉ lệ thời gian.

### **Phương pháp giấu dựa vào Patchwork**

Phương pháp Patchwork nhúng một lượng giá trị vào trong dữ liệu gốc, ban đầu được thiết kế cho giấu tin trong ảnh. Thuật toán giấu tin theo sơ đồ này gồm hai bước chính:

Bước 1) Chọn 2 tập ngẫu nhiên A, B

Bước 2) Thêm vào giá trị d cho mỗi phần tử thuộc tập A và trừ ra d đối với mỗi phần tử thuộc tập B, trong đó d là một hằng số có giá trị nhỏ.



Quá trình giải tin được thực hiện bắt đầu bằng cách trừ các giá trị các mẫu trên hai tập A, B. Tiếp theo giá trị 𝐸[𝑎̅∗ − 𝑏̅∗] được dùng để xác định các mẫu có tin giấu

Chart, histogram

Description automatically generatedhay không. Giá trị 𝑎̅∗ và 𝑏̅∗ là giá trị trung bình của A và B sau khi được giấu tin. Quá trình giải tin theo kỹ thuật Patchwork không cần dùng tin gốc.

*Hình 2.14. So sánh giá trị trung bình của tín hiệu không có và có giấu tin.*

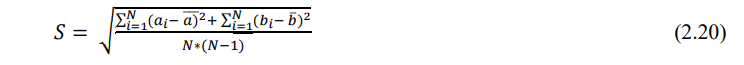
Kỹ thuật này có hạn chế là quá trình giải tin có thể nhận tin sai. Ta có:

Text

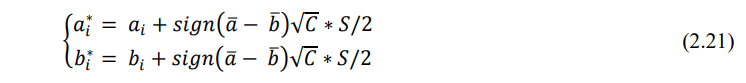
Description automatically generated

Kỹ thuật này hạn chế việc giải tin sai của thuật toán Patchwork. Các bước của quá trình giấu tin như sau:

Bước 1) Chọn hai tập ngẫu nhiên A, B sau đó thực hiện các phép tính 𝑎̅ và ̅𝑏 rồi tính giá trị S theo công thức (2.20) như sau:



Bước 2) Hàm nhúng được thực hiện theo công thức (2.21):



Trong công thức trên C là một hằng số, sign là hàm làm cho giá trị lớn hơn sẽ lớn hơn và giá trị nhỏ sẽ nhỏ hơn. Mục đích của việc này là làm cho giá trị trung bình khoảng cách giữa hai giá trị mẫu luôn lớn hơn 𝑑 = √𝐶 ∗ 𝑆.

Chart, histogram

Description automatically generated

*Hình 2.15. Giá trị trung bình của tín hiệu có và không có giấu tin theo MPA*

Theo cách điều chỉnh của thuật toán MPA, giá trị phân bố giá trị trung bình khoảng cách 2 mẫu sẽ như hình 2.15.

Thuật toán giải tin như sau:

Bước 1) Tính giá trị thống kê như công thức (2.20)



Bước 2) So sánh giá trị 𝑇2 với ngưỡng 𝛽. Nếu 𝑇2 > 𝛽 thì xác nhận có tin giấu, ngược lại thì không có.

**2.4.10 Phương pháp dựa vào các đặc trưng nổi bật**

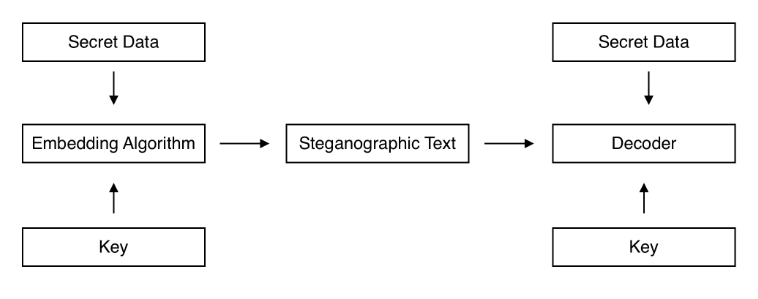
Phương pháp dựa vào các đặc trưng nổi bật trong việc Giấu tin trong tệp âm thanh là một phương pháp khác để Giấu tin vào tệp âm thanh. Thay vì trực tiếp Giấu tin vào dữ liệu âm thanh, phương pháp này dựa trên công việc tìm kiếm và sử dụng các đặc điểm nổi bật của tín hiệu âm thanh để Giấu tin. Một số đặc trưng nổi bật của tín hiệu âm thanh bao gồm biên độ, tần số, độ dốc và năng lượng của tín hiệu. Các đặc trưng này có thể được sử dụng để chặn tin bằng cách thay đổi giá trị của chúng theo cách ẩn danh. Ví dụ, các giá trị biên độ của các khối tín hiệu có thể được thay đổi nhẹ nhàng để đảm bảo tin mà không làm thay đổi quá nhiều thông tin âm thanh gốc. Phương pháp này có thể được áp dụng để giấu tin trong các tệp âm thanh có chất lượng cao, ví dụ như các tệp FLAC hay WAV, nơi các đặc trưng của tín hiệu âm thanh có thể được phân tích và sử dụng để giấu tin một cách hiệu quả và không được phát hiện. Tuy nhiên, phương pháp này không thể hiệu quả với các tệp âm thanh có chất lượng thấp hoặc bị nhiễu

