ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LÊ ANH DŨNG

GIẤU TIN TRONG FILE ÂM THANH BẰNG CÁC PHÉP BIẾN ĐỔI RỜI RẠC

LUẬN VĂN THẠC SỸ: KHOA HỌC MÁY TÍNH

THÁI NGUYÊN, NĂM 2015

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LÊ ANH DŨNG

GIẤU TIN TRONG FILE ÂM THANH BẰNG CÁC PHÉP BIẾN ĐỔI RỜI RẠC

Chuyên ngành: Khoa học máy tính
Mã số: 60 48 0101

LUẬN VĂN THẠC SỸ: KHOA HỌC MÁY TÍNH

HƯỚNG DẪN KHOA HỌC: **TS TRỊNH THANH LÂM**

THÁI NGUYÊN, NĂM 2015

LỜI CAM ĐOAN

Tôi xin cam đoan luận văn "*Giấu tin trong file âm thanh bằng các phép biến đổi rời rạc*" là sản phẩm của riêng cá nhân, không sao chép lại của người khác. Trong toàn bộ nội dung của luận văn, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp, nghiên cứu từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ và trích dẫn rõ ràng.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Thái Nguyên, ngày 15 tháng 05 năm 2015

Học viên

Lê Anh Dũng

LỜI CẢM ƠN

Lời đầu tiên, tôi xin bày tỏ lòng biết ơn đến thầy TS Trịnh Thanh Lâm - ĐHQG Hà Nội, người đã tận tình hướng dẫn, chỉ bảo và giúp đỡ tôi trong suốt quá trình nghiên cứu và hoàn thành luận văn này.

Tôi xin chân thành cảm ơn các thầy cô giáo trường Đại học Công nghệ Thông tin và Truyền thông - Đại học Thái Nguyên đã giảng dạy và cung cấp cho chúng tôi những kiến thức rất bổ ích trong thời gian học cao học, giúp tôi có nền tảng tri thức để phục vụ nghiên cứu khoa học sau này.

Tôi cũng xin cảm ơn Lãnh đạo và đồng nghiệp tại đơn vị đã tạo điều kiện và giúp đỡ tôi trong suốt quá trình nghiên cứu và hoàn thành luận văn. Tôi cũng xin bày tỏ lòng cảm ơn đến gia đình và bạn bè, những người luôn quan tâm, động viên và khuyến khích tôi trong quá trình học tập.

Thái Nguyên, ngày 15 tháng 05 năm 2015

Lê Anh Dũng

MỤC LỤC

Trang

LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
MỤC LỤC iii
DANH MỤC CÁC HÌNH ẢNH vi
DANH MỤC CÁC BẢNG BIỂU vii
MỞ ĐẦU 1
CHƯƠNG 1. TỔNG QUAN VỀ GIẤU TIN VÀ ÂM THANH SỐ 5
1.1. Giới thiệu chung về giấu tin 5
1.1.1. Mã hóa và giấu tin 5
1.1.2. Phân loại kỹ thuật giấu tin 6
1.2. Các đối tượng của một bài toán giấu tin 8
1.2.1. Thông tin mật 8
1.2.2. Đối tượng chứa 8
1.2.3. Đối tượng đã nhúng 9
1.2.4. Khoá mật 9
1.3. Mô hình kỹ thuật giấu tin 9
1.4. Các tiêu chí đánh giá bài toán giấu tin 10
1.4.1. Khả năng không bị phát hiện 10
1.4.2. Tính bền vững 11
1.4.3. Khả năng lưu trữ 11
1.4.4. Tính vô hình 12
1.4.5. Độ phức tạp của thuật toán 12
1.5. Một số ứng dụng cụ thể 12
1.6. Các tấn công trên các hệ giấu tin 15
1.7. Âm thanh số 16
1.7.1. Khái niệm về âm thanh và âm thanh số 17
1.7.2. Một số định dạng file âm thanh trên máy tính 18
1.7.3. Cấu trúc file âm thanh dạng WAV 21

1.8. Một số kỹ thuật giấu tin trong file âm thanh 23
CHƯƠNG 2. KỸ THUẬT GIẤU TIN BẰNG CÁC PHÉP BIẾN ĐỔI RỜI RẠC 26
2.1. Các phép biến đổi từ miền không gian sang miền tần số 26
2.1.1. Phép biến đổi Fourier 26
2.1.2. Phép biến đổi Cosin rời rạc 27
2.1.3. Phép biến đổi Wavelet 29
2.2. Một số kỹ thuật giấu tin dựa trên biến đổi khối bít nhị phân 30
2.2.1. Mã hóa LSB (Least Significant Bit) 31
2.2.2. Mã hóa Parity (Parity Coding) 32
2.3. Thuật toán giấu tin bằng các phép biến đổi rời rạc trên số nguyên 34
2.3.1. Một số phép biến đổi rời rạc trên số nguyên 34
2.3.2. Thuật toán Wu-Lee 35
2.3.3. Thuật toán Wu-Lee cải tiến 38
2.3.4. Thuật toán giấu một chuỗi bít trong một khối tin 40
CHƯƠNG 3. TRIỂN KHAI CHƯƠNG TRÌNH THỬ NGHIỆM 48
3.1. Mục đích, yêu cầu 48
3.2. Yêu cầu về cấu hình hệ thống 48
3.3. Lựa chọn định dạng file âm thanh trong thực nghiệm 48
3.4. Sơ đồ chương trình 49
3.5. Thuật toán giấu tin và trích rút tin theo kỹ thuật đề xuất 50
3.5.1. Giấu tin 50
3.5.2. Trích rút tin mật 52
3.5.3. Một số hàm và thủ tục giấu tin 53
3.6. Kết quả thực nghiệm 54
3.7. Đánh giá kết quả thực nghiệm 64
3.8. Các khả năng ứng dụng 64
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 67
TÀI LIỆU THAM KHẢO 69

DANH MỤC CÁC TỪ VIẾT TẮT TRONG LUẬN VĂN

AAC - Định dạng âm thanh chuẩn (*Advanced Audio Coding*)

A/D D/A - Biến đổi tương tự/số và ngược lại (*Analog/Digital*)

AIFF - Định dạng không mất thông tin (*Audio Interchange File Format*)

DCT - Phép biến đổi Cosin rời rạc (*Discrete Cosine Tranform*).

DES - Hệ mật mã chuẩn (*Data Encryption Standard*)

DSP - Xử lý tín hiệu kỹ thuật số (*Digital signal processing*)

FLAC - Nén âm thanh không mất dữ liệu (*Free Lossless Audio Codec*),

FT - Biến đổi Fourier (*Fourier Transform*)

HAS - Hệ thống thính giác (*Human Auditory system*)

HVS - Hệ thống thị giác (*Human Vision System*)

IDE - Môi trường phát triển tích hợp (*Integrated Development Environment*)

IFT - Biến đổi Fourier ngược (*Inverse Fourier Transform*)

LSB - Bít ít quan trọng nhất (*Least Significant Bít*)

MP3 - Định dạng nén âm thanh (*Movie Picture Experts Group-Layer 3*)

PCM - Điều biến mã xung (*Pulse Code Modulation*)

RSA - Mã hóa công khai RSA (*Rivest, Shamir và Adleman*)

WAV - Định dạng âm thanh dạng sóng (*Waveform Audio Format*)

WMA - Định dạng âm thanh của Microsoft (*Windows Media Audio*)

DANH MỤC CÁC HÌNH ẢNH

Trang

Hình 1.1. Mô hình mã hóa thông tin 5
Hình 1.2. Một cách phân loại các kỹ thuật giấu tin 7
Hình 1.3. Lược đồ chung cho quá trình giấu tin 9
Hình 1.4. Lược đồ chung cho quá trình trích rút thông tin 10
Hình 1.5. Mối quan hệ giữa các yếu tố trong bài toán giấu tin..... 12
Hình 1.6. Ảnh gốc Lena và logo của viện khoa học Ấn Độ 13
Hình 1.7. Ảnh Lena đã được nhúng thủy vân hiển 14
Hình 1.8. Thông tin bị xuyên tạc 14
Hình 1.9. Tín hiệu âm thanh..... 17
Hình 1.10. Số hóa tín hiệu âm thanh..... 18
Hình 1.11. Mô tả định dạng của file.wav..... 21
Hình 1.12. Mô tả 72 byte của một file âm thanh WAV..... 23
Hình 2.1. Minh họa kỹ thuật LSB..... 31
Hình 2.2. Minh họa kỹ thuật mã hóa Parity..... 33
Hình 3.1. Sơ đồ chương trình thử nghiệm 49
Hình 3.2. Phổ biên độ và phổ pha của file chưa trước khi giấu tin 57
Hình 3.3. Phổ biên độ và phổ pha của file sau khi giấu tin 57
Hình.3.4. Trích đoạn các byte của file Sony.wav sau khi nhúng tin mật 63

DANH MỤC CÁC BẢNG BIỂU

Trang

Bảng 1.1. So sánh giấu thông tin mật và giấu thông tin thủy vân 8
Bảng 1.2. Một số định dạng file âm thanh trên máy tính 21
Bảng 1.3. Phần định dạng kiểu RIFF 22
Bảng 1.4. Phần định dạng thông tin âm thanh 22
Bảng 1.5. Phần dữ liệu âm thanh 23
Bảng 3.1. Một số phần mềm giấu tin 49

MỞ ĐẦU

1. Đặt vấn đề

Ngày nay, Internet là môi trường phổ biến cho việc trao đổi thông tin giữa các nhà cung cấp và người sử dụng. Do đó, vấn đề an toàn dữ liệu trên mạng luôn luôn là một thách thức đối với các nhà quản lý và các nhà nghiên cứu. Các thông tin trên Internet có thể dễ dàng bị làm giả mạo, sai lệch và bị đánh cắp bởi hacker trong quá trình truyền tải dữ liệu. Thông tin của cá nhân, tổ chức hoặc quốc gia đứng trước nguy cơ bị xâm nhập bất cứ lúc nào. Cùng với nó là vấn nạn ăn cắp bản quyền, xuyên tạc thông tin,... ngày càng gia tăng. Vì vậy, vấn đề đặt ra làm thế nào để đảm bảo được sự an toàn, và toàn vẹn thông tin trong quá trình truyền tải trên Internet. Hai giải pháp cho vấn đề này là mã hóa và giấu thông tin có vai trò quan trọng trong việc bảo vệ quá trình truyền tải thông tin mật. Sự xác thực và bản quyền trong môi trường trao đổi công cộng. Việc tìm giải pháp cho những vấn đề này giúp ta hiểu thêm về một công nghệ đang phát triển và còn tạo ra những cơ hội mới [1].

Trong những giải pháp đã và đang được triển khai thì giấu tin (Data Hiding) là một trong những giải pháp được các nhà nghiên cứu và phát triển coi đó là một hướng đi có nhiều triển vọng. Giấu thông tin là kỹ thuật nhúng một lượng thông tin số nào đó vào trong một đối tượng thông tin số khác mà các đối tượng đó thường là một tài liệu, hình ảnh, âm thanh hoặc video. Các kỹ thuật giấu tin có thể chia ra làm hai nhóm. Nhóm thứ nhất là các phương pháp che giấu thông tin trực tiếp. Nhóm này thường sử dụng các bít ít quan trọng nhất của một khối bít nhị phân được sửa đổi để giấu thông tin. Nhóm thứ hai lại che giấu thông tin thông qua các phép biến đổi chẳng hạn như phép biến đổi Cosin hay wavelet rời rạc được sử dụng rộng rãi [4].

Sau khi tiến hành nghiên cứu các tài liệu liên quan đến lĩnh vực giấu tin trong đa phương tiện và nhận thấy các kỹ thuật trên đều cho kết quả tốt với

việc đảm bảo được tính chất ẩn của thông tin được giấu và không làm ảnh hưởng đến chất lượng của dữ liệu gốc. Với mong muốn phát triển các kỹ thuật giấu thông tin nhằm bảo vệ các thông tin mật trong quá trình trao đổi. Được sự sự đồng ý, động viên của cán bộ hướng dẫn khoa học, tôi đã chọn đề tài "***Giấu tin trong file âm thanh bằng các phép biến đổi rời rạc***" làm vấn đề nghiên cứu cho luận văn cao học của mình. Mong rằng kết quả của đề tài khi được triển khai thực tế sẽ góp phần tăng thêm độ an toàn cho các thông tin mật trong quá trình trao đổi.

2. Mục tiêu nghiên cứu

Luận văn nghiên cứu hệ thống lý thuyết liên quan đến việc giấu tin trong dữ liệu đa phương tiện như: hình ảnh, âm thanh, video hay văn bản. Cụ thể trong luận văn là nghiên cứu về giấu tin và ứng dụng giấu tin trong file âm thanh.

3. Đối tượng và phạm vi nghiên cứu

Với mỗi dữ liệu đa phương tiện có các định dạng, tính chất, đặc trưng khác nhau. Để xây dựng một kỹ thuật giấu tin trên các dữ liệu này thường đòi hỏi các thuật toán phức tạp. Trong luận văn này, ngoài việc tìm hiểu khát quát về giấu tin, các kỹ thuật giấu tin. Nghiên cứu một số kỹ thuật giấu tin bằng các phép biến đổi rời rạc trên số nguyên. Luận văn còn tập trung nghiên cứu về file âm thanh và triển khi thực nghiệm giấu tin trong file âm thanh có định dạng WAV.

4. Phương pháp nghiên cứu

Luận văn sử dụng phương pháp nghiên cứu tư liệu kết hợp với triển khai thực nghiệm. Trên cơ sở nguyên cứu tổng hợp từ các kỹ thuật giấu tin trong file âm thanh. Luận văn đưa ra một kỹ thuật giấu tin mới và tiến hành cài đặt chương trình thực nghiệm giấu tin trong file âm thanh.

5. Ý nghĩa khoa học của đề tài

Về lý thuyết:

- Tiếp cận một hướng nghiên cứu trong lĩnh vực an toàn và bảo mật thông tin. Đây là phương pháp mới và phức tạp. Phương pháp này đang được xem như một giải pháp có nhiều triển vọng cho vấn đề bảo vệ bản quyền, nhận thức thông tin và điều khiển truy cập ứng dụng trong an toàn và bảo mật thông tin.

- Trình bày tương đối đầy đủ một hệ thống lý thuyết giấu tin và đưa ra một kỹ thuật giấu tin trong file âm thanh.

Về thực tiễn:

Với việc triển khai thực tế của đề tài, sẽ góp phần tăng thêm độ an toàn cho các thông tin mật trong việc bảo vệ và truyền thông tin mật.

6. Bố cục của luận văn

Dựa trên đối tượng và phạm vi nghiên cứu, luận văn sẽ được phân làm 3 chương chính với các nội dung cụ thể như sau:

Chương 1. Tổng quan về giấu tin và âm thanh số.

Ở chương này đề tài sẽ đi vào tìm hiểu các khái niệm về giấu tin, mục đích cũng như tính cấp thiết của việc giấu tin trong đa phương tiện, trong đời sống thông tin và truyền tin hiện nay.

Tìm hiểu một môi trường cụ thể mà luận văn sử dụng để giấu tin là âm thanh số. Khái quát một số ứng dụng và các tấn công trên hệ thống giấu tin.

Chương 2. Kỹ thuật giấu tin giấu tin bằng các phép biến đổi rời rạc.

Trong chương này sẽ đi vào tìm hiểu về các phép biến đổi từ miền không gian sang miền tần số.

Tìm hiểu một số kỹ thuật dấu tin dựa trên việc biến đổi bít có trọng số thấp nhất trong một khối bít nhị phân. Cũng trong chương này, tìm hiểu và giải thích một số phép biến đổi trên số nguyên. Dựa vào các kỹ thuật và thuật

toán giấu tin đã được công bố, đề tài sẽ đưa ra ý tưởng của thuật toán đồng thời mô tả chi tiết và cụ thể thuật toán giấu tin trong file âm thanh dựa trên các phép biến đổi rời rạc trên số nguyên.

Đánh giá thuật toán như độ phức tạp hay tính an toàn, bảo mật của thông tin được giấu.

Chương 3. Xây dựng chương trình thử nghiệm

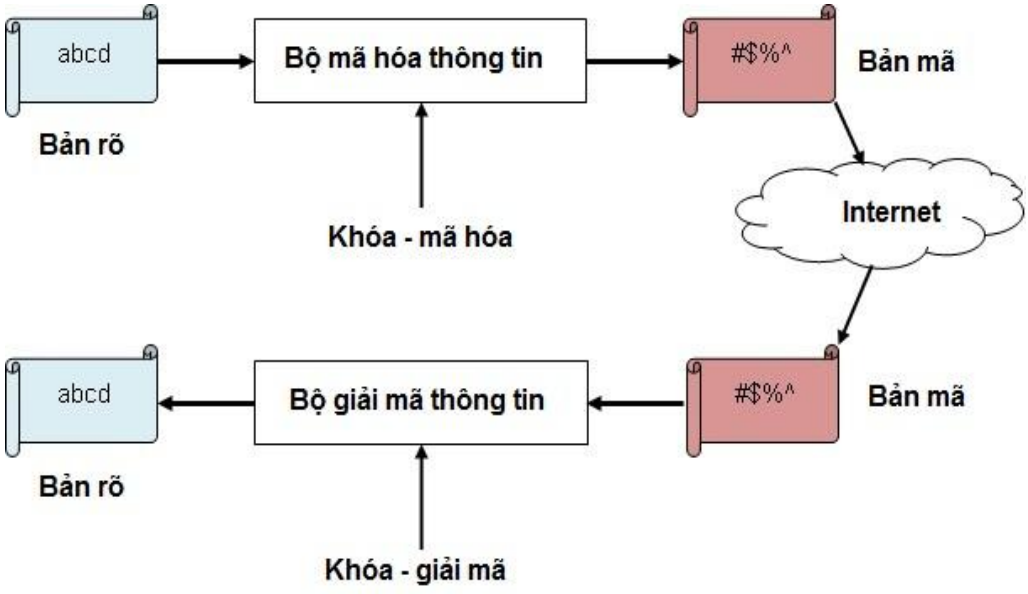
Chương này sẽ đưa ra mục đích, yêu cầu cũng như mô tả sơ đồ chương trình thực nghiệm đã được xây dựng. Lựa chọn định dạng file âm thanh để thử nghiệm thuật toán được mô tả tại Chương 2. Mô tả thuật toán giấu và trích rút thông tin trong file âm thanh. Các kết quả thực nghiệm và đối sánh. Đồng thời đánh giá kết quả thực nghiệm đạt được và đưa ra các khả năng ứng dụng của chương trình thực nghiệm.

CHƯƠNG 1. TỔNG QUAN VỀ GIẤU TIN VÀ ÂM THANH SỐ

1.1. Giới thiệu chung về giấu tin

1.1.1. Mã hóa và giấu tin

Nhiều phương pháp bảo vệ thông tin đã được đưa ra, trong đó giải pháp dùng mật mã được ứng dụng rộng rãi. Thông tin ban đầu được mã hoá, sau đó sẽ được giải mã nhờ khoá của hệ mã. Đã có rất nhiều hệ mã phức tạp được sử dụng như DES, RSA, NAPSACK..., rất hiệu quả và phổ biến.



