Tạp chí Khoa học ĐHQGHN, Khoa học Tự nhiên và Công nghệ 25 (2009) 69-74

Nghiên cứu kỹ thuật giấu tin trong audio hỗ trợ xác thực

Nguyễn Xuân Huy[1], Huỳnh Bá Diệu [2,*]

[1]*Viện Công nghệ thông tin, Viện Khoa học Công nghệ Việt Nam, 18 Hoàng Quốc Việt, Hà Nội, Việt Nam*
[2]*Khoa Công nghệ thông tin, Trường Đại học Công nghệ, ĐHQGHN, 144 Xuân Thuỷ, Hà Nội, Việt Nam*

Nhận ngày 26 tháng 12 năm 2008

Tóm tắt. Bài báo này trình bày một kỹ thuật giấu tin mật trong dữ liệu audio. Do dữ liệu chứa tin giấu được truyền trên các kênh công khai nên có thể bị một số tấn công dẫn đến thay đổi dữ liệu chứa. Kết quả là khi giải tin, ta có thể nhận lại tin giấu bị sai. Điểm mới của bài báo này là đề xuất áp dụng các kỹ thuật mã hóa nhằm mục đích phát hiện và sửa lỗi trên tin giấu sau khi giải tin.

Từ khóa: Dữ liệu chứa, mã Hamming.

1. Giới thiệu

Bảo mật dữ liệu là vấn đề đang được quan tâm hiện nay. Có hai khuynh hướng chính đang được nghiên cứu và triển khai là mã hóa và giấu thông tin. Mã hóa dữ liệu sẽ thực hiện việc biến đổi bản tin gốc M thành bản mã và gửi cho bên nhận. Bên nhận sẽ thực hiện việc giải mã bản mã để lấy lại bản tin gốc. Kỹ thuật giấu tin tiếp cận theo hướng khác, sẽ giấu tin vào các "khe hở" của dữ liệu chứa C. Dữ liệu chứa có thể là dữ liệu audio, ảnh hay video. Khe hở của dữ liệu được hiểu là khoảng biến thiên giá trị của dữ liệu có củng ảnh hưởng đến hệ thống tri giác của con người [1]. Sau khi giấu tin xong, dữ liệu chứa tin giấu C' sẽ được truyền đi cho bên nhận và bên nhận sẽ giải tin để lấy lại tin giấu. Trong qua trình truyền, đối tượng C' có thể chịu một số tấn công làm cho nội dung C' bị thay đổi. Vì vậy khi đến người nhận, thay vì nhận C' để giải tin họ nhận được C" có nội dung có thể sai khác với C'. Điều này dẫn đến việc có thể tin giấu khi nhận M' có thể bị sai khác với tin giấu M ban đầu và người nhận có thể không biết có sự sai này và vẫn sử dụng tin sai M'. Trong một số ứng dụng, việc sai sót này trong một mức nào đó là có thể cho phép nhưng đa số ứng dụng thì việc sai này là không chấp nhận và có thể gây ra hậu quả nghiêm trọng. Kỹ thuật giấu tin được đưa ra dưới đây là một cải tiến giúp cho người nhận tin hạn chế thấp nhất khả năng nhận tin sai, có thể biết được tin khi nhận M' có bị sai khác với bản tin gốc M không và đưa ra bản M'' là bản sửa lỗi của M'. Đối tượng dữ liệu chứa được chọn là các dữ liệu audio.

2. Một số kỹ thuật giấu tin trong audio

Các kỹ thuật giấu tin trong audio dựa vào hệ thống thính giác của con người [1-3]. Việc giấu tin trong audio thường là khó hơn trong các dữ liệu media khác do hệ thống thính giác của con người khá nhạy với các nhiễu. Sau đây là một số phương pháp giấu.

* Tác giả liên hệ. ĐT.: 84-511-3827111(201).
E-mail: dieuhb@gmail.com

2.1. Mã hóa LSB (Least Significant Bit)

Phương pháp mã hóa LSB là cách đơn giản nhất để nhúng thông tin vào trong dữ liệu audio. Phương pháp này sẽ thay thế bít ít quan trọng nhất (thường là bít cuối) của mỗi mẫu dữ liệu bằng bít thông tin giấu. Ví dụ mẫu 8 bít như sau:

Sau khi giấu bít 1 sẽ như sau:

Hình 1. Minh họa kỹ thuật giấu LSB.