**CHƯƠNG 3. THỰC NGHIỆM**

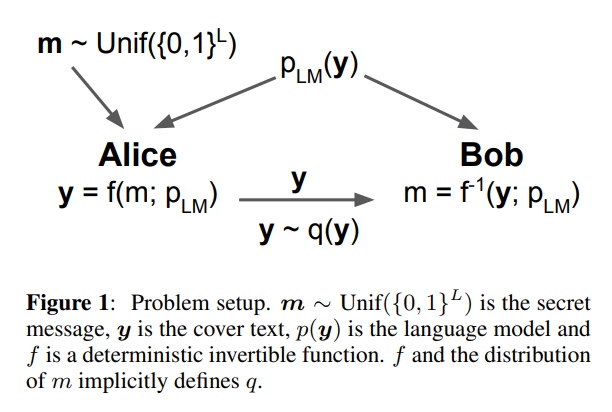
* 1. **Đặt Vấn Đề.**

Giả sử A muốn gửi cho B một tin nhắn chứa thông tin nhạy cảm qua một kênh được giám sát bởi C. Kênh này có thể được sử dụng chung để liên lạc bởi nhiều bên khác nhau. Thông tin được truyền đi trong kênh là văn bản chứa ngôn ngữ tự nhiên (natural language). A vừa phải đảm bảo chỉ có mình B hiểu được nội dung tin nhắn, vừa phải tránh gửi những nội dung "mất tự nhiên" có thể khiến C nghi ngờ. Vậy A và B có thể thực hiện các bước như sau:

1. A mã hóa nội dung tin nhắn thành một đoạn văn bản mã hóa (ciphertext), key để mã hóa được nắm giữ bởi cả A và B
2. A giấu đoạn ciphertext (thường là một chuỗi các bit) vào trong một tin nhắn dưới dạng ngôn ngữ tự nhiên bình thường (stegotext)
3. A gửi đoạn stegotext qua kênh liên lạc do C giám sát
4. B nhận được và trích xuất ra ciphertext
5. B giải mã đoạn ciphertext với key chung để lấy được thông tin mật.



**Mô hình tổng quát của bài toán giấu tin được thể hiện như sau:**



Như vậy ta có thể thấy quá trình này liên quan đến hai công đoạn chính: (1) mã hóa đoạn tin cần chuyển đi và (2) giấu nó vào trong một đoạn văn bản bình thường.

Vậy để hệ thống steganography hoạt động hiệu quả thì hai công đoạn này cần phải đạt những mục tiêu nào? Hay nói cách khác, có những metrics nào để đánh giá tính hiệu quả của của một phương pháp giấu tin?

* 1. **Mã Hóa**
     1. **Nguyên lý hoạt động**
* Thuật toán mã hóa dựa trên việc lấy ra các frames của tệp âm thanh:

frame\_bytes = bytearray(list(waveaudio.readframes(waveaudio.getnframes())))

* Mỗi frame có kích thước 1 byte tương đương 8bit nhị phân.
* Message được chuyển từ ký tự Ascii => Unicode => mã nhị phân.
* Mỗi ký tự message được chuyển sang dạng nhị phân gồm 8 bit.

string = string + int((len(frame\_bytes)-(len(string)\*8))/8) \*'#'

bits = list(map(int, ''.join([bin(ord(i)).lstrip('0b').rjust(8,'0') for i in string])))