Hình 1.1. Mô hình mã hóa thông tin

Một phương pháp mới khác đã và đang được nghiên cứu và ứng dụng ở nhiều nước trên thế giới, đó là phương pháp giấu tin. Giấu thông tin là kỹ thuật nhúng (embedding) một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác nhằm giữ bí mật và xác thực thông tin [6].

Một trong những yêu cầu cơ bản của giấu tin là đảm bảo tính chất ẩn của thông tin giấu đồng thời không làm ảnh hưởng đến chất lượng của dữ liệu gốc.

Sự khác biệt chủ yếu giữa mã hoá thông tin và giấu thông tin là mã hoá làm cho các thông tin hiện rõ là nó có được mã hoá hay không, còn với giấu

thông tin thì người ta sẽ khó biết được là có thông tin giấu bên trong. Tuy nhiên, ta có thể kết hợp cả hai phương pháp mã hóa và giấu tin để làm tăng tính bảo mật cho thông tin được giấu.

1.1.2. Phân loại kỹ thuật giấu tin

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 phương tiện chứa tin, hay phân loại dựa theo các mục đích sử dụng....

Theo mục đích sử dụng, giấu thông tin có hai loại:

Giấu thông tin mật

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

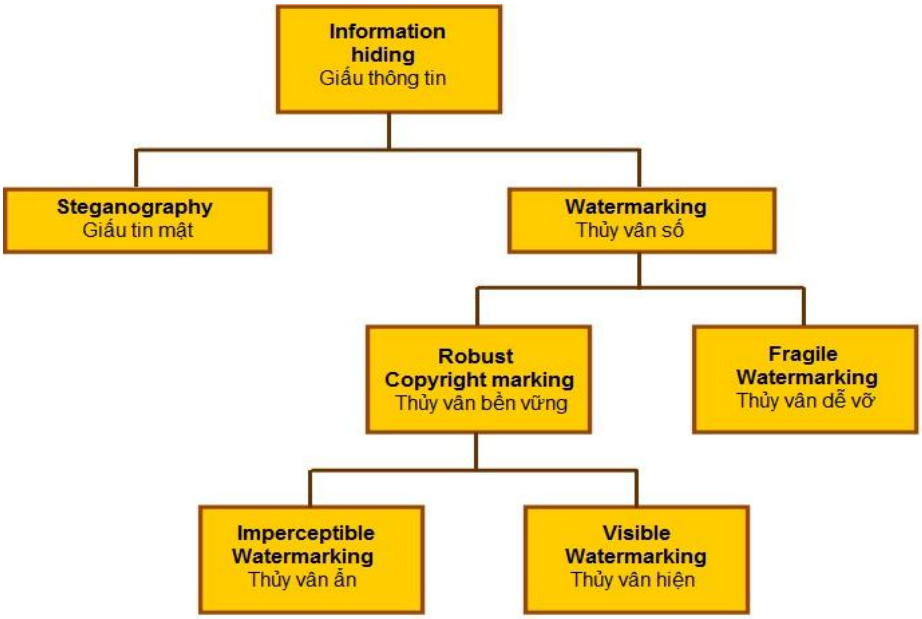
- Độ an toàn của tin giấu (khả năng không bị phát hiện của tin giấu).
- Lượng thông tin tối đa có thể giấu trong một phương tiện chứa cụ thể mà vẫn có thể đảm bảo an toàn.
- Độ bảo mật của thông tin trong trường hợp giấu tin bị phát hiện.

Giấu thông tin 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.

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 trừ khi hủy 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.



Hình 1.2. Một cách phân loại các kỹ thuật giấu tin

	Steganography **Giấu tin mật**	**Watermarking** **Thủy vân số**
Mục tiêu	- Tàng hình các phiên liên lạc để bảo mật thông tin - Dùng trong các liên lạc xác định	- Chủ yếu phục vụ cho mục đích bảo vệ bản quyền - Dùng trong các hoạt động xuất bản
Cách thực hiện	Không làm thay đổi phương tiện chứa thông tin	Có thể tác động nhỏ về cảm nhận tới phương tiện chứa

Yêu cầu	- Giấu được nhiều thông tin nhất - Không quan tâm đến độ bền của phương tiện chứa - Không quan sát được việc giấu thông tin - Không kiểm tra được nếu không có khóa thích hợp	- Chỉ cần nhúng ít dữ liệu - Dữ liệu nhúng cần phải bền vững - Đảm bảo trước các phương pháp nén dữ liệu - Dữ liệu nhúng có thể nhận thấy hay không nhận thấy - Không kiểm tra được nếu không có khóa thích hợp

Bảng 1.1. So sánh giấu thông tin mật và giấu thông tin thủy vân

1.2. Các đối tượng của một bài toán giấu tin

1.2.1. Thông tin mật

Định nghĩa: Là thông tin nhúng vào đối tượng chứa, và là thông tin cần được bảo vệ. Tuỳ theo từng phương pháp cụ thể, thông tin này sẽ được bảo vệ với các mức độ khác nhau.

Đặc điểm:

- Định dạng: Không giới hạn về kiểu định dạng.

- Kích thước: Đây là một trong các yếu tố chính cần phải được xem xét trước khi quyết định sử dụng phương pháp nào. Tuỳ thuộc vào yêu cầu bảo mật và lĩnh vực ứng dụng mà kích thước của nó sẽ khác nhau.

1.2.2. Đối tượng chứa

Định nghĩa: Là đối tượng dùng để chứa thông tin mật. Đối tượng này còn gọi là Cover - < data type >, tuỳ thuộc vào loại dữ liệu mà nó sẽ có những tên khác nhau, ví dụ như Cover - Image, Cover - Audio, Cover - Text…

Đặc điểm:

- Định dạng: Các phương pháp ẩn thông tin ngày nay hầu như đều hỗ trợ định dạng dữ liệu số, nên đối tượng chứa thường có chung đặc điểm là "số".

- Kích thước: Tuỳ mức độ yêu cầu hầu như các phương pháp đều đòi hỏi

kích thước đối tượng chứa lớn hơn nhiều lần kích thước thông tin mật.

1.2.3. Đối tượng đã nhúng

Định nghĩa: Là đối tượng chứa sau khi nhúng thông tin mật, gọi Stego-<data type>, có kiểu dữ liệu tương ứng với đối tượng chứa. Ví dụ nếu đối tượng chứa là ảnh (Cover - Image) thì đối tượng đã nhúng là Stego - Image.

1.2.4. Khoá mật

Định nghĩa: Là khóa tham gia vào quá trình nhúng, tuỳ theo từng thuật toán mà khoá có hay không. Đối tượng này có tên gọi khác là Stego - Key.

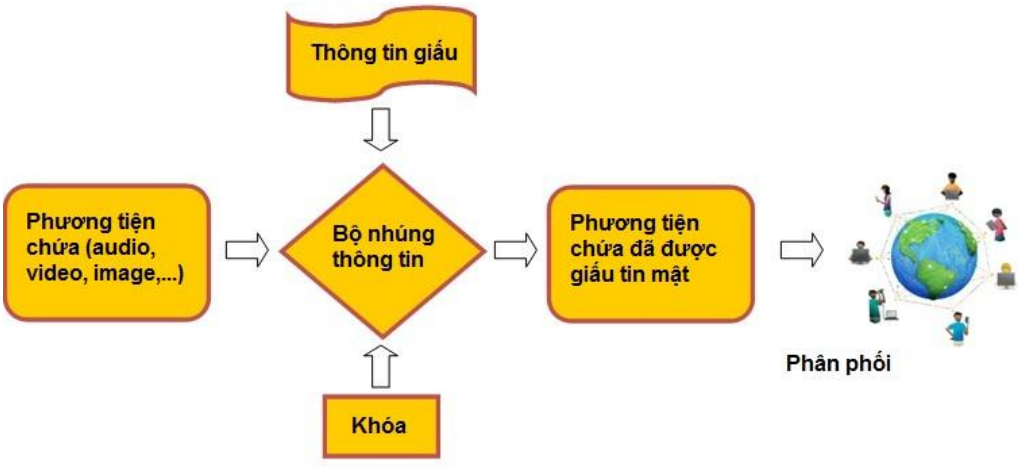
Đặc điểm

- Kích thước: Chiều dài của khoá tuỳ thuộc vào các thuật toán tạo khoá.

- Mức độ yêu cầu: Tuỳ thuộc vào thuật toán ẩn thông tin mà khoá này có thể chỉ dùng trong một giai đoạn mã hoá, hay có thể dùng trong cả hai giai đoạn mã hoá và giải mã.

1.3. Mô hình kỹ thuật giấu tin

Mô hình giấu tin vào môi trường chứa

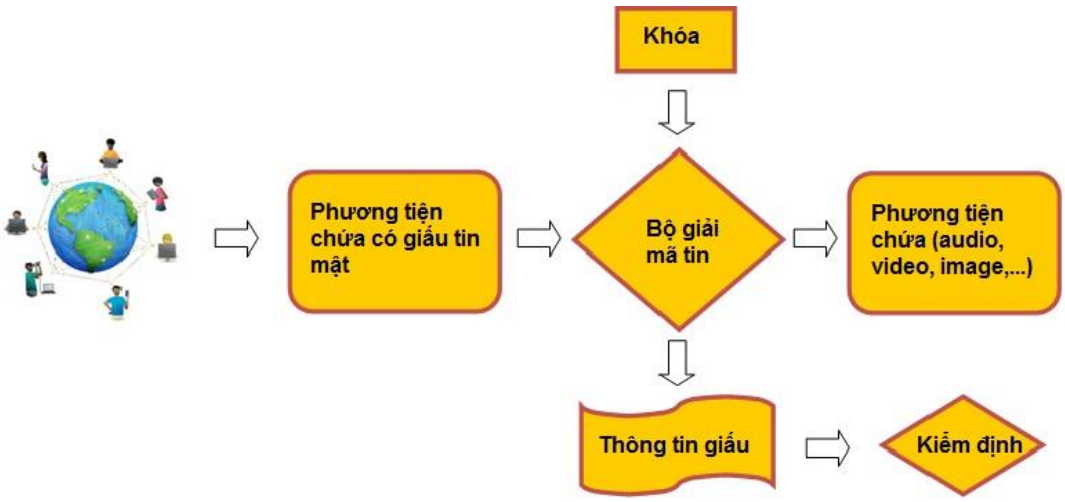


Hình 1.3. Lược đồ chung cho quá trình giấu tin

Hình vẽ trên biểu diễn quá trình giấu thông tin cơ bản. Phương tiện chứa là đối tượng được dùng làm môi trường để giấu tin như văn bản, hình ảnh, âm

thanh, video… Dữ liệu giấu là một lượng thông tin mang ý nghĩa nào đó, tuỳ thuộc vào mục đích của người sử dụng. Thông tin sẽ được giấu vào trong phương tiện chứa nhờ một bộ nhúng thông tin, đây là những chương trình, thuật toán để giấu tin và được thực hiện với một khoá bí mật giống như các hệ mật mã cổ điển. Sau khi giấu ta thu được phương tiện chứa đã mang thông tin và phân phối sử dụng trên mạng.

Mô hình trích rút thông tin



Hình 1.4. Lược đồ chung cho quá trình trích rút thông tin

Sau khi nhận được đối tượng có giấu thông tin, quá trình giải mã được thực hiện thông qua một bộ giải mã tương ứng với bộ nhúng thông tin cùng với khoá của quá trình nhúng. Kết quả thu được gồm phương tiện chứa gốc và thông tin đã giấu. Bước tiếp theo thông tin giấu sẽ được xử lí kiểm định so sánh với thông tin giấu ban đầu.

1.4. Các tiêu chí đánh giá bài toán giấu tin

1.4.1. Khả năng không bị phát hiện

Tính chất này thể hiện ở khả năng khó bị phát hiện, nghĩa là khó xác định một đối tượng có chứa thông tin mật hay không. Để nâng cao khả năng

này, hầu hết các phương pháp ẩn dữ liệu dựa trên đặc điểm của hai hệ tri giác của con người: thị giác và thính giác. Đây là hai cơ quan chủ yếu được dùng để đánh giá chất lượng của tín hiệu. Khả năng này còn được gọi là khả năng giả dạng.

Khả năng khó bị phát hiện tín hiệu mật phụ thuộc vào hai yếu tố sau:

• Kỹ thuật nhúng: Để thực hiện tốt yêu cầu này, ngoài những việc nghiên cứu các thuật toán trong lĩnh vực giấu dữ liệu, người thực hiện phải có kiến thức về định dạng tin mật và đối tượng mang tin.

Tùy theo kỹ thuật nhúng tin, dữ liệu được nhúng có thể phải phù hợp với đối tượng chứa hay không. Với cùng một thông tin mật nhưng nó sẽ rất khó bị phát hiện trên đối tượng này, nhưng lại quá dễ thấy trên đối tượng khác.

• Kinh nghiệm của kẻ tấn công: Việc phát hiện ra tin mật phụ thuộc rất nhiều vào kinh nghiệm của kẻ tấn công. Nếu như kẻ tấn công có nhiều kinh nghiệm thì khả năng phát hiện ra một đối tượng chứa có chứa thông tin mật là không quá khó.

1.4.2. Tính bền vững

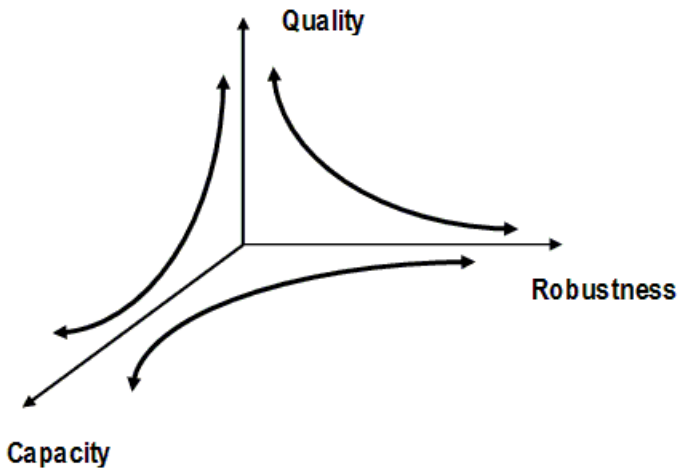
Thể hiện qua việc thông tin mật không hoặc ít bị thay đổi khi vật mang tin chịu tác động bởi các tấn công từ bên ngoài. Ví dụ như: các phép xử lý nén, lọc, biến đổi, tỷ lệ, thay đổi hệ màu,… (đối với hình ảnh) hay việc thay đổi tần số lấy mẫu, độ lớn biên độ,… (đối với âm thanh),… Hiện nay, chưa có kỹ thuật nào có thể bảo được chất lượng này một cách tuyệt đối [7].

1.4.3. Khả năng lưu trữ

Khả năng này thể hiện ở dung lượng tin mật được giấu trong đối tượng chứa. Do tính chất bảo mật nên lượng tin mật được giấu luôn hạn chế. Các phương pháp đều cố làm sao tăng được lượng thông tin cần giấu trong khi vẫn giữ được bí mật.

Tuy nhiên, trong thực tế người ta luôn phải cân nhắc giữa dung lượng và

các chỉ tiêu khác như khả năng không bị phát hiện và tính bền vững.



Hình 1.5. Mối quan hệ giữa các yếu tố trong bài toán giấu tin

1.4.4. Tính vô hình

Tùy theo mục đích sử dụng mức độ yêu cầu của tính chất này có thể khác nhau:

- Steganography: thông tin mật được giấu phải tuyệt đối bí mật, khi đó tính chất này sẽ là trọng tâm của bài toán. Đảm bảo sự vô hình của thông tin trước thị giác hoặc thính giác con người.

- Watermarking: trong một số ứng dụng, người ta có thể đọc (thấy) thông tin thủy vân nhưng không chỉnh sửa hoặc tẩy xóa được. Có những ứng dụng thông tin thủy vân cũng được giữ bí mật.

1.4.5. Độ phức tạp của thuật toán

Chỉ tiêu độ phức tạp trong mã hoá và giải mã cũng là một yếu tố quan trọng trong đánh giá các phương pháp giấu tin trong ảnh. Yêu cầu về độ phức tạp tính toán phụ thuộc vào từng ứng dụng. Ví dụ: một ứng dụng tạo thuỷ vân để đánh dấu bản quyền cần phải có độ phức tạp tính toán cao thì mới đảm bảo chịu được sự tấn công của nhiều hacker nhằm phá huỷ thuỷ vân.

1.5. Một số ứng dụng cụ thể

- Bảo vệ bản quyền tác giả (copyright protection)

Một thông tin nào đó mang ý nghĩa xác định quyền sở hữu của tác giả (ta gọi nó là thuỷ vân) sẽ được nhúng vào các sản phẩm dữ liệu đa phương tiện. Duy nhất người chủ sở hữu hợp pháp các sản phẩm đó có thủy vân và được dùng làm minh chứng cho bản quyền sản phẩm. Việc bảo vệ chống lại các hành vi lấy cắp hoặc làm nhái cần phải có một kỹ thuật để dán tem bản quyền vào dữ liệu này. Yêu cầu kỹ thuật đối với ứng dụng này là thuỷ vân phải tồn tại bền vững cùng với sản phẩm. Để hủy bỏ thuỷ vân này mà không được phép của người chủ sở hữu thì chỉ có cách là phá huỷ sản phẩm.

Hình 1.6. Ảnh gốc Lena và logo của viện khoa học Ấn Độ

Hình 1.7. Ảnh Lena đã được nhúng thủy vân hiển

- Xác thực thông tin hay phát hiện xuyên tạc thông tin (authentication and tamper detection)

Một tập các thông tin sẽ được giấu trong phương tiện chứa sau đó được sử dụng để nhận biết xem dữ liệu trên phương tiện gốc đó có bị thay đổi hay không. Các thuỷ vân nên được "ẩn" để tránh sự tò mò của đối phương. Hơn nữa, việc làm giả các thuỷ vân hợp lệ hay xuyên tạc thông tin nguồn cũng cần được xem xét. Trong các ứng dụng thực tế, người ta mong muốn tìm được vị trí bị xuyên tạc cũng như phân biệt được các thay đổi (ví như phân biệt xem một đối tượng đa phương tiện đã bị thay đổi, xuyên tạc nội dung hay bị nén mất dữ liệu). Yêu cầu chung đối với dữ liệu này là khả năng giấu được nhiều thông tin và thuỷ vân không cần bền vững trước các phép xử lý trên các đối tượng đã được giấu tin.

Hình 1.8 dưới đây là một dạng của xuyên tạc thông tin. Hình 1.8.a là ảnh gốc, hình 1.8.b là ảnh đã bị xuyên tạc.

(1.8.a) (1.8.b)

Hình 1.8. Thông tin bị xuyên tạc

- Giấu vân tay hay dán nhãn (fingeprinting and labeling)

Thuỷ vân được sử dụng để nhận diện người gửi hay người nhận của một thông tin nào đó trong ứng dụng phân phối sản phẩm. Thuỷ vân trong trường

hợp này cũng tương tự như số serial của sản phẩm phần mềm. Mỗi một sản phẩm sẽ mang một thuỷ vân riêng. Ví dụ như các thủy vân khác nhau sẽ được nhúng vào các bản copy khác nhau của thông tin gốc. Với những ứng dụng này sẽ đảm bảo toàn cao cho các thuỷ vân tránh sự xoá dấu vết trong khi phân phối.