Ưu điểm của phương pháp này là dễ cài đặt và cho phép giấu dữ liệu nhiều. Có thể tăng thêm dữ liệu giấu bằng cách dùng hai bít LSB. Tuy nhiên cách này làm cũng làm tăng nhiễu trên đối tượng chứa dẫn đến đối phương dễ phát hiện và thực hiện các tấn công. Vì vậy dữ liệu chứa cần phải được chọn trước khi giấu sử dụng phương pháp mã hóa LSB.

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

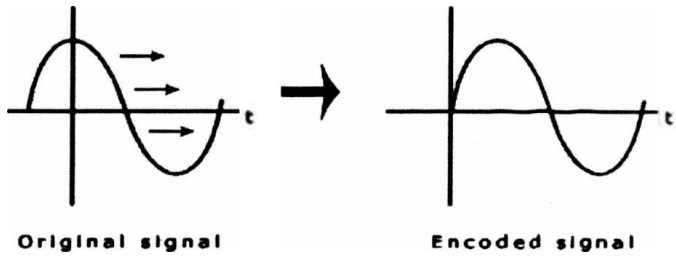
2.2. Mã hóa Parity (Parity Coding)

Thay vì chia dữ liệu thành các mẫu riêng lẻ, phương pháp mã hóa chẵn lẻ chia dữ liệu thành các nhóm mẫu và giấu từng bit thông tin vào trong các nhóm mẫu này. Nếu parity bít của nhóm mẫu này không trùng với bit thông tin giấu thì ta tiến hành điều chỉnh một bít nào đó trong nhóm mẫu này. Phương pháp này cho ta nhiều sự lựa chọn hơn khi thay đổi 1 bít và có vẻ "kín đáo" hơn so với phương pháp điều chỉnh LSB.

Cả hai phương pháp LSB và Parity đều có những hạn chế. Do tai người khá nhạy nên những thay đổi trên dữ liệu chứa sẽ sinh nhiễu và người nghe rất dễ nhận ra. Một điểm nữa là hai phương pháp này không bền vững và thông tin sẽ bị mất sau khi thực hiện việc lấy mẫu lại. Một trong những cách khắc phục là thực hiện việc giấu nhiều lần. Tuy nhiên cách này cũng có hạn chế là nó làm tăng thời gian xử lý.

2.3 Mã hóa Phase (Phase Coding)

Phương pháp mã hóa pha giải quyết được các hạn chế do sinh ra nhiễu của hai phương pháp giấu dữ liệu trên. Phương pháp mã hóa pha dựa vào tính chất là các thành phần của pha không gây ảnh hưởng đến hệ thống thính giác của con người như nhiễu. Việc giấu tin được thực hiện bằng cách điều chỉnh pha trong phổ pha của dữ liệu số[3].



Dữ liệu gốc Dữ liệu sau khi giấu

Hình 2. Kỹ thuật mã hóa pha.

Quá trình mã hóa pha được chia thành các bước sau:

a. Dữ liệu âm thanh gốc được chia thành các segment nhỏ hơn có dài bằng chiều dài chiều dài bằng với thông tin cần giấu.

b. Thực hiện biến đổi Fourier rời rạc DFT trên mỗi đoạn

c. Tính độ lệch pha giữa các đoạn kề nhau.

d. Giá trị chính xác các pha của các đoạn có thể thay đổi nhưng mối liên hệ về sự khác nhau về pha 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:

$$phase\_new = \begin{cases} \pi/2 & if \ \ message \ \ bit = 0 \\ -\pi/2 & if \ \ message \ \ bit = 1 \end{cases} \quad (1)$$

e. Ghép các segment lại và tiến hành DFT ngược để tạo lại dữ liệu âm thanh.

