PHƯƠNG PHÁP CẢI TIẾN KỸ THUẬT CHE GIẤU THÔNG TIN TRONG CÁC TÀI LIỆU THẺ DỰA TRÊN KỸ THUẬT STEGANOGRAPHY

Nguyễn Thị Hương Giang
Trường Đại học Sư phạm, Đại học Huế
Nguyễn Xuân Linh
Trung tâm Công nghệ Thông tin tỉnh Thừa Thiên Huế

TÓM TẮT

Che giấu thông tin là công nghệ nhúng các thông tin bí mật vào những dữ liệu ngụy trang và làm cho các thông tin bí mật này trở thành "vô hình". Ngày nay, các tài liệu thẻ như HTML, XML, XHTML và WML được biết đến như là định dạng chuẩn để lưu trữ các dữ liệu có cấu trúc cũng như để trình diễn dữ liệu trên các trình duyệt web. Chúng là những ngôn ngữ cơ sở cho việc trao đổi thông tin trên mạng Internet. Khác với các phương pháp che giấu thông tin trên dữ liệu hình ảnh hoặc âm thanh, hiện nay chỉ có một số ít phương pháp che giấu thông tin vào dữ liệu văn bản, đặc biệt trên các tài liệu thẻ. Hơn nữa, một trong những hạn chế của những phương pháp này là dễ dàng bị phát hiện nếu đối tượng tấn công biết được phương pháp được sử dụng để che giấu thông tin (stego-key). Trong bài báo này, chúng tôi đề xuất phương pháp cải tiến để nâng cao tính năng bảo mật của các phương pháp truyền thống thông qua việc sử dụng khái niệm khóa động (dynamic stego-key) để che giấu thông tin trong các tài liệu thẻ.

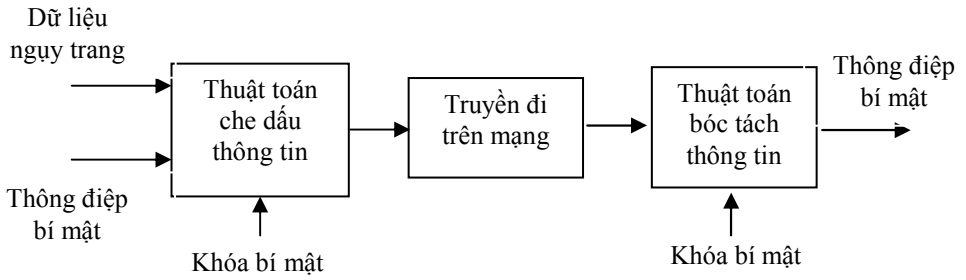
1. Giới thiệu

Steganography hay giấu dữ liệu trong dữ liệu được bắt nguồn từ thuật ngữ Hy Lạp *stegos*, có nghĩa bao phủ hoặc che giấu và *graphia* – nghĩa là viết, vừa là nghệ thuật vừa là ngành khoa học để che giấu thông tin bên trong thông tin. Với tốc độ phát triển nhanh chóng của các công nghệ dữ liệu trên Internet, lượng thông tin dữ liệu dưới dạng điện tử được truyền và nhận trên mạng đang tăng lên không ngừng. Khi công nghệ truyền nhận thông tin trên mạng cần phải được bảo mật thì tầm quan trọng của việc che giấu thông tin được nhìn nhận một cách rộng rãi hơn.

Bằng việc sử dụng các dữ liệu "ngụy trang" (*cover data*) để che giấu, các thông tin bí mật ở bên trong nó có thể truyền đi an toàn trên mạng mà không hề gây ra một sự nghi ngờ hay bất thường nào về mặt thông tin, thông tin bí mật có thể được trích xuất sau đó khi cần [3], [4], [5]. Hình 1 dưới đây sẽ mô tả phương pháp che giấu thông tin tổng quát [4].

Với những loại dữ liệu khác nhau, chúng ta cần những phương pháp che giấu

thông tin khác nhau. Ví dụ để giấu thông tin bí mật trên dữ liệu hình ảnh, chúng ta có thể sử dụng các bit không quan trọng của các điểm ảnh trên bức ảnh đó để che giấu thông tin. Sau khi giấu các thông tin này, sự thay đổi hình dạng và màu sắc của bức ảnh khó có thể nhận thấy bằng mắt thường. Đối với dữ liệu văn bản, sự thay đổi vị trí của các dấu chấm câu hoặc khoảng trắng giữa các từ là một trong các cách đơn giản để thêm các giá trị thông tin mà không làm thay đổi nội dung của văn bản gốc. Tương tự như vậy, các tài liệu thẻ có thể được sử dụng để che giấu các thông tin bằng cách thay đổi cấu trúc của các tài liệu đó mà không ảnh hưởng tới nội dung hiển thị của nó trên trình duyệt Web.