- Kiểm soát sao chép (copy control)

Điều mong muốn trong việc phân phối dữ liệu đa phương tiện là có một kỹ thuật chống sao chép trái phép dữ liệu. Các thuỷ vân trong trường hợp này được sử dụng để kiểm soát sao chép đối với các thông tin. Các thiết bị phát hiện ra thuỷ vân thường được gắn sẵn vào trong các hệ thống đọc - ghi. Ví dụ như hệ thống quản lý sao chép DVD đã được ứng dụng ở Nhật. Thuỷ vân mang các giá trị chỉ trạng thái cho phép sao chép dữ liệu như "không được sao chép" (copy never) hay "chỉ được sao chép một lần" (copy once). Sau khi copy xong, bộ đọc - ghi thuỷ vân sẽ ghi thuỷ vân mới chỉ trạng thái mới lên DVD. Các ứng dụng loại này cũng yêu cầu thuỷ vân phải đảm bảo an toàn.

- Giấu tin mật (steganography)

Các thông tin có thể giấu được trong những trường hợp này càng nhiều càng tốt sao cho vẫn đảm bảo yêu cầu là không thể phát hiện được. Việc giải mã để lấy được thông tin cũng không cần phương tiện mang gốc ban đầu. Các yêu cầu về chống tấn công của đối phương không cần cao lắm, thay vào đó là yêu cầu thông tin giấu phải được bảo mật.

1.6. Các tấn công trên các hệ giấu tin

Dữ liệu chứa sau khi được nhúng tin có thể chịu một số tấn công. Các tấn công này có thể làm sai lệch một phần hoặc toàn bộ tin giấu. Sau đây là một số loại tấn công [1].

- *Lấy lại mẫu*

Tấn công này làm thay đổi cấu trúc lưu trữ của file dữ liệu gốc. Một mẫu

dữ liệu trong file mới sẽ được lưu lại bằng một số bít có thể nhiều hoặc ít hơn so với trong file dữ liệu gốc.

- *Lọc thông*

Phương pháp này chỉ chọn lại tần số của dữ liệu thỏa mã điều kiện nằm trong một ngưỡng nào đó. Các phương pháp giấu trên miền tần số sẽ bị ảnh hưởng nếu chịu các tấn công loại này.

- *Thêm nhiễu*

Tấn công này được thực hiện bằng cách thêm các tín hiệu nhiễu vào trong dữ liệu chứa, dẫn đến khi giải mã để trích rút thông tin người nhận sẽ nhận được tin sai với tin giấu.

- *Biến đổi D/A A/D*

Tấn công này thực hiện bằng cách biến đổi vật mang tin mật mà ta gọi là C’ từ dạng số sang dạng analog sau đó lại thực hiện biến đổi từ analog sang dạng số, và kêt quả là được C’’ có thể khác C’.

- *Nén mất thông tin*

Nén dữ liệu là quá trình làm giảm lượng thông tin "dư thừa" trong dữ liệu gốc, do vậy lượng thông tin thu được sau nén thường nhỏ hơn dữ liệu gốc rất nhiều, những kỹ thuật nén mới như fratal cho tỉ lệ nén đến 30%. Việc này dẫn đến khi trích rút thông tin sẽ không còn đầy đủ như thông tin được giấu ban đầu.

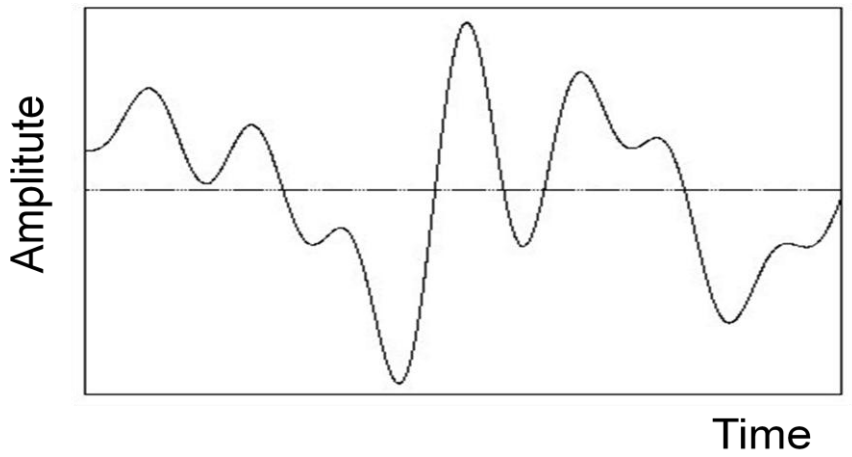
Ngoài ra còn các tấn công khác như nén lượng tử hóa, giảm dữ liệu: cắt bớt, sửa biểu đồ tần hay thủy vân nhiều lần,…

Các tấn công trên các hệ giấu tin có thể làm cho tin giấu nhận được khi giải tin bị sai. Để kiểm chứng lại tin giấu có bị sai không khi giải tin, ta có thể kết hợp các kỹ thuật mã hóa cho phép phát hiện và sửa lỗi.

1.7. Âm thanh số

1.7.1. Khái niệm về âm thanh và âm thanh số

Âm thanh: Là các dao động cơ học (biến đổi vị trí qua lại) của các phân tử, nguyên tử hay các hạt làm nên vật chất và lan truyền trong vật chất như các sóng. Âm thanh, giống như nhiều sóng, được đặc trưng bởi tần số, bước sóng, chu kỳ, biên độ và vận tốc lan truyền (tốc độ âm thanh).



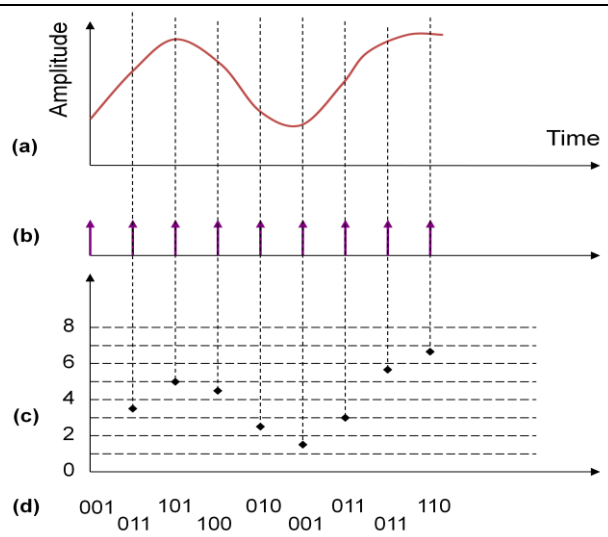
Hình 1.9. Tín hiệu âm thanh

Một đặc tính của âm thanh đây là một sóng dọc, tức là nó là sự lan truyền dao động của đại lượng vô hướng là áp suất. Là sự lan truyền dao động của đại lượng có hướng là vận tốc và vị trí của các phân tử hay nguyên tử trong môi trường, trong đó phương dao động luôn trùng với phương chuyển động của sóng.

Cũng như các sóng cơ học khác, sóng âm mang năng lượng tỉ lệ với bình phương biên độ sóng. Năng lượng đó truyền đi từ nguồn âm đến tai ta. Cường độ âm thanh là lượng năng lượng được sóng âm truyền đi trong một đơn vị thời gian qua một đơn vị diện tích đặt vuông góc với phương truyền âm. Ngoài ra trường độ cũng góp phần ảnh hưởng đến chất lượng âm thanh [8].

Âm thanh số: Là các mẫu được lấy theo phương pháp lượng tử hóa, chuyển đổi giá trị mẫu (âm lượng) liên tục thành giá trị rời rạc. Thông thường các mẫu này đã được lọc để loại bỏ những tần số không mong muốn (giữ lại

tiếng nói từ 50 Hz đến 10 KHz, âm nhạc từ 20 Hz đến 20 kHz).



(a) Tín hiệu gốc dạng tương tự
(b) Lấy mẫu
(c) Giá trị lấy mẫu và lượng tử hóa
(d) Dãy số sau khi số hóa

Hình 1.10. Số hóa tín hiệu âm thanh

1.7.2. Một số định dạng file âm thanh trên máy tính

Với sự phát triển của âm thanh số trong thời gian gần đây rất nhiều các loại định dạng âm thanh đã ra đời, mỗi loại định dạng âm thanh lại có những đặc trưng, cấu trúc khác nhau.

Có ba nhóm định dạng file âm thanh, gồm:

- Các loại định dạng không nén: WAV, AIFF và AU

Cả WAV và AIFF đều được coi là các định dạng âm thanh "không thể mất". Chúng được tạo ra dựa trên nền tảng PCM "điều biến mã xung" (Pulse Code Modulation) với một vài thay dổi nhỏ trong bộ dữ liệu lưu trữ, bên cạnh đó hai loại định dạng này có thể chuyển đổi được cho nhau mà không hề bị giảm chất lượng âm thanh.

Chúng cũng được coi là "không mất dữ liệu" - không bị nén - và một file âm thanh PCM stereo, chẳng hạn có tần số là 44.1kHz và độ nén là 16 bít (chất lượng đĩa CD) thì chất lượng âm thanh có thể lên đến 10MB một phút sau khi được chuyển đổi (convert).

- Các loại định dạng: FLAC, ALAC, APE

FLAC (Free Lossless Audio Codec), ALAC (Apple Lossless Audio Codec) và APE (Monkey's Audio) là các loại định dạng nén âm thanh, chúng sử dụng các thuật toán nén dữ liệu. Sự khác nhau giữa các file nén và các file FLAC đó là FLAC được thiết kế chuyên cho âm thanh thế nên tỉ lệ nén của nó tốt hơn và không bị mất dữ liệu. Thông thường thì file FLAC bằng khoảng một nửa kích cỡ file WAV. Một file FLAC cho âm thanh stereo với chất lượng CD chạy khoảng 5MB mỗi phút.

Hầu hết các định dạng ta sử dụng hàng ngày được xếp vào loại "dễ mất dữ liệu" (lossy); bởi lẽ đôi khi người ta phải giảm chất lượng âm thanh của file xuống để tăng "diện tích sử dụng" của file đó lên.

- Các loại định dạng "dễ mất dữ liệu": MP3, AAC, WMA, Vorbis

Một file MP3 với "chất lượng CD" trung bình chỉ chạy khoảng 1MB mỗi phút. Đây là một file đã được nén tuy nhiên không giống như các định dạng "không mất dữ liệu", ta không thể chuyển đổi lại các file định dạng "dễ mất dữ liệu" để chúng trở lại chất lượng âm thanh tốt được.

Những định dạng "dễ mất dữ liệu" khác nhau sử dụng những thuật toán khác nhau để lưu trữ thông tin, và vì thế chúng thường có chênh lệch về tỷ lệ giữa kích cỡ file và chất lượng âm thanh. Những định dạng "dễ mất dữ liệu" cũng sử dụng số bít để chỉ chất lượng âm thanh, thường vào khoảng "192kbít/s" hay "192kbps".

- Một số định dạng âm thanh trên máy tính

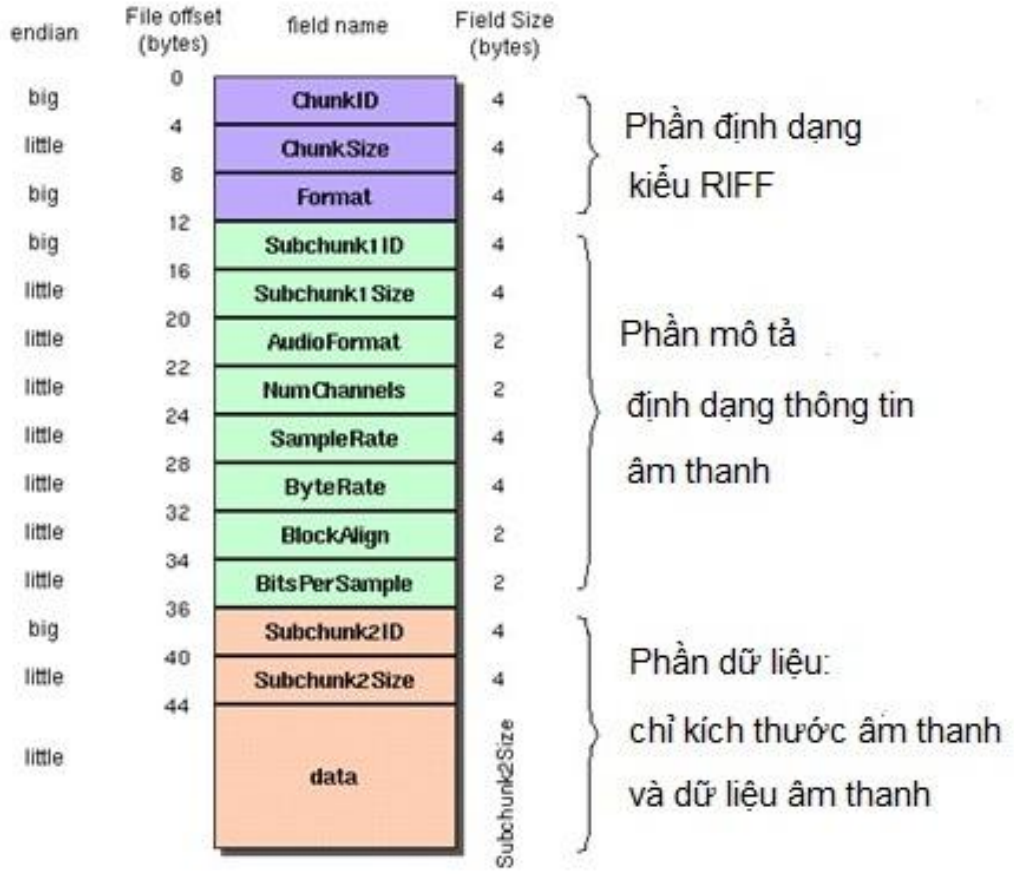
Phần mở rộng file	Công ty sáng tạo	Mô tả
.aiff	Apple	Định dạng tập tin âm thanh tiêu chuẩn được sử dụng bởi Apple được coi là tương đương Apple wav.
.au	Sun Microsystems	Định dạng tập tin âm thanh tiêu chuẩn được sử dụng bởi Sun, Unix và Java. Những

		âm thanh trong các tập tin AU có thể PCM hoặc nén với µ-law, a-law or G729 codecs.
.mp3		Layer-3 định dạng MPEG là định dạng phổ biến nhất để tải về và lưu trữ âm nhạc. Bằng cách loại bỏ các phần của tập tin âm thanh mà chủ yếu là không nghe được, các tập tin mp3 được nén để khoảng một phần mười kích thước của một tập tin PCM tương đương trong khi duy trì chất lượng âm thanh tốt
.wma	Microsoft	Định dạng Windows Media Audio. Được thiết kế với Digital Rights Management (DRM) cho khả năng bảo vệ bản quyền.
.wav		Định dạng tập tin âm thanh chuẩn được sử dụng chủ yếu trong các máy tính Windows. Thường được sử dụng để lưu trữ không nén (PCM), CD-chất lượng các file âm thanh, có nghĩa là nó có thể có kích thước lớn - khoảng 10MB mỗi phút của âm nhạc.
.ra,.ram	RealNetworks	Định dạng Real Audio được thiết kế để truyền tải âm thanh qua Internet. Các định dạng.ra cho phép các tập tin được lưu trữ trong một thời trang khép kín trên một máy tính, với tất cả các dữ liệu âm thanh chứa bên trong tập tin đó.
.acc	Fraunhofer, Dolby, Sony và At&T	AAC (Advanced Audio Coding) được phát triển nhằm thay thế cho định dạng âm thanh đã quá nổi tiếng MP3 để tích hợp trong container MP4 -một container của MPEG-4 tiêu chuẩn hỗ trợ đầy đủ các tính năng phụ
.qt	Apple	Được sử dụng để định dạng đa phương

		tiện từ máy tính Apple
.mid		Được ghi tắt của Music Instrument Digital Interface. Đây là chuẩn đại diện cho thông tin âm nhạc chuyển giao giữa phương tiện điện tử và máy tính

Bảng 1.2. Một số định dạng file âm thanh trên máy tính

1.7.3. Cấu trúc file âm thanh dạng WAV



Hình 1.11. Mô tả định dạng của file.wav

Cũng như AIFF, file âm thanh dạng WAV là định dạng âm thanh "không thể mất". Chúng được tạo ra dựa trên nền tảng PCM với một vài thay đổi nhỏ trong bộ dữ liệu lưu trữ, bên cạnh đó hai loại định dạng này có thể chuyển đổi được cho nhau mà không hề bị giảm chất lượng âm thanh.

Chúng cũng được coi là "không mất dữ liệu" - "không bị nén" - và là một file âm thanh PCM stereo, chẳng hạn nếu được lấy mẫu với tần số 44.1 kHz (44100 lần/giây), độ phân giải 16 bít (tương đương với chất lượng CD) thì 1 phút âm thanh sẽ tiêu tốn tới 10 MB, nghĩa là một bài hát khoảng 5 phút sẽ mất dung lượng 50MB ổ cứng.

Cấu trúc file WAV đươc mô tả cụ thể như sau:

Vị trí byte	Kích thước	Mô tả
00 - 03	4	Chuỗi "RIFF"
04 - 07	4	Số byte theo sau con số này (kích thước tệp tin)
08 - 11	4	Chuỗi "WAVE"

Bảng 1.3. Phần định dạng kiểu RIFF

Vị trí byte	Kích thước	Mô tả
12 - 15	4	Chuỗi "fmt " (ký tự cuối là dấu khoảng trắng, mã ASCII 32)
16 - 19	4	Kích thước FORMAT chunk, mặc nhiên là 16.
20 - 21	2	Định dạng mã hóa âm thanh, thường là 1 (PCM).
22 - 23	2	Số kênh, 1 (Mono) hay 2 (Stereo).
24 - 27	4	Tần số trích mẫu, tính bằng Hz (mẫu/giây).
28 - 31	4	Số byte dữ liệu mỗi giây.
32 - 33	2	Số byte trong một mẫu trích. BytesPerSecond = SampleRate * Channels * AudioSampleSize / 8.
34 - 35	2	Chiều sâu bít (AudioSampleSize), là 8 hoặc 16. BytesPerSample = Channels * AudioSampleSize / 8.