* Mỗi byte của frame gốc sẽ được AND nhị phân với 254 dưới dạng nhị phân cho ra bit cuối trong frame là bit rỗng (bằng 0).
* Sau khi đưa bit cuối của frame về dạng rỗng, byte frame đó sẽ OR tiếp với 1 bit của message tạo thành frame mới chứa 1 bit cuối đã được chỉnh sửa.

      for i, bit in enumerate(bits):

        frame\_bytes[i] = (frame\_bytes[i] & 254) | bit

      frame\_modified = bytes(frame\_bytes)

* + 1. **Source Code**
       1. **Module**
          1. Wave: cung cấp khả năng tiếp cận và thao tác thuận lợi hơn khi sử dụng các tệp âm thanh .wav
          2. Argparse: Cung cấp khả năng thao tác sử dụng các lệnh command-line
       2. **Code**

import os

import wave

import argparse

parser = argparse.ArgumentParser()

// truyền tham số đầu vào: file, messeage, name file new

parser.add\_argument('-f', help='Select Audio File', dest='audiofile')

parser.add\_argument('-m', help='Enter your Secret Message', dest='secretmsg')

parser.add\_argument('-o', help='Your Output file path and name', dest='outputfile')

args = parser.parse\_args()

af = args.audiofile

string = args.secretmsg

output = args.outputfile

arged = False

// Truyền đủ tham số => arged = True

if af and string and output:

    arged = True

// Hàm clear

def cls():

  os.system("clear")

// Hàm hỗ trợ

def help():

  print("\033[92mHide Your Secret Message in Audio Wave File.\033[0m")

  print ('''usage: HiddenWave.py [-h] [-f AUDIOFILE] [-m SECRETMSG] [-o OUTPUTFILE]

optional arguments:

  -h, --help    show this help message and exit

  -f AUDIOFILE Select Audio File

  -m SECRETMSG Enter your message

  -o OUTPUTFILE Your output file path and name''')

// Hàm mã hóa

def em\_audio(af, string, output):

// Nếu không truyền đủ tham số => chạy hàm help()

    if not arged:

      help()

    else:

      print ("Please wait...")

      waveaudio = wave.open(af, mode='rb')

// Lấy ra các frame của tệp âm thanh khi chưa có message ẩn

      frame\_bytes = bytearray(list(waveaudio.readframes(waveaudio.getnframes())))

// Chuyển từng ký tự message sang mã nhị phân 8bit và chuyển nó thành 1 mảng bit

      string = string + int((len(frame\_bytes)-(len(string)\*8))/8) \*'#'

      bits = list(map(int, ''.join([bin(ord(i)).lstrip('0b').rjust(8,'0') for i in string])))

// lấy từng byte của frame gốc AND 254 🡺 byte rỗng 🡺 OR với từng bit của message

      for i, bit in enumerate(bits):

        frame\_bytes[i] = (frame\_bytes[i] & 254) | bit

// Tạo ra 1 frame mới có chứa message trong các byte của video

      frame\_modified = bytes(frame\_bytes)

// Tạo ra một video mới đã gán message ẩn

      with wave.open(output, 'wb') as fd:

        fd.setparams(waveaudio.getparams())

        fd.writefames(frame\_modified)

      waveaudio.close()

      print ("Done...")

cls()

try:

  em\_audio(af, string, output)

except:

  print ("Something went wrong!! try again")

  quit('')

* 1. **Giải Mã**

**3.3.1 Nguyên lý hoạt động**

* Do phần mã hóa ở trên thay đổi 1 bit cuối cùng trong các frame của file âm thanh nên ở phần giải mã, chỉ cần thực hiện phép AND với bit cuối cùng đó, ta thu được frame gốc.

extracted = [frame\_bytes[i] & 1 for i in range(len(frame\_bytes))]

* Sau khi thu được các frame gốc, tiếp tục lấy 8 frame gốc hợp nhất sẽ thu được chuỗi 8 bit tương ứng với 1 ký tự message

string = "".join(chr(int("".join(map(str,extracted[i:i+8])),2))

for i in range(0,len(extracted),8))

msg = string.split("#")[0]

### **3.3.2. Code**

import os

import wave

import argparse

parser = argparse.ArgumentParser()

// Truyền vào file đã có message ẩn

parser.add\_argument('-f', help='audiofile', dest='audiofile')

args = parser.parse\_args()

af = args.audiofile

arged = False

if af:

    arged = True

def cls():

  os.system("clear")