Hình 1. *Sơ đồ che giấu thông tin tổng quát [4]*

2. Các phương pháp che giấu thông tin trên tài liệu thẻ

Mặc dù có nhiều phương pháp để giấu thông tin trên tài liệu thẻ nhưng trong phạm vi bài báo này chúng tôi chỉ giới thiệu hai phương pháp là "Chèn khoảng trắng trong thẻ" và "Thay đổi thứ tự của các thuộc tính trong thẻ" vì hai phương pháp này cho phép chúng ta tận dụng được tất cả các thẻ trong tài liệu để che giấu thông tin (lượng thông tin có thể được che giấu là lớn nhất so với các phương pháp khác). Đồng thời chúng tôi đề xuất phương pháp cải tiến để nâng cao tính bảo mật với ý tưởng sử dụng khóa động thay vì khóa tĩnh như trong phương pháp truyền thống.

2.1. Phương pháp truyền thống

a. Chèn khoảng trắng trong thẻ

Phương pháp "Chèn khoảng trắng trong thẻ" là một trong những phương pháp hiệu quả nhất được sử dụng để che giấu thông tin trên các tài liệu thẻ vì chúng ta có thể khai thác và sử dụng tất cả các thẻ có trong tài liệu. Theo W3C, một thẻ có thể chứa nhiều khoảng trắng hoặc không có khoảng trắng nào trước ký hiệu đóng của thẻ. Bằng việc thêm vào hoặc xóa đi các khoảng trắng này, chúng ta có thể nhúng các dữ liệu vào mà vẫn bảo đảm giữ nguyên ý nghĩa của nội dung thẻ gốc [4]. Chẳng hạn, chúng ta có thể định nghĩa một quy tắc theo ví dụ sau:

Ví dụ 1: Văn bản XML [4]

Khóa tĩnh:

<tag>, </tag> or <tag attribute=value>: Không có khoảng trắng nào trong thẻ trước khi đóng thẻ được kí hiệu là đại diện cho bit 0

<tag >, </tag >, or <tag attribute=value >: Có một khoảng trắng trước khi đóng thẻ được kí hiệu là đại diện cho bit 1

Dữ liệu ngụy trang trước khi nhúng thông tin

<user><name>Peter</name><id>01</id></user>

<user><name>Mary</name><id>02</id></user>

Dữ liệu ngụy trang sau khi được nhúng thông tin:

<user ><name>Peter</name ><id >01</id></user>

<user><name >Mary</name><id>02</id ></user >

Rõ ràng chúng ta thấy dữ liệu ngụy trang trước và sau khi nhúng thông tin sẽ hiển thị hoàn toàn giống nhau trên trình duyệt Web.

If (có một khoảng trắng trước ký hiệu đóng thẻ) {

Bit "1" đã được mã hóa trong thẻ

}

Else {

Bit "0" đã được mã hóa trong thẻ

}

Trong ví dụ trên, dữ liệu đã được nhúng là: 101100 010011

Ví dụ 2: Văn bản HTML [4]

Qui tắc giấu thông tin sử dụng khóa tĩnh (*static stego-key*):

<tag>, </tag> or <tag attribute=value>: Không có khoảng trắng nào trong thẻ trước khi đóng thẻ được kí hiệu là đại diện cho bit 0

<tag >, </tag >, or <tag attribute=value >: Có một khoảng trắng trước khi đóng thẻ được kí hiệu là đại diện cho bit 1

Giả sử chúng ta muốn nhúng ký tự A (A = 01000001)

Dữ liệu ngụy trang sau khi đã giấu thông tin (*stego data*):

<html xmlns="http://www.w3.org/1999/xhtml">..0

<head ><title>Microsoft Corporation</title>…100

<meta http-equiv= "X-UA-Compatible">...0

<meta http-equiv="Content-Type" charset=UTF-8">..0

</meta>0

</html >…1

Trong ví dụ trên, dữ liệu đã được nhúng là: 01000001 (A)