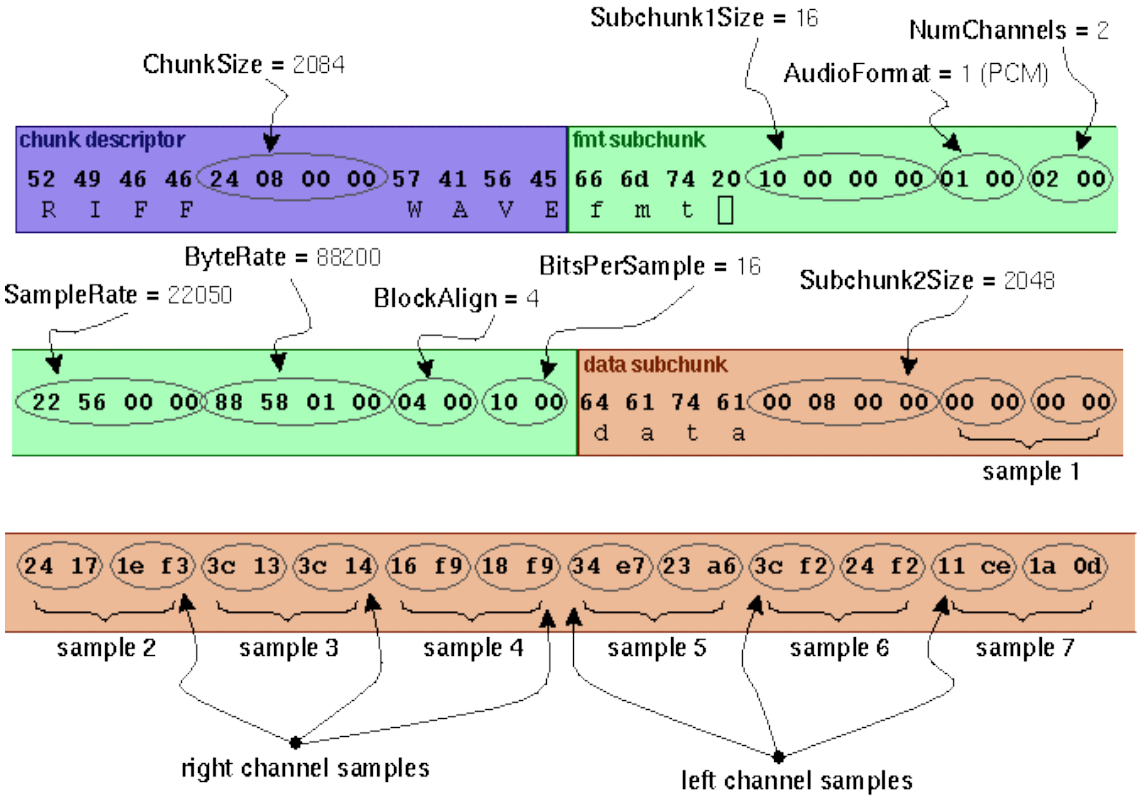
Bảng 1.4. Phần định dạng thông tin âm thanh

Vị trí byte	Kích thước	Mô tả
36 - 39	4	Chuỗi "DATA"
40 - 43	4	Kích thước dữ liệu âm thanh DataSize = Samples * Channels * AudioSampleSize / 8. *Samples là tổng số mẫu trích (ThờiLượng_Giây ****

		TầnSốTríchMẫu).
44 - …		Dữ liệu âm thanh

Bảng 1.5. Phần dữ liệu âm thanh

Thí dụ: Đây là việc mở 72 byte của một tập tin WAVE với các byte hiển thị dạng các số ở hệ thập lục phân (Hexa).



Hình 1.12. Mô tả 72 byte của một file âm thanh WAV

1.8. Một số kỹ thuật giấu tin trong file âm thanh

**** Mã hóa pha***

Phương pháp mã hóa pha dựa vào tính chất là các thành phần của pha không ảnh hưởng đến hệ thông thính giác của con người như nhiễu. Việc giâu tin được thực hiện bằng cách điều chỉnh pha trong phổ pha của dữ liệu số [1].

Quá trình mã hóa pha được thực hiện theo các bước sau

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

- Thực hiện biến đổi Fourier trên mỗi đoạn.

- Tính sự chênh lệch về phase giữa các segment liền kề.

- Giá trị chính xác các pha của các đoạn có thể thay đổi nhưng mối liên hệ về sự khác nhau giữa các segment liên tiếp phải được đảm bảo, vì vậy thông tin giấu chỉ được phép giấu trong vector pha của đoạn đầu tiên. Việc điều chỉnh pha của đoạn đầu được áp dụng dựa trên công thức sau:

$$phase\_new = \begin{cases} \frac{\pi}{2} \ if \ message \ bit = 0 \\ -\frac{\pi}{2} \ if \ message \ bit = 1 \end{cases}$$

- Kết hợp các phase mới cùng với biên độ gốc, ta được các segment mới

- Ghép các segment mới lại để tạo ra chuỗi âm thanh mới đã giấu tin.

Để lấy tin giấu bằng kỹ thuật này, người nhận tin cần phải biết độ dài của segment, sau đó thực hiện DFT để nhận tin.

**** Kỹ thuật trải phổ***

Thông thường các file âm thanh được truyền qua các kênh truyền thông. Các kênh truyên thông này sẽ tập trung dư liệu audio trong vùng hẹp của phổ tần số để duy trì năng lượng và tiết kiệm băng thông. Đặc điểm của hệ thống truyền thông trải phổ là độ rộng phổ của tín hiệu bị "trải" ra, lớn hơn nhiều lần so với tốc độ bít của thông tin cần truyền. Độ dư thừa của băng thông được sử dụng như một tiềm năng cho các phương pháp lập mã tự sửa sai, dẫn đến khả năng chống nhiễu cao của hệ thống thông tin trải phổ so với các phương pháp truyền tin khác. Một thành phần quan trọng trong kỹ thuật truyền tin trải phổ chính là chuỗi giả ngẫu nhiên. Vì chuỗi này mang đặc trưng của nhiễu nên tín hiệu trải phổ có ưu thế về độ bảo mật

**** Kỹ thuật giấu tin dựa vào tiếng vang (Echo)***

Giấu tin dựa vào tiếng vang bằng cách nhúng thông tin cần giấu vào

tiếng vang trong dữ liệu gốc. Cũng như trải phổ giấu tin dựa vào tiếng vang cho phép một tốc độ truyền dữ liệu cao hơn và bền vững trước tấn công

Để nhúng dữ liệu cần giấu ta cần thực hiện thay đổi ba tham số của tiếng vang đó là: biên độ ban đầu, tỷ lệ phân rã và độ trễ. Khi thời gian giữa tín hiệu gốc và tiếng vang giảm xuống, hai tín hiệu đó có thể trộn lẫn và người nghe khó có thể phân biệt giữa hai tín hiệu. Số lượng tin giấu sẽ liên quan đến thời gian trễ của tiếng vang. Để giấu được nhiều hơn một bít, tín hiệu gốc được chia thành các đoạn ngắn hơn và mỗi đoạn sau đó có thể tạo tiếng vang để giấu số bít mong muốn.

**** Kỹ thuật mã hóa echo***

Bằng cách dùng thời gian trễ khác nhau giữa tín hiệu gốc và tiếng vang để thể hiện tương ứng giá trị nhị phân 1 hoặc 0, theo cách đó dữ liệu được giấu vào file âm thanh. Cũng như kỹ thật giấu dự vào tiếng vang, ta có thể chia tín hiệu gốc thành các đoạn ngắn hơn và mỗi đoạn sau đó có thể tạo tiếng vang để giấu số bít mong muốn. Một cách tiếp cận khác là tiến hành mã hóa chuỗi bít theo một cách nào đó giúp ta phát hiện ra lỗi. Thay vì giấu trực tiếp L bít vào đối tượng chứa, ta biến đổi chuỗi bít bằng cách bổ sung một số bít vào S nhằm mục đích kiểm tra lỗi.

Tổng kết chương 1

Chương 1 đã trình bày một hệ thống lý thuyết bao gồm các khái niệm, các tính chất, ứng dụng và các kỹ thuật giấu tin. Giấu tin được hiểu là chèn một thông tin vào một đối tượng chứa nào đó mà không làm thay đổi kích thước dữ liệu của đối tượng chứa đồng thời tỷ lệ thay đổi về chất lượng của đối tượng chứa là thấp nhất. Trong chương này cũng đã trình bày về các khái niệm về âm thanh và âm thanh số. Tìm hiểu một số định dạng âm thanh trên máy tính và cấu trúc file âm thanh dạng wav. Đồng thời khái quát một số kỹ thuật giấu tin trên file âm thanh.

CHƯƠNG 2. KỸ THUẬT GIẤU TIN BẰNG CÁC PHÉP BIẾN ĐỔI RỜI RẠC

2.1. Các phép biến đổi từ miền không gian sang miền tần số

Người ta nhận thấy rằng, việc nghiên cứu tín hiệu và hệ thống rời rạc trong miền thời gian gặp nhiều khó khăn trong việc tính toán và phân tích hệ thống trong miền này như việc tính tích chập, giải phương trình sai phân,… [3]. Để khắc phục các hạn chế trên người ta sử dụng các phương pháp gián tiếp để nghiên cứu tín hiệu và hệ thống bằng cách chuyển từ cách biểu diễn từ miền không gian sang một miền trung gian, miền này thuận lợi cho việc nghiên cứu, xử lý và có thể chuyển đổi ngược lại được.

Phương pháp xử lý gián tiếp này sẽ làm đơn giản đi rất nhiều các công việc mà chúng ta thường gặp phải trong quá trình xử lý trực tiếp trong miền biến số độc lập. Có nhiều phép biến đổi cho dữ liệu âm thanh, trong đó có một số phép biến đổi thường được sử dụng như: biến đổi Fourier, biến đổi cosin rời rạc, biến đổi wavelet,…

2.1.1. Phép biến đổi Fourier

Phổ Fourier là một hàm chuyển đổi rất hay được dùng trong xử lý tín hiệu số (DSP: Digital signal processing). Nó có thể được hiểu đơn giản là hàm biểu thị sự tương quan của 1 tín hiệu nào đó với 1 tập hợp các hàm sin và cos. Tại sao phải cần tìm sự tương quan này? Có nhiều lý do, nhưng lý do chính có lẽ là do sin và cos là những hàm tuần hoàn hay sử dụng nhất trong thông tin bởi khả năng mang thông tin của chúng. Một tín hiệu nếu được chuyển thành các hàm sin và cos thì sẽ có khả năng dùng trong thông tin. Như ta đã biết, các hàm sin, cos được đặc trưng bởi 3 thông số: biên độ, tần số và pha. Trong miền thời gian, cả 3 thông số này đều được biểu diễn theo hàm của thời gian. Phổ Fourier biểu diễn các thông số biên độ và thời gian theo thông số tần số. Như vậy mục đích chính của ta là chuyển đổi 1 tín hiệu (từ

miền thời gian) sang miền tần số. Việc chuyển đổi này cho phép ta có thể xử lý tín hiệu một cách chính xác và tiện lợi hơn nhiều do làm việc trực tiếp với tần số, tài nguyên quan trọng bậc nhất của thông tin.

Biến đổi Fourier (FT) của tín hiệu rời rạc x(n) được định nghĩa:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

Như vậy, biến đổi Fourier đã chuyển việc biểu diễn tín hiệu x(n) trong miền biến số độc lập n thành biểu diễn tín hiệu $X(e^{j\omega})$ trong miền tần số ω. Vậy $X(e^{j\omega})$ là hàm số phức của biến số ω.

Biến đổi Fourier ngược (IFT – Inverse Fourier Transform)

Hàm $X(e^{j\omega})$ là một hàm tuần hoàn với chu kỳ 2π vì vậy chúng ta có thể triển khai hàm $X(e^{j\omega})$ thành chuỗi Fourier trong khoảng [-π, π] và coi các hệ số của khai triển này chính là X(n) tứ là có thể tìm thấy các giá trị của X(n) từ $X(e^{j\omega})$. Công thức biến đổi Fourier ngược:

$$X(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{j\omega n}\, d\omega$$

Phép biến đổi Fast Fourier

Đây là phép biến đổi thường được sử dụng cho nhiều ứng dụng trong khoa học, kỹ thuật và toán học [3]. Phép biến đổi này có độ phức tạp là O(nlgn).

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}$$

$$x(j) = (1/N)\sum_{k=1}^{N} X(k)\omega_N^{-(j-1)(k-1)}$$

2.1.2. Phép biến đổi Cosin rời rạc

Các quá trình nén là xác định các thông số tin dư thừa trong miền không

gian của hình ảnh hay tín hiệu video. Nén không gian được thực hiện bởi phép biến đổi DCT (Discrete Cosine Tranform). DCT biến đổi dữ liệu dưới dạng biên độ thành dạng tần số. Mục đích của quá trình biến đổi là tách liên kết của từng ảnh con, hoặc gói càng nhiều năng lượng của ảnh con vào một phần nhỏ các hệ số hàm truyền. Việc mã hoá và truyền chỉ thực hiện đối với các hệ số năng lượng này, và có thể cho kết quả tốt khi tạo lại tín hiệu Video có chất lượng cao.

DCT đã trở thành tiêu chuẩn quốc tế cho các hệ thống mã chuyển vị bởi nó có đặc tính gói năng lượng tốt, cho kết quả là số thực và có thuật toán nhanh.

Phép biến đổi Cosin rời rạc của dãy X(n):

$$X(k) = \sum_{n=0}^{N-1} x(n).W_N^{kn} \text{ Với n} = 0,1,2,\dots,\text{N} - 1 \qquad (2.1)$$

Trong đó:

$$W_N^{kn} = e^{-j\,2\pi kn\,/N} = W^{kn} = Cos\left(\frac{2\pi kn}{N}\right) - j.Sin(\frac{2\pi kn}{N})$$

Phép biến đổi rời rạc ngược của X(k) là:

$$X(n) = \frac{1}{N}.\sum_{n-0}^{N-1} X(k).W_N^{-kn} \text{ Với n} = 0,1,2,\dots,\text{N} - 1 \qquad (2.2)$$

Trong (2.1) và (2.2) X(k) và X(n) có thể là số phức.

X(n) = a(n) + j.b(n)

X(k) = A(k) + j.B(k).

Do đó:

$$A(k) + j.B(k) = \sum_{n=0}^{N-1} [a(n) + j.b(n)][Cos\left(\frac{2\pi kn}{N}\right) - j.Sin\left(\frac{2\pi kn}{N}\right)]$$

$$A(k) = \sum_{n=0}^{N-1} [a(n).Cos\left(\frac{2\pi kn}{N}\right) - b(n).Sin\left(\frac{2\pi kn}{N}\right)]$$

$$B(k) = \sum_{n=0}^{N-1} [b(n).Cos\left(\frac{2\pi kn}{N}\right) - a(n).Sin\left(\frac{2\pi kn}{N}\right)]$$

2.1.3. Phép biến đổi Wavelet

Biến đổi waveler rời rạc (Discrete Wavelet Transform) bắt đầu với một wavelet mẹ là một tín hiệu thời gian chu kỳ ngắn và có trung bình bằng không, ψ(t) kết hợp với chuỗi thời gian cần xét f(t) để lọc ra chuỗi thời gian. Wavelet mẹ được dãn ra theo thời gian ở các hệ số dãn cố định tạo thành các wavelet con. Trong mỗi tỷ lệ đều có chứa f(t). Do vậy wavelet mẹ và các bản ảnh trễ của nó tạo thành một dãy các bộ lọc chồng nhau mà mỗi đoạn của dãy có cùng hệ số phẩm chất (Q_w = độ rộng băng tần / tần số trung tâm).

Waveler là các hàm cơ sở $\omega_{jk}(t)$ trong miền thời gian liên tục. Cơ sở là là một tập hợp các hàm độc lập tuyến tính mà có thể dùng để tạo ra các hàm f(t).

$$f(t) = \sum_{j,k} b_{jk}\, \omega_{jk}(t) \qquad (2.3)$$

Đặc tính của waveler là các hàm $\omega_{jk}(t)$ đề được xây dượng từ một hàm wareler mẹ ω(t). Wareler là một sóng (một xung) nhỏ. Thông thường nó bắt đầu ở thời điểm t = 0 và kết thúc ở thời điểm t = N.

Wareler đã được trễ đi ω_{0k} bắt đầu từ t = k và kết thúc ở t = k + N. Các wareler được tỷ lệ ω_{j0} thì bắt đầu từ t = 0 và kết thúc ở $t = N/2^j$. Đồ thị của chúng được nén lại với hệ số là 2^j, trong khi đồ thị của ω_{0k} thì lại được dịch đi (về bên phải) một lượng là k:

Nén: $\omega_{j0} = \omega(2^j t)$

Trễ: $\omega_{0k} = \omega(t - k)$

Một wareler điển hình ω_{jk} vừa bị nén đi j lần, vừa bị làm trễ đi k lần có

công thức:

$\omega_{jk}(t) = \omega(2^j k - t)$

Wareler còn có một tính chất quan trọng khác đó là tính trực giao (orthogonality). Các wareler trực giao khi tích vô hướng (inner product) của chúng bằng 0.

$$\int_{-\infty}^{\infty} \omega_{jk}(t)\omega_{JK}(t)dt = \text{tích vô hướng của } \omega_{jk} \text{ và } \omega_{JK} = 0 \qquad (2.4)$$

Trong trường hợp này thì các wareler đó sẽ có một cơ sở wareler trực giao đối với không gian hàm.Cơ sở đó tương ứng với một tập hợp các trục tạo ra với nhau một góc 90^0. Tính trực giao dẫn đến một công thức đơn giản hơn đối với mỗi hệ số b_{JK} trong công thức mở rộng của f(t). Nhân f(t) trong (2.3) với $\omega_{JK}(t)$ và lấy tích phân ta được:

$$\int_{-\infty}^{\infty} f(t)\omega_{JK}(t)dt = b_{JK} \int_{-\infty}^{\infty} (\omega_{JK}(t))^2 dt \qquad (2.5)$$

Phương trình giới hạn tất cả các tích phân của ω_{jk} nhân với ω_{JK}, trừ trường hợp j = J và k = K. Thành phần đó tạo ra $(\omega_{JK}(t))^2$. Khi đó b_{JK} là tỷ số của hai tích phân trong phương trình (2.5).

2.2. Một số kỹ thuật giấu tin dựa trên biến đổi khối bít nhị phân

Các thuật toán giấu tin hầu như chỉ tập trung vào đối tượng mang tin là hình ảnh và video, rất ít thuật toán được phát triển trên vật mang tin là âm thanh. Nghiên cứu giấu tin trên âm thanh số là tập trung khai thác về cảm nhận hệ thính giác của con người. Theo các nghiên cứu về sinh học cho thấy: hệ thính giác của con người khá nhạy cảm với nhiễu. Ngoài ra các kỹ thuật tấn công trên âm thanh số cũng rất nhiều và đa dạng. Có thể vì vậy mà việc nghiên cứu các kỹ thuật giấu tin trên âm thanh số ít khi được thực hiện. Vì vậy, việc giấu tin trong âm thanh thường là khó hơn trong các dữ liệu đa

phương tiện khác [1].

Sau đây là một số kỹ thuật giấu tin trong file âm thanh dựa trên việc biến đổi khối bít nhị phân, mà cụ thể là biến đổi bít có trọng số thập nhất.

2.2.1. Mã hóa LSB (Least Significant Bit)

LSB nghĩa là bít có trọng số thấp nhất hay là bít có ảnh hưởng ít nhất đến việc quyết định chất lượng của âm thanh. Kỹ thuật này được sử dụng nhiều trong các thuật toán giấu tin. Kỹ thuật sử dụng bít ít quan trọng nhất (thường là bít cuối) của chuỗi bít được sử dụng làm môi trường để giấu tin mật. Vì vậy, khi ta thay đổi bít này chất lượng của âm thanh gần như không thay đổi so với chất lượng âm thanh ban đầu.

Thí dụ: ta có chuỗi 8 bít như sau

1	0	0	1	1	1	0	0
7	6	5	4	3	2	1	0

Nếu bít thông tin mật cần giấu là 0 thì thì ta coi như đã được thực hiện.

Nếu bít thông tin cần giấu là 1.