// Hàm hỗ trợ

def help():

  print("\033[92mExtract Your Secret Message from Audio Wave File.\033[0m")

  print ('''usage: ExWave.py [-h] [-f AUDIOFILE]

optional arguments:

  -h, --help    show this help message and exit

  -f AUDIOFILE  Select Audio File''')

// Hàm giải mã

def ex\_msg(af):

// Nếu không truyền file có message ẩn 🡺 chạy hàm help()

    if not arged:

      help()

    else:

        print ("Please wait...")

        waveaudio = wave.open(af, mode='rb')

// Lấy ra các frame của tệp âm thanh khi có message ẩn

        frame\_bytes = bytearray(list(waveaudio.readframes(waveaudio.getnframes())))

// Lấy từng bit byte trong frame có message ẩn AND 1 🡺 frame gốc

        extracted = [frame\_bytes[i] & 1 for i in range(len(frame\_bytes))]

// lấy 8 frame gốc ==> 8bit tương ứng với 1 ký tự message

        string = "".join(chr(int("".join(map(str,extracted[i:i+8])),2))

for i in range(0,len(extracted),8))

// Loại bỏ phần thêm sau messeage 🡺 lấy ra message ẩn

        msg = string.split("#")[0]

        print("Your Secret Message is: \033[1;91m"+msg+"\033[0m")

        waveaudio.close()

cls()

try:

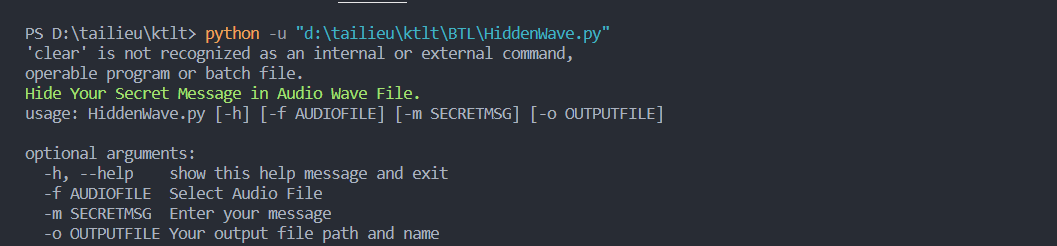
  ex\_msg(af)

except:

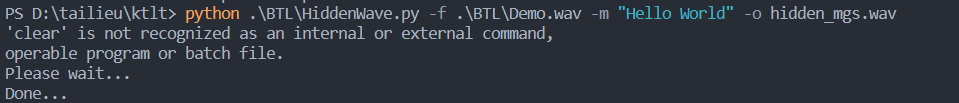
  print ("Something went wrong!! try again")

* 1. **Demo**

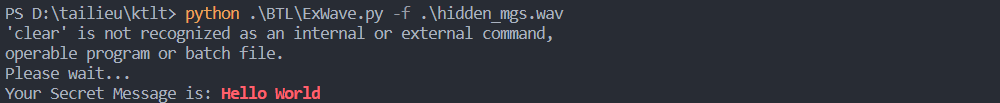
**Màn hình trợ giúp**



**Mã hóa**



**Giải mã**



**TỔNG KẾT**

Giấu thông tin và các ứng dụng của nó là một trong những vấn đề được nhiều người quan tâm hiện nay. Giấu thông tin được coi là một trong các giải pháp để bảo mật dữ liệu. Giấu tin trong âm thanh hiện nay vẫn còn nhiều hướng mở, nhất là các nghiên cứu ứng dụng trong các hệ thời gian thực. Các nghiên cứu để tăng tính bền vững, sự đồng bộ của tin giấu trong dữ liệu âm thanh đã có nhưng chưa nhiều. Trên cơ sở nghiên cứu các kỹ thuật giấu tin trong âm thanh, các kỹ thuật bổ trợ, luận án đã đề xuất một số cải tiến nhằm tăng cường tính mật, đưa ra các giải pháp khác nhau có thể lựa chọn cho bài toán giấu tin trong âm thanh. Cụ thể, bài tập đã đạt được những kết quả chính:

* Có cái nhìn tổng quan về kỹ thuật giấu tin trong hệ thống số.
* Tiếp cận và thực hành với ngôn ngữ lập trình python.
* Áp dụng kỹ thuật lập trình vào trường hợp thực tế, đạt được mục tiêu môn học.