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

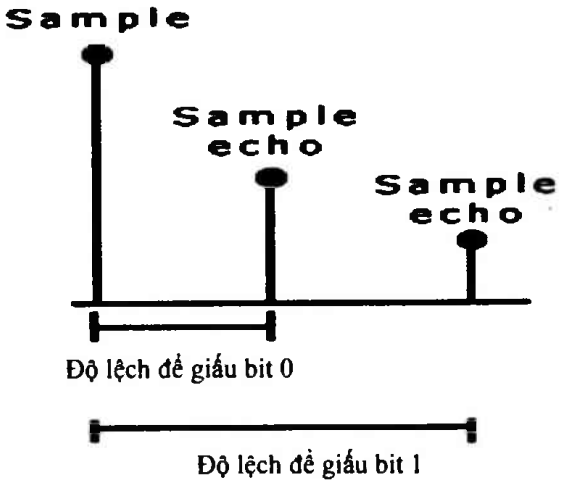
Một yếu điểm của phương pháp này là tỉ lệ dữ liệu thấp do thông tin chỉ được giấu vào segment đầu tiên. Có thể cải thiện bằng cách tăng độ dài segment. Tuy nhiên cách này sẽ làm cho tin giấu dễ phát hiện. Phương pháp mã hóa pha chỉ thích hợp cho việc giấu lượng nhỏ thông tin.

2.4. Kỹ thuật trải phổ

Thông thường các file audio được truyền qua các kênh truyền thông, các kênh truyền thông này sẽ tập trung dữ liệu audio trong vùng hẹp của phổ tần số để duy trì năng lượng và tiết kiệm băng thông. Các kỹ thuật trải phổ cố gắng trải thông tin mật vào trong phổ tần số của dữ liệu audio càng nhiều càng tốt. Nó cũng tương tự như kỹ thuật LSB là trải ngẫu nhiên thông tin giấu trên toàn bộ file audio. Lợi điểm của phương pháp trải phổ là nó bền vững trước một số tấn công. Tuy nhiên nó cũng có hạn chế là sinh nhiễu và dễ nhận ra. Hai phương pháp trải phổ sử dụng trong giấu tin audio là DSSS (Direct Sequency Spread Spectrum) và FHSS (Frenquency Hopped Spread Spectrum).

2.5. Kỹ thuật giấu dựa vào tiếng vang (Echo)

Kỹ thuật giấu dựa vào tiếng vang thực hiện giấu tin bằng cách thêm vào tiếng vang trong tín hiệu gốc. Dữ liệu nhúng được giấu bằng cách thay đổi 3 tham số của tiếng vang : Biên độ ban đầu, tỉ lệ phân rã và độ trễ. Khi thời gian giữa tín hiệu gốc và tiếng vang giảm xuống, hai tín hiệu có thể trộn lẫn và người nghe khó có thể phân biệt giữa hai tín hiệu. Số lượng tin giấu có liên quan đến thời gian trễ của tiếng vang và biên độ của nó[3].



Hình 3. Kỹ thuật giấu điều chỉnh echo.

2.6. 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 audio. Để giấu nhiều hơn một bit, tín hiệu gốc được chia thành các đoạn ngắn hơn và mỗi đoạn sau đó có thể được tạo tiếng vang để giấu số bit mong muốn. Dữ liệu chứa cuối cùng bao gồm các đoạn được mã độc lập nối lại theo thứ tự chia ban đầu. Kỹ thuật giấu tin dựa vào tiếng vang rất hiệu quả trong các file audio chất lượng cao. Các file âm thanh chưa làm giảm chất lượng và không có quá nhiều đoạn yên lặng thường dùng kỹ thuật này để giấu tin.

Một cách tiếp cận khác là tiến hành mã hóa chuỗi bit theo một cách nào đó giúp ta phát hiện ra lỗi. Thay vì giấu trực tiếp L bit vào đối tượng chứa, ta biến đổi chuỗi bit bằng cách bổ sung một số bit vào S nhằm mục đích kiểm tra lỗi.

3. Các tấn công trên các hệ giấu tin

Dữ liệu chứa sau khi được nhúng tin C' 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 loại số tấn công[5].

3.1. 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 bit có thể nhiều hoặc ít hơn so với trong file gốc.

3.2. Lọc thông

Phương pháp này chỉ chọn lại tần số của dữ liệu thỏa 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.