Khi đó ta sẽ biến đổi bít thứ 0 từ 0 thành 1 và khi đó bít 1 cần giấu đã được thực hiện và chuỗi bít sẽ được biến đổi là:

1	0	0	1	1	1	0	**1**
7	6	5	4	3	2	1	0

Hình 2.1. Minh họa kỹ thuật LSB

Với kỹ thuật này, ta có thể coi các mẫu trích âm thanh là một chuỗi bít để thực hiện việc ẩn giấu 1 bít thông tin tin mật. Khi đó giá trị của mẫu trích nếu bị thay đổi cũng chỉ thay đổi 1 đơn vị.

Thí dụ: Với một file âm thanh có giá trị chiều sâu của bít là 1 nghĩa là 8 bít hay gọi là một byte. Thì khi đó giá trị của byte sẽ là từ 0 đến 255. Giả sử ta

có một byte có giá trị là 135. Khi đó để giấu bít 0 vào byte dữ liệu này. Sau khi thực hiện biến đổi thì byte này có giá trị là 134. Nếu cũng giá trị của byte là 135 mà bít cần giấu là 1 thì nghiễm nhiên ta không cần biến đổi. Còn nếu trường hợp byte có giá trị là 134 mà bít cần giấu là 1 thì khi đó ta biến đổi byte này thành giá trị là 135.

Như vậy với việc thay đổi giá trị 1 đơn vị trong 1 byte âm thanh sẽ không ảnh hưởng nhiều đến chất lượng âm thanh ban đầu.

Việc trích rút thông tin từ thuật toán này cũng rất đơn giản. Với việc ta chỉ cần xác định giá trị của các mẫu chứa tin ban đầu và lấy giá trị của bít có trọng số thấp nhất trong mẫu để trích rút thông tin.

2.2.2. Mã hóa Parity (Parity Coding)

Hay còn gọi là mã hóa chẵn lẻ. Thay vị sử dụng các mẫu riêng lẻ để giấu tin như kỹ thuật LSB, kỹ thuật mã hóa chẵn lẻ lại chia dữ liệu thành các nhóm mẫu và thực hiện việc giấu từng bít tin trong các nhóm mẫu này.

Parity-bit là dùng một bít để báo hiệu số lượng bít có giá trị là 1 trong một nhóm bít cho trước là một số chẵn hay là một số lẻ. Nếu Parity-bit của mẫu không trùng với bít tin cần giấu, ta có thể thay đổi một bít nào đó (thường cũng là bít có trọng số thấp nhất) trong mẫu này để phù hợp với bít tin cần giấu.

Thí dụ: giả sử ta dùng bít 1 để báo hiệu số lượng bít 1 trong mẫu là lẻ và dùng bít 0 để báo hiệu số lượng bít 1 trong mẫu là chẵn và vị trí của bít báo hiệu là thứ 0.

Khi đó, nếu bít tin cần giấu là bít 1 và ta có chuỗi bít mẫu sau:

1	0	0	1	1	1	0	1
7	6	5	4	3	2	1	0

Với chuỗi này, ta thấy có 5 bít có giá trị là 1 và đương nhiên bít 1 của ta

đã được giấu.

Nếu như bít tin cần giấu vẫn là bít 1 và ta lại có chuỗi bít mẫu sau:

1	0	1	1	0	0	1	0
7	6	5	4	3	2	1	0

Với chuỗi này, ta thấy có 4 bít có giá trị là 1. Ta có thể thay đổi một bít để chuỗi thỏa mãn có số lượng bít 1 là một số lẻ như sau:

1	0	1	1	0	0	1	**1**
7	6	5	4	3	2	1	0

Hình 2.2. Minh họa kỹ thuật mã hóa Parity

Như ta thấy bít được thay đổi chính là bít có trọng số thấp nhất trong nhóm bít mẫu.

Với kỹ thuật LSB ta chỉ việc dùng 1 bít riêng lẻ để giấu tin nhưng với kỹ thuật Parity-bit việc giấu tin việc thay đổi (nếu có) bít có trọng số thấp nhất còn chịu tác động bởi giá trị các bít của cả nhóm bít mẫu.

Việc trích rút thông tin từ kỹ thuật này cũng được thực hiện hết sức đơn giản. Bằng cách ta xác định số bít của nhóm mẫu. Thực hiện tính tổng giá trị của nhóm mẫu này với một số nguyên:

- Nếu là số nguyên lẻ ta sẽ trích rút được bít thông tin mật là bít 1.
- Nếu là số nguyên chẵn ta sẽ trích rút được bít thông tin mật là bít 0.

Đánh giá kỹ thuật mã hóa LSB và Parity-bit

Ưu điểm của 2 kỹ thuật này là dễ cài đặt, cho phép giấu được nhiều dữ liệu mật. Với kỹ thuật LSB có thể tăng thêm dữ liệu giấu bằng cách sử dụng hai bít. Tuy nhiên cách này có thể làm tăng nhiễu trên dữ liệu gốc dẫn đến việc dễ bị đối phương phát hiện và thực hiện các tấn công. Vì vậy nếu muốn sử dụng 2 kỹ thuật này ta lại phải đặc biệt quan tâm đến việc lựa chọn dữ liệu

chứa tin mật.

Để tăng độ an toàn khi sử dụng các kỹ thuật này, ta có thể sử dụng bộ sinh số nguyên ngẫu nhiên để sinh ra các vị trí mẫu được chọn giấu chứ không phải là các mẫu liên tục. Bộ sinh số nguyên ngẫu nhiên này có thể sử dụng một khóa bí mật key như là một phần tử được dụng để khởi tạo bộ sinh số. Như lưu ý là bộ sinh số không tạo ra các vị trí mẫu trùng nhau để tránh trường hợp một vị trí có thể được giấu hai lần. Khóa key này sẽ được sửa dụng trong cả hai quá trình giấu và kết xuất tin được giấu

Hạn chế của hai kỹ thuật LSB và Parity là do tai người khá nhạy cảm với nhiễu nên những thay đổi trên dữ liệu chứa có thể dễ nhận ra đồng thời cả hai kỹ thuật này đều không bền vững thông tin dễ bị mất khi thực hiện lấy mẫu lại

2.3. Thuật toán giấu tin bằng các phép biến đổi rời rạc trên số nguyên

Như đã trình bày tại phần mở đầu của luận văn. Các kỹ thuật giấu tin có thể chia ra làm hai loại. Nhóm thứ nhất là các phương pháp che giấu thông tin trực tiếp. Nhóm này thường sử dụng các bít ít quan trọng nhất của một khối bít nhị phân được sửa đổi để giấu thông tin. Nhóm thứ hai lại che giấu thông tin thông qua các phép biến đổi chẳng hạn như phép biến đổi Cosin hay wavelet rời rạc được sử dụng rộng rãi [4]. Kỹ thuật mà luận văn này hướng đến thuộc nhóm thứ nhất trong hai nhóm trên. Che giấu thông tin trực tiếp thông qua việc sửa đổi bít có trọng số thấp.

2.3.1. Một số phép biến đổi rời rạc trên số nguyên

*** Phép nhân**: (ký hiệu ⊗) là nhân các phần tử của hai ma trận có cùng kích thước m×n

$(F \otimes G)_{i,j} = F_{i,j} \times G_{i,j}$, i = 1, 2, …, m và j = 1, 2, …, n.

Thí dụ:

1	0	1
1	1	0

1	1	0
0	1	1

1	0	0
0	1	0

0	0	1

F

1	1	0

G

0	0	0

F ⊗ G

*** Phép AND**: (ký hiệu là ^) là biểu thị phép toán AND trên từng cặp bít của hai số nguyên không âm.

Thí dụ: 7 ^ 12 = 0111 ^ 1100 = 0100 = 4

*** Phép XOR**: (ký hiệu là ⊕) là biểu thị phép toán XOR trên từng cặp bít của hai số nguyên không âm.

Thí dụ: 5 ⊕ 12 = 0101 ⊕ 1100 = 1001 = 9

*** Phép SUM(D)**: Với ma trận D mà mỗi phần tử của nó là một số nguyên không âm. SUM(D) là tổng các phần tử của D.

Thí dụ: SUM(F) là tổng tất cả các phần tử của F. SUM(F) = 5

*** Phép XSUM(D)**: Với ma trận D mà mỗi phần tử của nó là số nguyên không âm. XSUM(D) hoặc $\sum_{i,j}^{\oplus} D_{i,j}$ là tổng XOR của tất cả các phần tử của D.

Thí dụ: XSUM(F) là tổng XOR tất cả các phần tử của F. XSUM(F) = 1

Nhận xét 1:

Nếu $D_{i,j} \in \{0, 1, \ldots, 2^r - 1\}$ $\forall(i,j)$ thì $XSUM(D) \in \{0, 1, \ldots, 2^r - 1\}$

2.3.2. Thuật toán Wu-Lee

Thuật toán này được đề xuất bởi hai tác giả là M.Y.Wu và J.H.Lee. Với ý tưởng là phân tích đối tượng mang tin thành một ma trận các điểm với các giá trị 0 và 1.

Xét hai ma trận có cùng kích thước F và K. Khi đó:

F ⊗ K được coi là phép toán Nhân giữa các cặp bít tương ứng của hai ma trận.

Thí dụ:

1	0	1
1	1	0

1	1	0
0	1	1

1	0	0
0	1	0

0	0	1

F

1	1	0

K

0	0	0

F ⊗ K

F ⊕ K được coi là phép toán XOR giữa các cặp bít tương ứng của hai ma trận.

Thí dụ:

1	0	1
1	1	0
0	0	1

F

1	1	0
0	1	1
1	1	0

K

0	1	1
1	0	1
1	1	1

F ⊕ K

Với một ma trận M bất kỳ SUM(M) là tổng của tất cả các phần tử trong M. Với ma trận F trên ta có SUM(F) = 5.

Thuật toán Wu-Lee được mô tả như sau:

Giả sử ta có cần giấu một bít thông tin mật b vào một khối tin F của vật mang tin, một khóa bí mật K. Trong đó khóa K và F là các ma trận các số nhị phân có kích thước m × n. Khi đó việc nhúng bít thông mật b vào F sẽ được thực hiện như sau:

Bước 1: $s = SUM(F \otimes K) \bmod 2$

Nếu s = b thì G = F và kết thúc, nếu không thì sang bước 2

Bước 2: $d = s \oplus b$

Tìm phần tử tại vị trí (u,v) mà $K_{u,v} = d$ và $F_{u,v} = 0$

$F_{u,v} = 1 - F_{u,v}$

$G = F$ và kết thúc

Để trích rút bít thông tin mật b ta chỉ cần biến đổi ngược

$b = SUM(F \otimes K) \bmod 2$

Thí dụ: Giả sử ta có bít thông tin mật cần giấu b = 1, ma trận F, và ma trận khóa K được mô tả như trên. Khi đó quá trình giấu bít b vào F được thực hiện như sau:

Bước 1: $s = SUM(F \otimes K) \bmod 2 = 0$

$s \neq b$

Bước 2: $d = s \oplus b = 1$

Tìm một phần tử tại vị trí (u,v) mà $K_{u,v} = 1$ và thỏa mãn $F_{u,v} = 0$. Tại đây ta thấy có 4 vị trí mà $F_{u,v} = 0$ là $F_{1,2}$, $F_{2,3}$, $F_{3,1}$, $F_{3,2}$

Ta chọn một vị trí bất kỳ trong 4 vị trí trên của F. Giả sử ta chọn $F_{1,3}$

$F_{1,2} = 1 - F_{1,2} = 1$

Vậy ta có ma trận G là:

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

G

Để trích rút thông tin ta thực hiện theo công thức sau

$b = SUM(F \otimes K) \bmod 2$

1	1	0
1	1	0
0	0	1

G

1	1	0
0	1	1
1	1	0

K

1	1	0
0	1	0
0	0	0

F ⊗ K

b = 3 mod 2 = 1

Đánh giá thuật toán:

Thuật toán Wu-Lee được biết đến như một thuật toán đơn giản nhất cho việc giấu tin mật vào hình ảnh nhị phân [4]. Trong thuật toán này ma trận K có kích thước m×n được sử dụng ngẫu nhiên một như là một chìa khóa bí mật

và có thể giấu một bít b vào khối tin F bằng cách thay đổi một bít của F để có được ma trận số nhị phân G thỏa mã điều kiện *SUM(G⊗K) mod 2 = b*

Nhận xét 2: Thuật toán Wu-Lee thực hiện việc sửa 1 bít trong khối tin F nhưng chỉ có thể giấu 1 bít thông tin mật vào trong khối tin F. Với thuật toán này ta chưa thể mở rộng để có thể giấu một chuỗi các bít tin mật.

Xét một thuật toán mới bằng cách sử dụng các phần tử trong *XSUM(G⊗K)* thay vì trong *SUM(G⊗K)* của thuật toán Wu-Lee để từ thuật toán mới này ta có thể xây dựng một thuật toán hoàn toàn mới có khả năng gia tăng lượng thông tin mật được giấu.

2.3.3. Thuật toán Wu-Lee cải tiến

Thuật toán này thay đổi ít nhất một phần tử của F để có được ma trận G thỏa mãn điều kiện:

XSUM(G ⊗ K) = b

Mô tả thuật toán:

Bước 1: s = *XSUM(G ⊗ K)*

Nếu s = b thì G = F và kết thúc, nếu không thì sang bước 2

Bước 2: d = s ⊕ b

Tìm một phần tử tại vị trí (u,v) mà $K_{u,v}$ = d

$F_{u,v} = 1 - F_{u,v}$

G = F và kết thúc.

Nhận xét 3: Với cả 2 thuật toán Wu-Lee và Wu-Lee cải tiến, giá trị d luôn bằng 1 vì vậy bước 2 được thực hiện, ma trận K thỏa mãn điều kiện sau:

$\{1\} \subset \{K_{i,j} | i = 1, 2, \ldots, m \text{ và } j = 1, 2, \ldots, n\}$

Thí dụ: Ta cần giấu bít b = 1 vào ma trận điểm ảnh F và sử dụng ma trận ngẫu nhiên K

Khi đó ta có:

Bước 1: Tính s = ?

1	0	1
1	1	0
0	0	1

F

1	1	0
0	1	1
1	1	0

K

1	0	0
0	1	0
0	0	0

F ⊗ K

s = XSUM(F ⊗ K)

s = 0 nên s ≠ b

Bước 2: d = s ⊕ b

d = 0 ⊕ 1 = 1

Tìm một phần tử tại vị trí (u,v) mà $K_{u,v}$ = d

$F_{u,v} = 1 – F_{u,v}$

Cho G = F

Vậy ta có ma trận G là:

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

G

Để trích rút thông tin mật ta biến đổi ngược b = XSUM(G ⊗ K)

Thí dụ:

1	1	1
1	1	0
0	0	1

G

1	1	0
0	1	1
1	1	0

K

1	1	0
0	1	0
0	0	0

G ⊗ K

b = XSUM(G ⊗ K)

b = 1 Chính là bit tin được giấu ban đầu

Nhận xét 4: Với thuật toán Wu-Lee được cải tiến. Trong một ma trận F các bít tin ta chỉ thay đổi không quá 1 bít và cũng chỉ có thể che giấu 1 bít tin mật. Tuy nhiên, từ thuật toán Wu-Lee đã cải tiến này. Ta có thể mở rộng nó

để có thể giấu một chuỗi r bít vào một khối tin và đồng thời ta cũng chỉ cần thay đổi không quá 1 bít trong khối tin đó.

2.3.4. Thuật toán giấu một chuỗi bít trong một khối tin

Dựa trên việc cải tiến thuật toán Wu-Lee và các kỹ thuật giấu tin dựa trên biến đổi khối bít nhị phân. Ta sẽ tìm hiểu một thuật toán có khả năng gia tăng được tỷ lệ thông tin mật được giấu. Nhưng cũng chỉ cần biến đổi không quá 1 bít trong khối tin. Thuật toán này được trình bày tại bài báo "A Novel Data Hiding Scheme for Binary Images" của các tác giả Đỗ Văn Tuấn, Trần Đăng Hiền và Phạm Văn Ất [4].

Với thuật toán này ta có thể giấu r bít tin mật $b = (b_0, b_1, \ldots, b_r)$ trong một khối tin $F = (f_0, f_1, f_2, \ldots, f_{n-1})$ bằng cách sử dụng thêm một khối trọng số $P = (p_0, p_1, p_2, \ldots, p_{n-1})$.

Trong đó: $n = 2^r$.

F là dãy n các bít 0 hoặc 1

P là dãy hoán vị của dãy số (0, 1, 2, …, n-1).

Tương tự thuật toán Wu-Lee, thuật toán sau đây thay đổi ít nhất một phần tử của khối bít F để thỏa mãn điều kiện sau:

$$XSUM(F \otimes P) = b \qquad (2.6)$$

Phép biến đổi $XSUM(F \otimes P)$ được thực hiện như sau:

a. $F \otimes P = (c_0, c_1, c_2, \ldots, c_{n-1})$ mà $c_i = f_i \times p_i$

b. $XSUM(G \otimes P) = c_0 \oplus c_1 \oplus c_2 \oplus \ldots \oplus c_{n-1}$

2.3.4.1. Thuật toán 1:

Quá trình giấu tin:

Input: Khối tin làm phương tiện chứa F (có kích thước 2^r)

Chuỗi tin mật cần giấu b (có kích thước r)

Output: Khối tin F' chứa chuỗi tin mật b

Thuật toán:

Bước 1:

$$s = XSUM(F \otimes P) \oplus b \qquad (2.7)$$

Bước 2:

Xét các trường hợp của s:

2.1. Nếu $s = 0 \Rightarrow XSUM(F \otimes P) = b$: Không làm gì.

2.2. Nếu $s \neq 0$ tìm một phần tử (i) thỏa mãn $P_i = s$

Thay $f_i = 1 - f_i$

Khi đó ta thu được $XSUM(F \otimes P) = b$.

Chứng minh thuật toán 1:

Thật vậy: đặt $a = XSUM(F \otimes P)$, ta có $s = XSUM(F \otimes P) \oplus b = a \oplus b$.

Khi đó, $s \oplus s = (a \oplus b) \oplus s = 0$.

Từ đây suy ra $(a \oplus b) \oplus s = (a \oplus s) \oplus b = 0$ và do đó $a \oplus s = b$.

Giả sử $p_i = s$ và $a \oplus p_i = (c_0 \oplus c_1 \oplus ... \oplus c_{n-1}) \oplus p_i$.

Để ý rằng $c_i = f_i \times p_i$. Nếu $f_i = 1$ thì $c_i = p_i$ và khi đó $c_i \oplus p_i = p_i \oplus p_i = 0$, điều này tương đương với việc thay $f_i = 1$ thành $1 - f_i = 0$ trong F. Nếu $f_i = 0$ thì $c_i = 0$ và khi đó $c_i \oplus p_i = 0 \oplus p_i = p_i$, điều này tương đương với việc thay $f_i = 0$ thành $1 - f_i = 1$ trong F.