Đối với phương pháp truyền thống này, một trong những hạn chế lớn nhất đó là việc sử dụng khóa tĩnh. Đó là định dạng thẻ <tag> hoặc </tag> luôn luôn đại diện cho một giá trị (bit 0 hoặc bit 1) trong dữ liệu kết quả sau khi đã chứa thông tin bí mật (*stego data*). Chính vì vậy, đối tượng tấn công có thể tìm thấy thông tin bí mật được che giấu nếu biết được phương pháp che giấu.

b. Thay đổi thứ tự xuất hiện của các thuộc tính trong thẻ

Phương pháp "Thay đổi thứ tự xuất hiện của các thuộc tính trong thẻ" cũng được xem là một phương pháp hiệu quả để che giấu các thông điệp bí mật trong các tài liệu thẻ. Mặc dù chúng ta không thể chèn thêm bất cứ dữ liệu gì vào tài liệu thẻ vì chúng hoặc sẽ hiển thị trên trình duyệt hoặc sẽ được nhìn thấy trong mã nguồn; nhưng các thông điệp bí mật có thể được giấu trong các tài liệu XML, HTML, XHTML v.v.. thông qua việc thay đổi thứ tự xuất hiện của các thuộc tính bên trong thẻ mà không làm thay đổi sự hiển thị nội dung của thẻ trong trình duyệt cũng như gia tăng kích thước của tài liệu [4], [6].

Trong phương pháp này, thứ tự xuất hiện của các thuộc tính sẽ được quy ước như là một khóa (*stego-key*) cho việc che giấu và trích xuất thông tin. Quy ước này là cố định áp dụng cho toàn bộ tài liệu, do đó còn gọi là khóa tĩnh (*static stego-key*). Chúng ta có thể định nghĩa nhiều cặp thuộc tính để tăng khối lượng thông tin có thể che giấu. Ví dụ sau sẽ mô tả rõ hơn về phương pháp này.

Ví dụ 3: Đoạn mã XML [4]

Định nghĩa khóa thông qua cặp thứ tự:

<event month="MONTH" date="DATE">EVENT</event> đại diện bit 0 (khi thuộc tính "month" đứng trước thuộc tính "date")

<event date="DATE" month="MONTH">EVENT</event> đại diện bit 1 (khi thuộc tính "month" đứng sau thuộc tính "date")

Dữ liệu sau khi được giấu thông tin sẽ có dạng:

<event month="JUL" date="4">Independence day</event>

<event date="24" month="DEC">Christmas</event>

Chuỗi bit thông điệp đã được che giấu trong đoạn dữ liệu trên là: 01

Ví dụ 4: Đoạn mã HTML [4]

What is the difference?

What is the difference?

Chúng ta có thể định nghĩa thứ tự các cặp thuộc tính từ ví dụ trên như sau:

Bảng 1. *Khóa tĩnh cho phương pháp "Thay đổi thứ tự xuất hiện của các thuộc tính trong thẻ"*

Thuộc tính thứ nhất	Thuộc tính thứ hai
Month	Date
Class	Style
Cellpadding	Cellspacing
Align	Valign
Width	Height

If ("Thuộc tính thứ nhất" đứng trước "Thuộc tính thứ hai")

{

Bit "0" đã được mã hóa

}

Else {

Bit "1" đã được mã hóa

}

Dữ liệu đã được che giấu là: 01

2.2 Phương pháp cải tiến

Trong các phương pháp cải tiến này, chúng tôi sử dụng khái niệm khóa động để nhúng các thông tin cần che giấu thay vì sử dụng khóa tĩnh (biểu diễn bit 0 và bit 1 bằng các ký hiệu cố định) để tăng độ an toàn cho các thông tin bí mật được gửi đi. Khóa động là kết quả của thuật toán XOR kết hợp giữa khóa tĩnh truyền thống và một khóa phụ bí mật, trong đó khóa phụ là do người gửi và người nhận tự qui ước. Trong phương pháp sử dụng khóa động, các bit 0 và bit 1 được biểu diễn bằng các ký hiệu không cố định, tùy thuộc vào chuỗi mật khẩu bí mật và tên của mỗi thẻ.

a. Khoảng trắng trong thẻ

Chúng ta có thể định nghĩa một khóa động dựa vào sự kết hợp giữa một khóa

định trước và tên thẻ trong mỗi thẻ. Bảng sau đây mô tả khóa động cho phương pháp "Khoảng trắng trong thẻ".