3.3. Thêm nhiễu

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

3.4. Biến đổi D/A A/D

Tấn công này thực hiện bằng cách biến đổi C' từ dạng số sang dạng analog sau đó thực hiện biến đổi từ analog sang lại số, và kết quả là được C'' có thể khác C'.

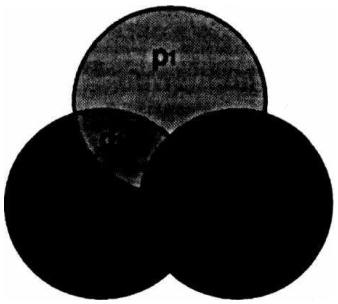
Ngoài ra còn các tấn công khác như nén, lượng tử hóa v.v.

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. Kỹ thuật đề xuất trong bài báo này là sử dụng mã Hamming để mã hóa tin giấu trước khi nhúng vào trong dữ liệu chứa.

4. Kỹ thuật đề xuất và các kết quả đạt được

4.1 Mã Hamming hỗ trợ xác thực

Mã Hamming được công bố năm 1950. Nguyên lý của mã Hamming bắt nguồn từ việc khai triển và mở rộng quan điểm chẵn lẻ . Với mỗi nhóm 4 bit dữ liệu, mã Hamming thêm 3 bit kiểm tra. Thuật toán (7,4) của Hamming có thể sửa chữa bất cứ một bit lỗi nào, và phát hiện tất cả lỗi của 1 bit, và các lỗi của 2 bit gây ra [6,7]. Điều này có nghĩa là đối với tất cả các phương tiện truyền thông không có chùm lỗi đột phát (burst errors) xảy ra, mã (7,4) của Hamming rất có hiệu quả (trừ phi phương tiện truyền thông có độ nhiễu rất cao thì nó mới có thể gây cho 2 bit trong số 7 bit truyền bị đảo lộn).



Hình 4. Sơ đồ bit dữ liệu và bít kiểm tra của mã Hamming.

Mã Hamming sử dụng hai ma trận, gọi là ma trận sinh G và ma trận kiểm tra H. Đối với mã (7,4) ma trận G và H sẽ như sau:

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{H} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Chuỗi bit thông tin u sẽ được mã hóa bằng cách nhân với ma trận sinh G, kết quả được từ mã v. Từ mã v được tính theo công thức:

$$v = u*G \quad (2)$$

Nếu nhân v với H sẽ được 0. Đây là công thức để kiểm tra lỗi:

$$v*H = 0 \quad (3)$$

Ví dụ chuỗi bít thông tin giấu là u=1011 sẽ được mã hóa thành từ mã là v=0110011.

Để kiểm tra chuỗi bít r có độ dài n nhận được có bị sai hay không, ta tiến hành nhân r với H. Nếu kết quả khác 0 thì r bị sai.

Ta có thể viết r= v+e trong đó e là vector lỗi. Nếu e chỉ gồm có một bit lỗi thì mã Hamming có thể sửa được. Nếu e gồm 2 lỗi thì mã Hamming chỉ phát hiện được lỗi chứ không sửa được.

Mã Hamming (7,4) có thể mở rộng sang (8,4) bằng cách thêm một dòng bit chẵn lẻ ở dòng đầu tiên của ma trận G và thêm bít chẵn lẽ vào dòng cuối cùng của ma trận H.

4.2. Quá trình giấu tin

Bước 1: Mã hóa

Chuỗi bit thông tin được chia thành các đoạn có độ dài 4, tiến hành mã hóa cho từng đoạn này ta thu được chuỗi M'

Bước 2: Giấu tin

Đọc header, trích phần dữ liệu audio của dữ liệu chứa C, tùy thuộc vào kỹ thuật giấu tin được chọn, có thể thực hiện các biến đổi từ miền thời gian sang miền tần số hoặc thực hiện giấu tin trực tiếp trên miền thời gian.

Trong quá trình giấu có thể có sử dụng các khóa mật.

Ghi lại dữ liệu sau khi đã thực hiện giấu tin, ta được C'.