Thí dụ:

Để minh họa cho thuật toán 1, ta xét một ví dụ với chuỗi tin cần giấu là $b = b_1b_2b_3$ và khối tin F, P được giả sử như sau:

$b = b_1b_2b_3 = 110$

Các khối tin F, P:

F	0	0	1	1	1	0	1	0

P	0	1	2	3	4	5	6	7

Bước 1:

F	0	0	1	1	1	0	1	0
P	0	1	2	3	4	5	6	7
(F⊗ P)	0	0	2	3	4	0	6	0

- *XSUM(F ⊗ P)* = 010⊕011⊕100⊕110 = 011
- s = 011 ⊕ 110 = 101

Bước 2:

- $s \neq 0$
- Tìm (i) mà $P_i = s = 5$
- Thay thế F: $f_5 = 1 - 0 = 1$

Vậy sau khi dấu 3 bít 110 vào F, ta có F mới như sau:

F’	0	0	1	1	1	**1**	1	0

Nhận xét 5: Theo thuật toán vừa trình bày, cứ 2^r bít của khối tin làm phương tiện chứa thì ta giấu được r bít thông tin mật. Nếu càng chia nhỏ file âm thanh làm phương tiện chứa thì khối lượng thông tin mật giấu được càng nhiều. Tuy nhiên, như đã trình bày tại Chương 1 ta luôn phải cân nhắc giữa dung lượng và các chỉ tiêu khác như khả năng không bị phát hiện và tính bền vững.

Quá trình trích rút thông tin mật:

Input: Khối tin có chứa thông tin mật F’ (có kích thước 2^r)

Output: Chuỗi thông tin mật được giấu b (có kích thước r bít).

Thuật toán:

b = XSUM(F’ ⊗ P)

Thí dụ:

F’	0	0	1	1	1	1	1	0
P	0	1	2	3	4	5	6	7
(F’⊗ P)	0	0	2	3	4	5	6	0

XSUM(F’⊗ P) = 010⊕011⊕100⊕101⊕110 = 110

b = 110

2.3.4.2. Thuật toán 2:

Để có thể cải thiện mức độ an toàn của thuật toán 1, ta có thể sử dụng một khóa K = (k_0, k_1, k_2, …, k_{n-1}) với $k_i \in \{0, 1\}$ như là một yếu tố bảo mật trong quá trình giấu và trích rút thông tin mật. Ta có một thuật toán mới với nội dung tương tự thuật toán 1 đã được trình bày trên ngoại trừ điều kiện (2.6) thay vì sử dụng hệ thức XSUM(F⊗P) = b, ta sẽ sử dụng bởi hệ thức XSUM((F⊕K)⊗P) = b.

Phép biến đổi XSUM((F⊕K)⊗P) được thực hiện như sau:

a. V = F⊕K = (v_0, v_1, v_2,..., v_{n-1}) mà $v_i = f_i \oplus k_i$

b. V⊗P = (c_0, c_1, c_2,..., c_{n-1}) mà $c_i = v_i \times p_i$

c. XSUM((F⊕K)⊗P) = $c_0 \oplus c_1 \oplus c_2 ... \oplus c_{n-1}$

Khi đó ta có thuật toán như sau:

Quá trình giấu tin:

Thuật toán:

Bước 1:

$$s = XSUM((F \oplus K) \otimes P) \oplus b \qquad (2.8)$$

Bước 2:

Xét các trường hợp của s:

2.1. Nếu s = 0 ⇒ XSUM((F⊕K)⊗ P) = b: Không làm gì.

2.2. Nếu s ≠ 0 tìm một phần tử (i) thỏa mãn P_i = s

Thay $f_i = 1 – f_i$ (lật bít f_i)

Khi đó ta thu được XSUM((F⊕K)⊗ P) = b.

Chứng minh thuật toán 2

Thật vậy:

Đặt a = XSUM((F⊕K)⊗P), ta có s = XSUM((F⊕K)⊗P) ⊕ b = a ⊕ b.

Khi đó, s ⊕ s = (a ⊕ b) ⊕ s = 0.

Từ đây suy ra (a ⊕ b) ⊕ s = (a ⊕ s) ⊕ b = 0 và do đó a ⊕ s = b.

Giả sử p_i = s và a ⊕ p_i = (c_0 ⊕ c_1 ⊕... ⊕ $c_{n–1}$) ⊕ p_i.

Để ý rằng $c_i = (f_i \oplus k_i) \times p_i$.

Ta xét bốn tình huống sau:

f_i	k_i	$f_i \oplus k_i$	$c_i = (f_i \oplus k_i) \times p_i$	Tương đương
0	0	0	0	Lật f_i
0	1	1	p_i	Lật f_i
1	0	1	p_i	Lật f_i
1	1	0	0	Lật f_i

Thí dụ thuật toán 2:

Vẫn xét như thí dụ thuật toán 1 với chuỗi tin cần giấu là $b = b_1b_2b_3$ và khối tin F, P được giả sử như sau:

$b = b_1b_2b_3 = 110$

F	0	0	1	1	1	0	1	0

P	0	1	2	3	4	5	6	7

và khóa K là khối bít sau:

K	0	1	1	0	1	0	0	1

Bước 1:

F	0	0	1	1	1	0	1	0
K	0	1	1	0	1	0	0	1
V=(F⊗K)	0	1	0	1	0	0	1	1
P	0	1	2	3	4	5	6	7
V⊗P	0	1	0	3	0	0	6	7

- XSUM((F⊕K)⊗ P) = 001⊕011⊕110⊕111 = 011

Bước 2:

- s = XSUM((F⊕K)⊗ P) ⊕ b = 011 ⊕ 110 = 101 = 5
- Tìm (i) mà P_i = s = 5
- Thay thế F: f_3 = 1 – 1 =0

Vậy sau khi dấu 3 bít 110 vào F, ta có F mới như sau:

F’	0	0	1	1	1	**1**	1	0

Quá trình trích rút thông tin mật:

Thuật toán:

b = XSUM((F’⊕K)⊗ P)

Thí dụ:

F’	0	0	1	1	1	1	1	0

Là chuỗi tin của vật mang thông tin mật

khóa K là khối bít sau:

K	0	1	1	0	1	0	0	1

Thực hiện:

F	0	0	1	1	1	1	1	0
K	0	1	1	0	1	0	0	1
V=(F⊗K)	0	1	0	1	0	1	1	1
P	0	1	2	3	4	5	6	7
V⊗P	0	1	0	3	0	5	6	7

- XSUM((F⊕K)⊗ P) = 001⊕011⊕101⊕110⊕111 = 110
- b = XSUM((F⊕K)⊗ P)
- b = 110 là chuỗi thông tin mật được trích rút.

Đánh giá thuật toán:

Các kỹ thuật và thuật toán mà luận văn đã tìm hiểu như kỹ thuật LSB, kỹ thuật Parity-bit hay thuật toán Wu-Lee đều là các kỹ thuật và thuật toán đơn giản nhất cho việc giấu thông tin mật vào khối bít nhị phân. Tuy nhiên, các kỹ thuật và thuật toán này đều chỉ có thể giấu được 1 bít thông tin mật trong khối bít nhị phân. Đối với thuật toán giấu một chuỗi bít thì số bít thông tin mật được gia tăng đáng kể. Với việc sửa không quá 1 bít tin trong khối bít nhị phân. Thuật toán này có thể giấu được $\log_2(n)$ bít trong khối n bít.

Thí dụ: với khối bít là 256 ta có số bít tin mật được giấu là $\log_2(256) = 8$ bít (1 byte).

Mỗi một chương trình ẩn giấu dữ liệu thường sửa dụng khóa để bảo vệ dữ liệu được ẩn giấu. Việc sửa dụng chuỗi trọng số P và khóa K chính là yếu tố làm gia tăng mức độ khó thám mã cho các hacker.

Thí dụ: Đối với thuật toán 1 mới chỉ sử dụng chuỗi trọng số P được xác định là hoán vị của n phần tử của dãy n số nguyên liên tiếp. Tức là ta có n! cách chọn P. Giả sử ta sử dụng chuỗi trọng số P là 256 phần tử, tương ứng với 256! cách chọn. Đây là một giá trị khá lớn cho Hacker nếu muốn thám mã thuật toán này.

Vì vậy, Nếu cần gia tăng độ an toàn cho thuật toán. Ta có thể sử dụng thuật toán 2 với cả trọng số P và khóa K. Do vậy, thuật toán này nằm ở mức độ an toàn cao cho nhu cầu ẩn giấu thông tin mật trong quá trình truyền tải thông tin.

Tổng kết chương 2

Trong chương 2, luận văn đã trình bày khái quát về các phép biến đổi rời rạc từ mền không gian sang miền tần số. Tìm hiểu một số kỹ thuật giấu tin trong file âm thanh dựa trên việc biến đổi bít có trọng số thấp nhất trong một khối bít nhị phân. Trong các kỹ thuật, thuật toán đã tìm hiểu, luận văn tập trung vào thuật toán Wu-Lee đưa ra các nhận xét ưu điểm, nhược điểm và thực hiện việc cải tiến kỹ thuật này. Luận văn cũng đã tìm hiểu một số phép biến đổi rời rạc trên số nguyên. Từ đó tìm hiểu và triển khai một kỹ thuật giấu tin có khả năng giấu $\log_2(n)$ bít tin trong khối n bít của vật mang tin.

CHƯƠNG 3. TRIỂN KHAI CHƯƠNG TRÌNH THỬ NGHIỆM

3.1. Mục đích, yêu cầu

Mục đích của chương này trong luận văn là xây dựng một chương trình ứng dụng giúp cho người dùng có thể trao đổi thông tin bí mật bằng cách che giấu dữ liệu mật trong các file âm thanh trong quá trình truyền tải dữ liệu.

Chương trình đảm bảo được một số yêu cầu như sau:

- File âm thay sau khi đã giấu tin mật không bị thay đổi về kích thước và khó có thể nhận ra sự thay đổi bằng thính giác.

- Gia tăng được dung lượng tin mật được giấu so với một số kỹ thuật.

- Đảm bảo an toàn, bí mật cho hoạt động trao đổi thông tin.

3.2. Yêu cầu về cấu hình hệ thống

Chương trình không đòi hỏi nhiều về phần cứng của hệ thống nhưng do được xây dựng trên phần mềm Dev C^{++}, một môi trường phát triển tích hợp tự do (IDE - Integrated Development Environment) vì vậy hệ thống phần cứng chỉ cần đáp ứng tối thiểu cho công nghệ này như:

- CPU Pentum III

- Ram 256 MB trở lên

- Hệ điều hành Windows XP (tuy nhiên không nên sử dụng HĐH 64 bít).

3.3. Lựa chọn định dạng file âm thanh trong thực nghiệm

Giấu tin trong file âm thanh đã được một số hãng hầm mềm phát triển phục vụ cho việc trao đổi tin mật. Những phần mềm này có thể giấu tin trong nhiều định dạng file âm thanh khác nhau, có thể kể ra một số phần mềm như:

Tên phần mềm	Định dạng file chứa
OpenPuff	MP3, WAV
DarkCryptTC	WAV
MP3Stego	MP3

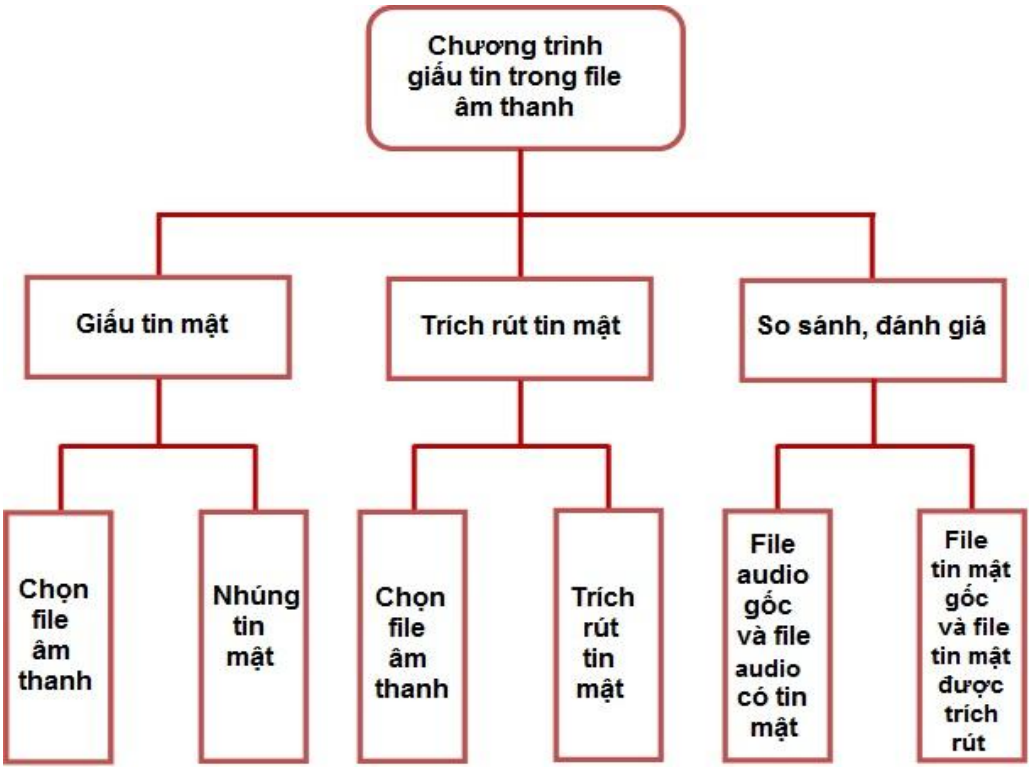
S-Tools	WAV

Bảng 3.1. Một số phần mềm giấu tin

Trong luận văn này sau khi nghiên cứu các kỹ thuật giấu tin mà cụ thể là các kỹ thuật như: mã hóa LSB (Least Significant Bít), mã hóa Parity (Parity Coding), Wu-Lee và việc cải tiến thuật toán Wu-Lee và từ đó xây dựng 2 thuật toán có khả năng tăng tỷ lệ tin mật được giấu. Trong quá trình tìm kiếm phương pháp giấu không làm thay đổi chất lượng âm thanh. Luận văn đã lựa chọn file âm thanh có định dạng WAV. Như đã nói ở Chương 1, đây được coi là một định dạng "không mất dữ liệu" - không bị nén - và là một file âm thanh PCM stereo

Trong chương trình thực nghiệm file âm thanh và tin mật được biến đổi thành chuỗi các bít, sau đó chuỗi bít tin mật được giấu và chuỗi bít file WAV.

3.4. Sơ đồ chương trình



Hình 3.1. Sơ đồ chương trình thử nghiệm

3.5. Thuật toán giấu tin và trích rút tin theo kỹ thuật đề xuất

Thuật toán được triển khai dựa theo kỹ thuật Mã hóa LSB (*Least Significant Bít*), thuật toán Wu-Lee cải tiến và thuật toán được trình bày tại bài báo "A Novel Data Hiding Scheme for Binary Images" của các tác giả Đỗ Văn Tuấn, Trần Đăng Hiền và Phạm Văn Ất [4] (*như đã đề cập ở chương 2*).

Như đã trình bày ở phần mở đầu và Chương 2. Thuật toán giấu tin mà luận văn lựa chọn là sửa dụng việc biến đổi bít ít quan trọng nhất của một khối tin đển thực hiện che giấu thông tin mật. Thuật toán được lựa chọn để đưa vào chương trình thực nghiệm là Thuật toán 1 trong thuật toán giấu một chuỗi bít. Với kích thước của khối tin mang tin mật là 256 tương ứng với 2^8. Với khối tin này ta sẽ giấu được 8 bít tin mật tương ứng 1 byte. Khối trọng số P được sử dụng tương ứng có 256 phần tử có giá trị từ 0 đến 255.

***Một số biến sử dụng trong thuật toán**:*

- W: File âm thanh định dạng WAV.
- M: File thông tin mật cần được che giấu.
- Sizeof(X): Kích thước file X tính bằng byte.
- BlockSize: Kích thước của các khối tin tính bằng byte (trong thuật toán sửa dụng các khối tin có kích thước 256).
- rate: Chiều sâu bít (AudioSampleSize - số byte trong một mẫu trích âm thanh)

  rate = 1 chiều sâu bít là 8 = 1 byte

  rate = 2 chiều sấu bít là 16 = 2 byte

3.5.1. Giấu tin

Trước khi thực hiện kỹ thuật giấu tin ta cần thực hiện một số bước tiền xử lý thông tin mật và phương tiện mang tin mật như sau:

- Chuyển file thông tin cần giấu sang dạng nhị phân bởi vì bởi thuật toán sẽ giấu từng chuỗi bit thông tin mật vào trong các khối tin của file âm thanh.

Quá trình giải tin là biến đổi ngược. Từ các khối tin của file âm thanh sẽ trích rút được các chuỗi bit tin mật được giấu để thu được file thông tin đã giấu.

- Xác định Header của file âm thanh. Sau đó đọc toàn bộ phần dữ liệu của file âm thanh vào một mảng một chiều để tiến hành thực hiện kỹ thuật giấu tin.

- Với mỗi một khối tin của file âm thanh ta sẽ thực hiện việc biến đổi một bít có trọng số thấp nhất trong khối để che giấu một byte thông tin mật.

Input:

- File âm thanh W theo định dạng WAV
- File chứa thông điệp mật cần giấu M

Output:

- File âm thanh W' chứa tin mật M

Thuật toán:

Bước 1:

Kiểm tra kích thước 2 file W và M có thỏa mãn công thức:

$$\frac{\text{Sizeof(W)}}{\text{rate} \times \text{BlockSize}} > Sizeof\,(M)$$

Nếu không thỏa mãn thì báo dừng, ngược lại tiến hành giấu.

Bước 2:

- Biến đổi W và M thành các chuỗi bít
- Xác định phần data của file W (phần sẽ giấu tin mật M)

Bước 3:

Lặp:

- Đọc từng byte của file M
- Thay đổi giá trị của (rate × BlockSize) trong file W (nếu có) để có thể giấu được 1 byte của file M

Bước 4:

- Ghi file W với một tên W'.

Trong đó: việc thực hiện thay đổi giá trị của (rate × BlockSize) được thực hiện như sau:

B1. Biến đổi (rate × BlockSize) thành khối tin F = $f_0f_1…f_{255}$ có kích thước 256 bít theo công thức nếu byte thứ (rate × i) với i = 0,1,…,255 có giá trị là một số nguyên chẵn thì $f_i = 0$ và ngược lại nếu byte thứ (rate × i) có giá trị là số nguyên lẻ thì $f_i = 1$

B2. Thực hiện Thuật toán 1 như đã trình bày tại Chương 2.

B3. Xác định giá trị bít f_i bị thay đổi và thay đổi giá trị của byte (rate × i) tương ứng tăng hoặc giảm 1 đơn vị.

3.5.2. Trích rút tin mật

Để thực hiện việc trích rút thông tin mật, ta cũng có một số thao tác tiền xử lý như sau:

- Chuyển file âm thanh có chứa tin mật về dạng các byte

- Xác định Header của file âm thanh chứa thông tin mật. Sau đó đọc toàn bộ phần dữ liệu của file âm thanh vào một mảng một chiều để tiến hành thực hiện kỹ thuật trích rút thông tin mật.

- Xác định kích thước khối byte của phần dữ liệu file âm thanh và thực hiện quá trình trích rút thông tin mật

Input: File W' có chứa tin mật M

Output: File tin mật M'

Thuật toán

Bước 1: Xác định phần data của file W'

Bước 2:

- Lặp: Đọc từng chuỗi (rate × BlockSize) byte và trích rút các byte tin mật ra khỏi file âm thanh W.
- Dừng khi gặp ký hiệu hết dữ liệu.

Bước 3: Ghi tin mật vào file M'

Trong đó: Việc đọc từng chuỗi (rate × BlockSize) byte để thực hiện việc trích rút tin mật được thực hiện như sau:

B1. Biến đổi (rate × BlockSize) thành khối tin F' = $f'_0 f'_1 \ldots f'_{255}$ có kích thước 256 bít theo công thức nếu byte thứ (rate × i) với i = 0,1,…,255 có giá trị là một số nguyên chẵn thì $f'_i = 0$ và ngược lại nếu byte thứ (rate × i) có giá trị là số nguyên lẻ thì $f'_i = 1$

B2. Thực hiện Thuật toán 1 (trích tin mật) đã trình bày tại Chương 2.

Nhận xét:

- Ngoài việc lựa chọn rate là chiều sâu bít của file âm thanh để xác định khối tin trong quá trình giấu và trích rút thông tin mật. Ta cũng có thể lựa chọn Số byte trong mẫu trích để xác định kích thước. Thông thường với file âm thanh định dạng WAV thì chiều sâu bít là 2 byte và số byte trong mẫu trích là 4 byte.

- Để đảm bảo chất lượng của file âm sau khi nhúng thông tin mật, cần lựa chọn chuỗi byte luôn ổn định về giá trị nghĩa là sự thay đổi giá trị giữa các byte liền kề trong khối tin là không quá lớn

3.5.3. Một số hàm và thủ tục giấu tin

Sau đây là một số hàm và thủ tục cơ bản mà chương trình thử nghiệm xây dựng phục vụ cho kỹ thuật giấu tin mà luận văn đề xuất:

3.5.3.1. Hàm tính XSUM

```
Byte Xsum(int id) {
 Byte t = 0;
  for (int i = 0; i < _BlockSize; ++i)
 t ^= i*_GetBít1(Bytedd(id,i), LSB);
 return t;
}
```

3.5.3.2. Hàm giấu một byte tin mật M

```
bool Hidc(Byte c, int id) {
 if (id >= FF.Fsize) return false;
 Byte t = Xsum(id);
  if (c != t) {
 t ^= c;
 _Inv1(Bytedd(id,t),LSB);
 }
 return true;
}
void Hid(const char *owav) {
 int b;
 b = FF.DataBeg;
 int i;
 Byte mask = 0xff;
 if (!Hidc((Byte)len & mask, b)) return;
 b += segment;
 if (!Hidc((len >> 8) & mask, b)) return;
 b += segment;
 for (i = 0; i < len; ++i) {
 if (!Hidc(msg[i], b)) break;
 b += segment;
 }
```

3.6. Kết quả thực nghiệm

File được chọn làm file chứa là **Sony.wav**

Một file nhạc chuông của hãng Sony

File tin mật cần giấu là file **ABBA.txt**

Chứa nội dung bài hát Happy New Year của ban nhạc ABBA

Quá trình thực hiện giấu tin

◆ *Thông tin File nguồn*

WAVE file: Sony.wav

File size: 2336326

ChunkID = RIEF len = 4

ChunkSize = 2336318

Subchunk1ID = fmt

Subchunk1Size = 18

Subchunk2ID = data

Subchunk2Size = 2336256

Data = 50

Giấu tối đa: 4562 (ký tự)

Bắt đầu từ byte: 570

◆ *Tên File kết quả và số byte tin mật đươc giấu*

Giấu file văn bản ABBA.txt trong file WAV Sony.wav

Tên file kết quả: Sonynew.wav

Tổng số byte cần giấu: 1529

Quá trình trich rút thông tin

◆ *Thông tin File kết quả*

Trích tin từ file WAVE Sonynew.wav

Tên file kết quả ABBANEW.txt

Thông tin file nguồn Sonynew.wav

WAVE file: Sonynew.wav

File size: 2336326

ChunkID = RIEF len = 4

ChunkSize = 2336318

Subchunk1ID = fmt

Subchunk1Size = 18

Subchunk2ID = data

Subchunk2Size = 2336256

Data = 50

Giấu tối đa: 4562 (ký tự)

Bắt đầu từ byte: 570

Số ký tự đã giấu 1529

◆ *So sánh file tin mật trước khi giấu và sau khi được trích rút*

So sanh 2 file

Size of ABBANEW.txt = 1529

Size of ABBA.txt = 1529

Số byte khác nhau: 0

◆ *So sánh file âm thanh trước và sau khi đã giấu tin mật*

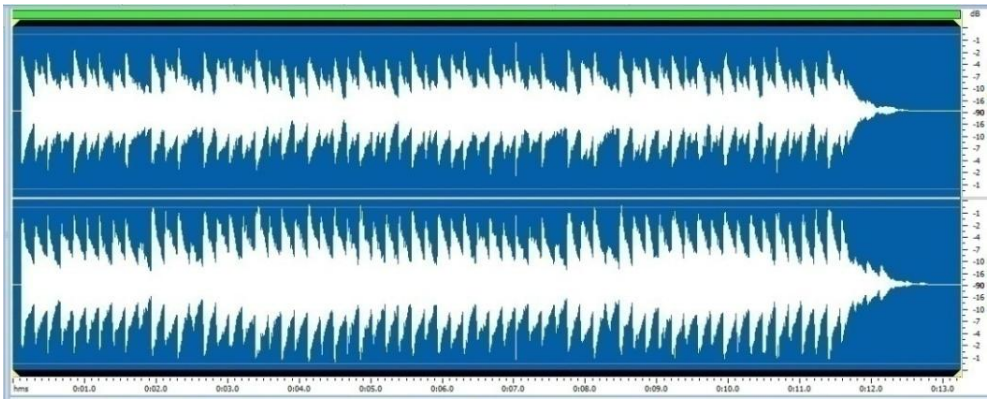
So sánh 2 file

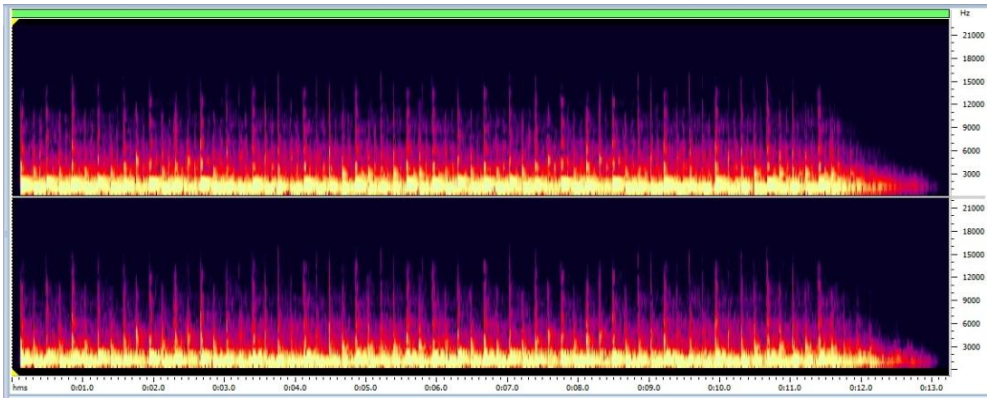
Size of Sonynew.wav = 2336326

Size of Sony.wav = 2336326

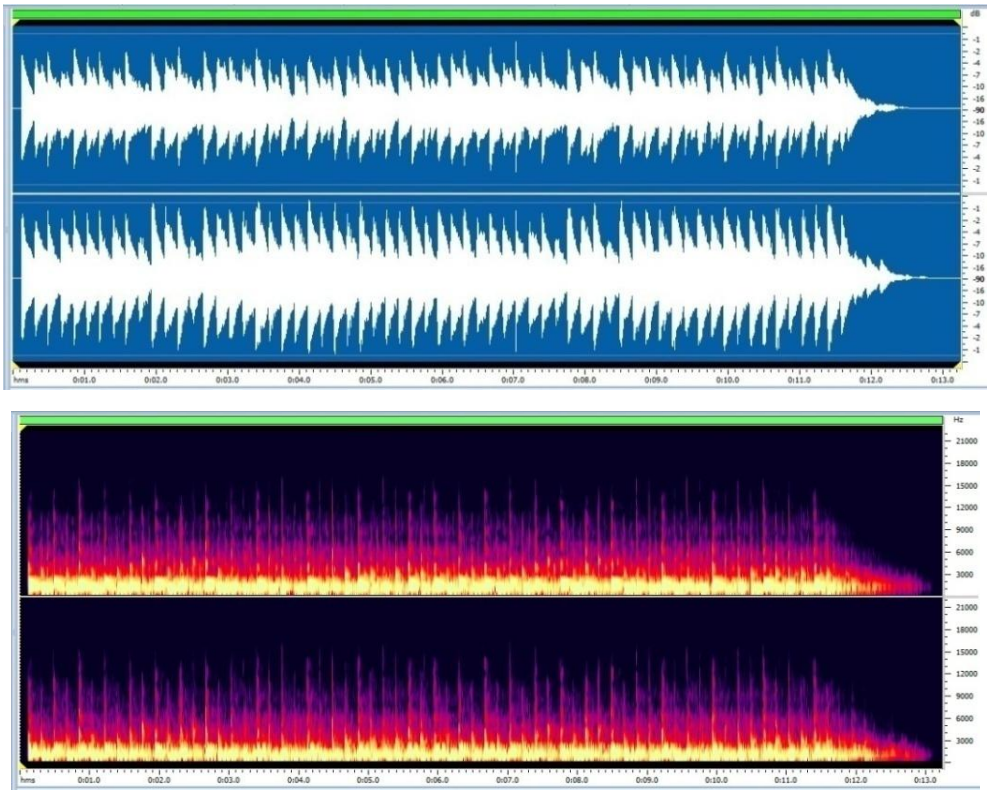
Số byte khác nhau: 1529

Hình ảnh dưới đây là phổ pha và phổ biên độ của file âm thanh gốc Sony.wav và file âm thanh sau khi đã được nhúng thông tin mật SonyNew.wav. Nếu chỉ bằng mắt thường ta khó có thể phát hiện được sự khác nhau giữa chúng.





Hình 3.2. Phổ biên độ và phổ pha của file chưa trước khi giấu tin



Hình 3.3. Phổ biên độ và phổ pha của file sau khi giấu tin

Dưới đây là hình ảnh trích một phần sự thay đổi các byte tin của file âm thanh sau khi nhúng tin mật (các byte tin bị thay đổi được in đậm). Trong đó giá trị thay đổi của 1 byte chỉ là tăng hoặc giảm 1 đơn vị.