4.3. Quá trình giải tin xác thực tin giấu

Bước 1: Trích thông tin

Dựa vào khóa k, kỹ thuật giấu và số bit giấu đã được biết trước, ta tiến hành trích chuỗi bit từ dữ liệu C'', kết quả ta thu được M'

Bước 2: Xác thực

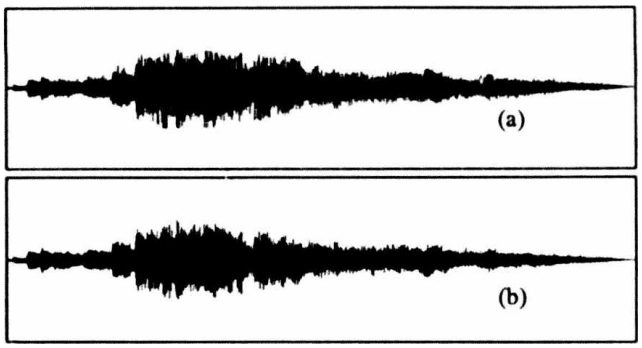
Chia M' thành các đoạn có độ dài 7, tiến hành nhân từng đoạn này với H.

Trường hợp 1: Nếu như toàn bộ kết quả nhân các đoạn của M' với H đều cho kết quả 0 thì kết luận là không có tấn công trên C' và thực hiện trích các bit dữ liệu của M' để tạo tin giấu.

Trường hợp 2: Nếu như trong quá trình nhân các đoạn của M' với H có đoạn cho kết quả khác 0 thì kết luận là có tấn công hoặc do nhiễu và ghi nhận cách sửa lỗi. Trong trường hợp này cũng rút trích dữ liệu từ M' sau khi thực hiện sửa lỗi và tạo lại tin giấu (chưa chắc chắn đúng, có thể đề nghị gửi lại).

Các kết quả thử nghiệm dưới đây sử dụng phần mềm WavePad để thực hiện các tấn công. Dữ liệu âm thanh là file WindowsLogOn.wav. Chuỗi thông tin giấu là "AAA".

Hình 5. Dữ liệu gốc trước khi giấu tin.



Hình 6. Dữ liệu sau khi giấu tin (a) và thực hiện tấn công lọc thông cao (b).

5. Kết luận

Kỹ thuật giấu tin đề xuất ở trên đã giải quyết được một phần vấn đề xác thực tin giấu trong kỹ thuật giấu tin. Các kết quả thử nghiệm cho thấy hầu hết các tấn công làm sai lệch tin

giấu có thể phát hiện ra nhưng khả năng sửa lỗi thấp. Có thể cải tiến bằng cách sử dụng các kỹ thuật mã hóa khác có khả năng phát hiện và sửa lỗi cao hơn.

Tài liệu tham khảo

[1] Min Wu, *Multimedia Data Hiding*, Princeton University, USA, 2001.

[2] Michael Arnold, Dr. Christoph Busch, *Watermarking of Audio, Music Scores and 3D Models*, INI-GraphicsNet - Press & Media, 2003.

[3] Chun-Shien Lu, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*, IDEA Group Publishing, 2005.

[4] Nedeljko Cvejic, Tapio Seppänen, Fusing Digital Audio Watermarking And Authentication In Diverse Signal Domains, *EUSIPCO Proceedings*, 2005.

[5] Min Wu, Scott A, Craver, Edward W Felten, Bede Liu, *Analysis Of Attacks On Sdmi Audiowatermarks*, Princeton University, USA, 2003.

[6] Shi-Cheng Liu, Shinfeng d. Lin, BCH Code-Based Robust Audio Watermarking in the Cepstrum Domain, *Journal of Information Sience and Engineering* 22 (2006) 535.

[7] http://en.wikipedia.org/wiki/Hamming_code.

An approach of hiding data in audio support authentication

Nguyen Xuan Huy[1], Huynh Ba Dieu[2]

[1]*Institute of Infomation Technology, Vietnamese Academy of Science and Technology, 18 Hoang Quoc Viet, Hanoi, Vietnam*

[2]*Faculty of Information Technology, College of Technology, VNU, 144 Xuan Thuy, Hanoi, Vietnam*

This article presents an approach relate to hiding data in audio support authentication. Because the host data is transmitted on the public channels should have been a number of the attacks, as a result, we may receive incorrect information in extract phase. Our proposal lies on the encryption technology aimed at detecting and correct errors.

Keywords: Host data, Hamming code.