Thí dụ: Trước khi giấu, byte tin có giá trị là 0 hay 248 thì sau khi giấu thông tin mật giá trị byte tin có thể thay đổi là 1 hay 247 tương ứng.

```
82 73 70 70 62 166 35 0 87 65 86 69 102 109 116 32 18 0 0 0 1
0 2 0 68 172 0 0 16 177 2 0 4 0 16 0 0 0 102 97 99 116 4 0 0
0 128 233 8 0 100 97 116 97 0 166 35 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 255 255 255 255 0 0 255 255 0 0 255
255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 255 255 0 0 255 255 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 255 255 0 0 0 0 255 255 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 255 255 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 255 255 0 0 255 255 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **0** 0 0 0 1
0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 255 255
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 255 255 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 255 255 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0
0 0 0 255 255 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 1
0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 255 255 255 255 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 255
255 0 0 255 255 0 0 0 0 0 0 255 255 0 0 255 255 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 255 255 0 0 0
0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255 255 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 255
255 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0

0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 0
0 0 0 0 0 0 0 255 255 0 0 255 255 0 0 0 0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 255 255 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 255 255 0 0 255 255 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0 0 255 255
0 0 1 0 0 0 1 0 0 0 255 255 0 0 255 255 0 0 0 0 0 0 0 0 0 0 0
0 255 255 0 0 255 255 0 0 255 255 255 255 255 255 255 255 0 0
0 0 255 255 0 0 255 255 255 255 255 255 255 255 255 0 0
255 255 0 0 255 255 255 255 255 255 255 255 255 0 0 255
255 255 255 0 0 255 255 0 0 0 0 0 0 0 255 255 0 0 254 255 0
0 255 255 0 0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255
255 0 0 0 0 1 0 0 0 1 0 255 255 255 255 255 255 255 255 255
255 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255
255 255 0 0 255 255 0 0 255 255 0 0 255 255 255 255 255 255
255 255 255 255 255 255 255 255 0 0 255 255 255 255 255 255
254 255 255 255 255 255 255 255 1 0 0 0 0 0 0 0 255 255 255
255 0 0 255 255 0 0 0 0 255 255 0 0 0 0 255 255 0 0 255 255
255 255 255 255 255 255 255 255 0 0 255 255 255 255 0 0 254
255 255 255 255 255 255 254 255 255 255 254 255 255 255 255 255
255 255 254 255 255 255 255 255 255 255 0 0 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 0 255 255 255 255
255 255 255 255 0 0 0 0 0 0 0 0 255 255 255 255 255 255 0 0
255 255 255 255 255 255 254 255 0 0 255 255 0 0 255 255 0 0
254 255 255 255 255 255 254 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 0 0 255 255 255
255 255 255 255 255 255 255 0 0 255 255 255 255 255 255 255
255 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 0 0 255
255 255 255 255 255 255 255 255 255 **1** 0 255 255 0 0 255 255
255 255 255 255 255 255 255 255 255 255 255 255 254 255 255
255 255 255 255 255 0 0 255 255 255 255 255 255 255 255 0 0 0
0 255 255 0 0 255 255 254 255 255 255 254 255 254 255 255 255
254 255 255 255 255 255 255 255 253 255 255 255 253 255 255
255 255 255 255 255 255 255 255 255 255 255 255 0 0 255 255 0 0
255 255 255 255 255 255 255 255 255 255 254 255 255 255 254
255 255 255 0 0 255 255 0 0 0 0 255 255 254 255 255 255 254
255 0 0 0 0 255 255 255 255 255 255 254 255 255 255 0 0 255
255 255 255 255 255 255 255 255 255 255 255 255 255 254 255 0
0 255 255 1 0 255 255 254 255 254 255 253 255 254 255 255 255
255 255 0 0 255 255 255 255 255 255 255 255 254 255 255 255
255 255 255 255 255 255 255 255 254 255 255 255 255 255 0 0
255 255 1 0 255 255 0 0 0 0 255 255 0 0 255 255 255 255 0 0
255 255 0 0 255 255 255 255 254 255 255 255 255 255 0 0 0 0 0
0 255 255 255 255 255 255 255 255 255 255 255 0 0 255 255 255 255

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 0 0 0 0 255 255 254 255 255 255 254 255 255 255 0 0 255
255 0 0 255 255 0 0 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 0 0 255 255 0
0 0 0 0 0 0 0 255 255 0 0 255 255 0 0 0 0 0 0 0 0 0 0 255 255
0 0 0 0 255 255 **1** 0 255 255 0 0 255 255 1 0 255 255 0 0 255
255 0 0 255 255 0 0 255 255 0 0 255 255 0 0 0 0 0 0 255 255 0
0 255 255 0 0 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 0 0 255 255 0 0 255
255 255 255 255 255 255 255 254 255 255 255 255 255 0 0 0 0 1
0 255 255 0 0 255 255 255 255 255 255 0 0 254 255 0 0 255 255
255 255 255 255 0 0 254 255 0 0 255 255 0 0 255 255 0 0 255
255 255 255 0 0 254 255 255 255 255 255 255 255 0 0 255 255
255 255 0 0 255 255 0 0 0 0 0 0 0 255 255 255 255 255 255 0
0 255 255 0 0 255 255 0 0 255 255 0 0 254 255 255 255 255 255
255 255 255 255 255 255 254 255 255 255 254 255 0 0 0 0 255
255 255 255 254 255 254 255 254 255 254 255 255 255 254 255
255 255 254 255 255 255 255 255 0 0 0 0 0 0 254 255 255 255
254 255 254 255 255 255 254 255 254 255 255 255 255 255 0 0
255 255 0 0 255 255 255 255 0 0 255 255 0 0 255 255 255 255 0
0 255 255 0 0 255 255 0 0 255 255 1 0 255 255 0 0 255 255 255
255 255 255 1 0 255 255 0 0 254 255 255 255 254 255 0 0 254
255 0 0 255 255 255 255 0 0 255 255 255 255 254 255 254 255
254 255 255 255 0 0 255 255 0 0 255 255 0 0 255 255 0 0 255
255 255 255 255 255 255 255 255 255 0 0 255 255 1 0 0 0 0 0 0
0 255 255 255 255 0 0 255 255 1 0 255 255 0 0 0 0 0 0 255 255
255 255 254 255 255 255 255 255 255 255 255 255 0 0 255 255 0
0 255 255 255 255 254 255 255 255 255 255 255 255 0 0 0 0 254
255 0 0 253 255 255 255 254 255 0 0 254 255 0 0 255 255 255
255 255 255 254 255 255 255 255 255 255 255 0 0 255 255 255
255 255 255 255 255 255 255 0 0 254 255 255 255 254 255 255
255 254 255 255 255 254 255 0 0 255 255 0 0 0 0 255 255 255
255 255 255 254 255 0 0 255 255 0 0 255 255 0 0 0 0 255 255
255 255 255 255 254 255 0 0 254 255 0 0 255 255 0 0 254 255 0
0 255 255 0 0 0 0 0 0 255 255 0 0 0 0 255 255 255 255 255 255
255 255 0 0 0 0 255 255 255 255 255 255 254 255 0 0 0 0 0 0
255 255 255 255 254 255 255 255 254 255 0 0 255 255 0 0 0 0 0
0 0 0 0 0 0 255 255 254 255 255 255 0 0 1 0 1 0 0 0 0 0 255 255
255 255 255 255 0 0 1 0 255 255 1 0 0 0 255 255 0 0 255 255 0
0 254 255 0 0 255 255 255 255 0 0 254 255 254 255 255 255 255
255 0 0 0 0 255 255 255 255 255 255 255 255 0 0 0 0 0 0 0 0 0
0 1 0 255 255 255 255 255 255 255 255 0 0 0 0 0 0 255 255 0 0
255 255 255 255 0 0 255 255 255 255 255 255 0 0 255 255 0 0
254 255 254 255 255 255 254 255 1 0 255 255 0 0 255 255 254
255 254 255 0 0 254 255 1 0 255 255 0 0 255 255 0 0 0 0 255
255 0 0 0 0 255 255 0 0 255 255 255 255 0 0 255 255 0 0 1 0 1
0 0 0 0 0 255 255 253 255 255 255 253 255 255 255 253 255 0 0
255 255 0 0 0 0 255 255 255 255 0 0 255 255 0 0 255 255 255

255 255 255 254 255 0 0 0 0 255 255 255 255 254 255 255 255
255 255 0 0 0 0 255 255 255 255 254 255 255 255 255 255 255
255 0 0 255 255 0 0 255 255 255 255 255 255 255 255 0 0 255
255 0 0 0 0 255 255 0 0 255 255 255 255 254 255 0 0 255 255 0
0 0 0 255 255 0 0 255 255 0 0 0 0 255 255 0 0 255 255 255 255
255 255 255 255 255 255 255 255 254 255 255 255 254 255 255
255 255 255 255 255 255 255 254 255 254 255 255 255 255 255
255 255 0 0 254 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 254 255 254 255 255 255 254 255 1 0 255 255 1
0 0 0 255 255 255 255 255 255 255 255 255 255 254 255 254 255
254 255 255 255 255 255 0 0 255 255 0 0 0 0 0 0 0 0 255 255
255 255 255 255 254 255 0 0 255 255 255 255 254 255 255 255
255 255 0 0 255 255 0 0 255 255 255 255 255 255 255 255 254
255 255 255 254 255 255 255 255 255 255 255 255 255 255 255 0
0 255 255 0 0 **1** 0 255 255 255 255 255 255 254 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 254 255
255 255 255 255 255 255 255 255 0 0 0 0 0 0 0 0 255 255 255
255 255 255 255 255 255 255 255 255 0 0 255 255 0 0 0 0 255
255 0 0 253 255 254 255 254 255 254 255 0 0 255 255 0 0 0 0
255 255 0 0 255 255 0 0 255 255 255 255 255 255 254 255 255
255 254 255 0 0 0 0 0 0 0 0 254 255 255 255 254 255 255 255
255 255 0 0 254 255 254 255 254 255 253 255 255 255 0 0 0 0 1
0 0 0 255 255 0 0 255 255 255 255 254 255 253 255 254 255 254
255 255 255 0 0 255 255 255 255 0 0 255 255 0 0 255 255 255
255 254 255 254 255 255 255 255 255 0 0 254 255 0 0 255 255 1
0 254 255 0 0 255 255 255 255 255 255 0 0 255 255 0 0 0 0 0 0
0 0 255 255 255 255 254 255 0 0 255 255 1 0 255 255 0 0 255
255 255 255 0 0 255 255 0 0 255 255 0 0 0 0 0 0 255 255 0 0
254 255 1 0 1 0 0 0 0 0 255 255 253 255 0 0 254 255 0 0 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
254 255 255 255 255 255 0 0 255 255 0 0 254 255 255 255 254
255 254 255 254 255 255 255 0 0 255 255 255 255 254 255 253
255 254 255 254 255 1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 2 0 1 0
255 255 255 255 254 255 253 255 0 0 253 255 0 0 255 255 0 0
255 255 255 255 0 0 0 0 1 0 1 0 0 0 255 255 255 255 0 0 0 0 2
0 1 0 0 0 0 0 255 255 254 255 1 0 255 255 1 0 0 0 0 0 255 255
0 0 255 255 0 0 0 0 1 0 255 255 255 255 255 255 254 255 254
255 255 255 255 255 255 255 255 255 255 254 255 254 255 255 255
255 255 255 255 254 255 254 255 252 255 254 255 253 255 0 0 0
0 0 0 0 0 255 255 255 255 0 0 0 0 1 0 1 0 0 0 255 255 255 255
255 255 0 0 0 0 0 0 255 255 255 255 254 255 255 255 254 255 1
0 255 255 1 0 0 0 0 0 255 255 1 0 0 0 255 255 0 0 253 255 253
255 255 255 255 255 1 0 2 0 1 0 1 0 1 0 0 0 3 0 1 0 3 0 2 0 1
0 1 0 0 0 255 255 0 0 255 255 0 0 255 255 1 0 0 0 2 0 2 0 1 0
1 0 254 255 254 255 255 255 254 255 2 0 1 0 1 0 255 255 253
255 253 255 253 255 254 255 254 255 253 255 253 255 253 255
255 255 254 255 2 0 1 0 1 0 1 0 255 255 253 255 255 255 253
255 255 255 255 255 254 255 253 255 255 255 254 255 2 0 1 0 2

0 0 0 255 255 254 255 0 0 0 0 1 0 0 0 255 255 254 255 254 255
253 255 0 0 255 255 1 0 0 0 254 255 253 255 252 255 252 255
253 255 254 255 254 255 254 255 0 0 254 255 3 0 2 0 3 0 3 0 0
0 0 0 0 0 0 0 4 0 2 0 2 0 1 0 **254** 255 0 0 255 255 1 0 255 255
255 255 253 255 252 255 253 255 252 255 255 255 255 255 0 0
255 255 0 0 253 255 0 0 254 255 0 0 255 255 1 0 0 0 2 0 1 0 4
0 4 0 4 0 5 0 2 0 3 0 2 0 2 0 1 0 0 0 254 255 255 255 1 0 0 0
4 0 1 0 4 0 1 0 2 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 255 255 1 0
254 255 0 0 253 255 254 255 252 255 253 255 254 255 254 255 1
0 255 255 1 0 254 255 0 0 254 255 1 0 0 0 1 0 255 255 0 0 254
255 0 0 255 255 1 0 1 0 0 0 0 0 255 255 254 255 254 255 253
255 252 255 253 255 253 255 252 255 255 255 254 255 2 0 2 0 2
0 2 0 254 255 253 255 252 255 252 255 254 255 255 255 255 255
0 0 254 255 254 255 0 0 255 255 2 0 1 0 0 0 255 255 255 255
253 255 0 0 255 255 2 0 2 0 0 0 1 0 255 255 254 255 0 0 254
255 255 255 254 255 253 255 253 255 253 255 252 255 254 255
253 255 0 0 0 0 3 0 3 0 6 0 4 0 2 0 1 0 254 255 254 255 253
255 254 255 0 0 1 0 1 0 2 0 0 0 2 0 2 0 2 0 4 0 3 0 5 0 3 0 3
0 2 0 1 0 0 0 2 0 1 0 3 0 2 0 3 0 2 0 3 0 2 0 4 0 2 0 3 0 2 0
254 255 255 255 250 255 250 255 251 255 251 255 255 255 255
255 1 0 0 0 254 255 254 255 250 255 250 255 249 255 249 255
254 255 253 255 2 0 1 0 255 255 255 255 252 255 253 255 0 0 0
0 5 0 4 0 3 0 3 0 255 255 255 255 255 255 255 255 1 0 1 0 1 0
255 255 253 255 252 255 251 255 249 255 250 255 247 255 251
255 248 255 254 255 252 255 0 0 255 255 1 0 1 0 3 0 2 0 1 0 1
0 255 255 0 0 0 0 1 0 4 0 4 0 6 0 6 0 3 0 3 0 0 0 0 0 0 0 255
255 1 0 255 255 2 0 0 0 0 0 0 0 254 255 253 255 255 255 253
255 3 0 2 0 4 0 3 0 1 0 0 0 0 0 255 255 3 0 1 0 4 0 3 0 4 0 3
0 4 0 2 0 255 255 255 255 251 255 251 255 255 255 253 255 3 0
3 0 0 0 1 0 250 255 251 255 251 255 252 255 0 0 0 0 0 0 255
255 255 255 254 255 1 0 1 0 0 0 1 0 253 255 254 255 255 255
255 255 0 0 255 255 0 0 255 255 255 255 255 255 254 255 255
255 254 255 0 0 0 0 0 0 0 0 254 255 255 255 254 255 255 255
255 255 0 0 254 255 254 255 254 255 253 255 255 255 0 0 0 0 1
0 0 0 255 255 0 0 255 255 255 255 254 255 253 255 254 255 254
255 255 255 0 0 255 255 255 255 0 0 255 255 0 0 255 255 255
255 254 255 254 255 255 255 255 255 0 0 254 255 0 0 255 255 1
0 254 255 0 0 255 255 255 255 255 255 0 0 255 255 0 0 0 0 0 0
0 0 255 255 255 255 254 255 0 0 255 255 1 0 255 255 0 0 255
255 255 255 0 0 255 255 0 0 255 255 0 0 0 0 0 0 255 255 0 0
254 255 1 0 1 0 0 0 0 0 255 255 253 255 0 0 254 255 0 0 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
254 255 255 255 255 255 0 0 255 255 0 0 254 255 255 255 254
255 254 255 254 255 255 255 0 0 255 255 255 255 254 255 253

Hình.3.4. Trích đoạn các byte của file Sony.wav sau khi nhúng tin mật

3.7. Đánh giá kết quả thực nghiệm

Như đã trình bày tại Chương 1. Thông thường các mẫu trích của một file âm thanh đã được lọc để loại bỏ những tần số không mong muốn (giữ lại tiếng nói từ 50 Hz đến 10 KHz, âm nhạc từ 20 Hz đến 20 kHz). Điều này đồng nghĩa với việc là tất cả các file âm thanh đều nằm trong ngưỡng nghe tốt nhất của tai người. Vì vậy, việc phát hiện ra nhiễu trong một file âm thanh là không quá khó. Việc sử dụng kỹ thuật giấu tin trong môi trường ảnh nhị phân sang môi trường file âm thanh cần có một số điều chỉnh sao cho phù hợp với môi trường mới. Giấu tin trong file âm thanh cần xác định việc lựa chọn giấu tin và trường độ hay cao độ của âm thanh hoặc xác định đoạn biên độ ổn định, ít thay đổi để tiến hành che giấu thông tin.

Trong quá trình thực nghiệm trên nhiều file âm thanh. Sau khi nghe các file âm thanh trước và sau khi che giấu thông tin mật. Nhận thấy rằng đối với những file âm thanh là âm nhạc thì việc xảy ra tín hiệu nhiễu ít hơn so với những file âm thanh có tiếng nói. Sau khi đọc chuỗi các byte của của 2 loại file âm thanh là file âm nhạc và file có tiếng nói. Chúng tôi nhận thấy

- Giá trị thay đổi giữa các byte của một chuỗi byte của file âm nhạc thấp hoặc ít hơn so với file âm thanh có tiếng nói.

- Đối với file âm thanh có tiếng nói, việc thay đổi giá trị của các byte có thể gây nhiễu rõ hơn đối với file âm thanh là âm nhạc.

Tuy nhiên, nếu tìm chuỗi byte ổn định rồi mới tiến hành che giấu thông tin sẽ gây phức tạp cho thuật toán và có thể rất khó thực hiện. Thí dụ: việc lựa chọn chuỗi byte có giá trị thay đổi thấp cũng đã là một bái toán khó.

Để khắc phục nhược điểm trên, ta chỉ nên che giấu thông tin mật vào các file âm thanh là âm nhạc.

3.8. Các khả năng ứng dụng

Trong thực tiễn thủ tục giấu tin có thể được vận dụng như một pha trong

các giao dịch truyền tin trên mạng hoặc trong quá trình truyền tải thông tin.

Trước hết ta xét nhiệm vụ sau đây:

Giả sử Hội đồng thi trung ương (TƯ) cần gửi một đề thi T đến các địa điểm thi, tạm gọi là Hội đồng thi cơ sở (CS). Khi đó, TƯ có thể lựa chọn một trong các sơ đồ sau:

SĐ1. TƯ gửi T đên CS theo con đường bảo mật về mặt vật lí.

Sơ đồ này đòi hỏi phải tổ chức các tuyến đường an toàn có đầy đủ lực lượng an ninh đến hàng trăm địa điểm cơ sở và các thủ tục giao nhận đề phiền hà, tốn kém. Không những thế yếu tố về mặt thời gian luôn là một thách thức trong quá trình vận chuyển.

SĐ2. Vận dụng hệ mật mã khóa công khai như sau:

2.1. Dùng hàm băm H để băm đề thi thành một bản đại diện T' có kích thước cố định, thí dụ, 128 bit: T' = H(T).

2.2. Chủ tịch Hội đồng thi TƯ dùng khóa mật K để kí vào bản băm T' để thu được bản kí T": T" = K(T').

2.3. TƯ mã hóa T và T" rồi gửi cho các hội đồng cơ sở CS.

2.4. Đến giờ G, TƯ gửi khóa giải mã cho các hội đồng CS để giải mã các bản T và T" và đối chiếu, xác nhận bản kí của Chủ tịch TƯ.

SĐ3. Tăng độ bảo mật bằng thủ tục giấu tin

Để nâng cao bảo mật, có thể bổ sung các pha giấu tin vào SĐ2 như sau:

Sau khi mã hóa tại bước 2.3 ta thực hiện

2.3.1. Nén T và T" thành một file F duy nhất có kích thước nhỏ.

2.3.2. Giấu F vào một audio M rồi gửi cho các CS.

Khi đó bước 2.4 sẽ được sửa lại như sau:

2.4. Đến giờ G:

2.4.1. TƯ gửi cho các cơ sở khóa K1 dùng để trich tin F từ audio M. sau đó giải nến để thu được T và T"

2.4.2. TƯ gửi tiếp cho các cơ sở khóa K2 để giải mã các bản T và T" và đối chiếu, xác nhận bản kí của Chủ tịch TƯ.

Với các file nguồn lớn chúng ta có thể cắt thành các file có kích thước nhỏ rồi giấu vào nhiều audio khác nhau. Thậm chí có thể chỉ dùng 1 audio nhiều lần, mỗi lần giấu 1 file theo qui trình sau:

Giả sử ta cần gửi F có kích thước lớn. Khi đó ta có thể cắt file F thành các file f1,…, fn và chỉ dùng một audio M. Quá trình giấu thực hiện như sau:

Giấu f1 vào M để thu được M1;

Giấu f2 vào M để thu được M2;

…

Giấu fn vào M để thu được Mn.

Quá trình trích rút thông tin lại được thực hiện tương tự:

Trích rút từ M1 để thu được f1;

Trích rút từ M2 để thu được f2;

…

Trích rút từ Mn để thu được fn.

Kết nối các file f1, f2,…, fn ta có được F ban đầu.

Việc kết hợp giữa các kỹ giấu tin và mã hóa sẽ càng làm tăng mức độ an toàn cho thông tin mật trong quá trình truyền tải thông tin

Tổng kết chương:

Trong chương này đã trình bày các yêu cầu của một chương trình giấu tin. Lựa chọn định dạng file âm thanh mà luận văn sử dụng để cài đặt chương trình thực nghiệm. Mô tả cấu trúc, thuật toán chương trình cài đặt thử nghiệm giấu tin mật trong một file âm thanh và đưa ra các kết quả đánh giá về chương trình cũng như các kết quả cụ thể mà sau thử nghiệm đã thu được. Đồng thời cũng đưa ra một trong các khả năng ứng dụng mà chương trình đem lại trong việc trao đổi thông tin mật trong quá trình truyền tải thông tin.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Từ vài thập niên gần đây, với những tác động mạnh mẽ của các tiến bộ khoa học và công nghệ, đặc biệt của công nghệ thông tin và truyền thông, thế giới đang biến chuyển tới một nền kinh tế và xã hội mới mà thông tin và tri thức được xem là nguồn lực chủ yếu. Nhờ vào sự hỗ trợ của các thiết bị khoa học hiện đại, việc trao đổi thông tin của con người càng trở nên thuận tiện và dễ dàng hơn. Việc trao đổi thông tin liên lạc trên Internet ngày càng trở lên phổ biến. Do có hạ tầng tốt, tính tiện lợi và phổ dụng, Internet được các tổ chức, các cá nhân tích cực sử dụng trong việc trao việc liên lạc, trao đổi thông tin. Nhưng vấn đề đặt ra là nguy cơ là thông tin, bị đánh cắp, bị xuyên tạc,… ngày càng gia tăng, đòi hỏi phải có cơ chế bảo mật, bảo đảm an ninh, an toàn cho thông tin trao đổi trên mạng.

Bảo mật thông tin, trong đó có mật mã học và giấu tin mật đang là những lĩnh vực đang được quan tâm nghiên cứu. Nhiều giải thuật mã hoá, nhiều thuật toán giấu tin mật đã được đề xuất nhằm tăng cường an ninh thông tin.

Trên cơ sở nghiên cứu các kỹ thuật, thuật toán giấu tin đã được các nhà khoa học triển khai luận văn đã đưa ra một kỹ thuật giấu tin mới bằng cách sử dụng các biến đổi rời rạc là thay đổi bít có trọng số thấp nhất trong một byte dữ liệu nguồn để tiến hành giấu thông tin mật.

Trong quá trình nghiên cứu, thử nghiệm, có thể đưa ra một số kết luận sau:

1. Dữ liệu âm thanh có nhiều không gian hơn để giấu tin mật so với hình ảnh. Tuy nhiên tai người lại nhạy cảm với nhiễu do vậy cần điều chỉnh mật độ và lựa chọn các byte phù hợp để giấu tin.

2. Việc lựa chọn file âm thanh như thế nào để có thể làm giảm thiểu đến mức thấp nhất của việc sinh ra nhiễu trong quá trình giấu tin.

3. Muốn tăng cường mức độ bền vững của thông tin mật được giấu cần

phải kết hợp với các phương pháp mã hóa thông tin.

Ưu điểm và hạn chế của luận văn

Luận văn đã trình bày một hệ thống các kiến thức tổng quan về giấu tin, âm thanh và âm thanh số. Một số các kỹ thuật giấu tin và thuật toán giấu đã được triển khai. Trên cơ sở hệ thống các kiến thức, luận văn đã xây dựng một kỹ thuật và thuật toán giấu tin phù hợp với yêu cầu đặt ra. Luận văn cũng đã trình bày một chương trình giấu tin dựa trên thuật toán và đưa ra kết quả thử nghiệm cụ thể.

Hiện tại luận văn mới chỉ nghiên cứu việc giấu tin trên file âm thanh dạng WAV, trong khi dữ liệu âm thanh trên Internet phần lớn là âm thanh dạng Mp3, Mp4,..

Luận văn cũng chưa thực hiện được việc điều chỉnh mật độ và lựa chọn các byte phù hợp để che giấu thông tin mật, tránh việc sinh nhiễu trong quá trình giấu tin.

Hướng phát triển tiếp theo của luận văn:

- Tiếp tục nghiên cứu sâu hơn về giấu tin bằng các phép biến đổi rời rạc
- Thực hiện việc giấu tin trên các file âm thanh dạng khác mà hiện đang được phổ biến trên Internet như FLAC, Mp3, OGG,…
- Kết hợp giấu tin với các phương pháp mã hóa thông tin.
- Phát triển thuật toán trong các môi trường khác như Image, Video,…

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Nguyễn Xuân Huy, Huỳnh Bá Diệu (2008). "Nghiên cứu kỹ thuật giấu tin trong audio hỗ trợ xác thực". *Tạp chí Khoa học Đại Học Quốc Gia Hà Nội*, Khoa học Tự nhiên và Công nghệ 25, p69-p74.

[2] Nguyễn Xuân Huy, Huỳnh Bá Diệu, Võ Thị Thanh (2013). "Một cải tiến cho kỹ thuật giấu LSB trên dữ liệu audio". *Tạp chí Khoa học Đại Học Quốc Gia Hà Nội*, Khoa học Tự nhiên và Công nghệ 29, số 2.

[3] Nguyễn Quốc Trung (1998). *Xử lý tín hiệu số*, NXB Khoa học kỹ thuật.

Tiếng Anh

[4] Do Van Tuan, Tran Dang Hien, Pham Van At. "A Novel Data Hiding Scheme for Binary Images". *International Journal of Computer Science and Information Security*, Vol. 10, No. 8, August 2012.

[5] Huỳnh Bá Diệu, Nguyễn Xuân Huy. "An Improved Technique for Hiding Data in Audio". *The Fourth International Conference on Digital Information and Communication Technology and its Applications*, University of the Thai Chamber of Commerce Bangkok Thailand, May 6-8, 2014.

[6] Min Wu (2001), *Multimedia Data Hiding*, Princeton University.

[7] Saraju P. Mohanty (1999), *Digital Watermarking: A tutorial Review*, University of South Florida, USA.

Internet

[8] http://en.wikipedia.org/wiki