Bảng 2. *Khóa động cho phương pháp "Khoảng trắng trong thẻ"*

Dynamic Stego – Key	**Nếu ((N mod 2) = 1)**	**Nếu ((N mod 2) = 0)**
<tên thẻ "thuộc tính"="giá trị"> or </tên thẻ>	Biểu diễn bit 1	Biểu diễn bit 0
<tên thẻ thuộc tính="giá trị" > or </tên thẻ >	Biểu diễn bit 0	Biểu diễn bit 1

Bước 1: Xác định một khóa bí mật từ người sử dụng

Bước 2: Với mỗi thẻ trong tài liệu thẻ

Bước 2.1: Trích xuất tên của mỗi thẻ, ví dụ html, title, meta, head, v.v

Bước 2.2: Tách chuỗi ký tự khóa bí mật và tên thẻ thành các ký tự riêng lẻ.

Ví dụ: khóa bí mật pswd sẽ được tách thành

p: 01110000 (mã ASCII tương ứng) s: 01110011 w: 01110111 d: 01100100

tên thẻ: title

t: 01110100 i: 01101001 t: 01110100 l: 01101100 e:01100101

Bước 2.3: Thực hiện phép toán XOR giữa khóa bí mật và tên thẻ

t i t l e

XOR

p s w d p (lặp lại các ký tự của khóa bí mật nhỏ hơn độ dài của tên thẻ)

Tương ứng với:

01110100 01101001 01110100 01101100 01100101

XOR

01110000 01110011 01110111 01100100 01110000

00000100 00011010 00000011 00001000 00010101

Gọi N là số lượng bit "1" trong chuỗi bit kết quả của phép toán XOR, trong trường hợp này, ta có N= 10

Bước 2.4: Nhúng thông tin bí mật

If (Có khoảng trắng trước khi đóng thẻ)

{

If ((N mod 2) = 1)

Bit "0" đã được mã hóa trong thẻ

Else

Bit "1"-đã được mã hóa trong thẻ

}

Else

{

If ((N mod 2) = 1)

Bit "1"-đã được mã hóa trong thẻ

Else

Bit "0" đã được mã hóa trong thẻ

}

Bước 3: Kết thúc thủ tục

Ví dụ 5: Đoạn mã HTML với khóa động [4]

Dữ liệu dùng để giấu thông tin:

<html xmlns="http://www.w3.org/1999/xhtml">

<head><title>Microsoft Corporation</title>

<meta http-equiv= "X-UA-Compatible">

<meta http-equiv="Content-Type" charset=UTF-8">

</meta>

</html>

Giả sử, chúng ta cần giấu một ký tự A vào dữ liệu trên để truyền đi trên mạng. (A = 01000001, mã ASCII)

Mật khẩu định nghĩa trước: pswd

Tên thẻ: html, head, title, /title, meta, /meta, /html

html XOR pswd sẽ cho chúng ta kết quả:

01101000 01110100 01101101 01101100

01110000 01110011 01110111 01100100

00011000 00000111 00011010 00001000 (có chín bit “1” trong chuỗi bit kết quả)

Do đó N = 9

Với các tên thẻ còn lại chúng ta có:

head XOR pswd sẽ cho kết quả N = 8

title XOR pswdp sẽ cho kết quả N = 10

/title XOR pswdps sẽ cho kết quả N = 19

meta XOR pswd sẽ cho kết quả N = 11

/meta XOR pswdp sẽ cho kết quả N = 15

/html XOR pswdp sẽ cho kết quả N = 17

Dữ liệu sau khi che giấu thông tin sẽ là:

<html xmlns="http://www.w3.org/1999/xhtml" >…0

<head ><title>Microsoft Corporation</title >…100

<meta http-equiv= "X-UA-Compatible" >...0

<meta http-equiv="Content-Type" charset=UTF-8" >...0

</meta >0

</html>…1

Chuỗi bit đã được che giấu: 01000001

Trong ví dụ 3 này, dữ liệu sau khi che giấu ký tự A (01000001) khác với ví dụ hai (ở phương pháp truyền thống). Dữ liệu sau khi che giấu thông tin ở trong ví dụ 3 cũng sẽ được thay đổi khi mật khẩu thay đổi và khoảng trắng trong thẻ sẽ đại diện bit 0 hoặc bit 1 (không cố định) tùy thuộc vào tên thẻ và mật khẩu. Tại nơi nhận dữ liệu, cần có mật khẩu để trích xuất thông tin bí mật.

b. Thay đổi thứ tự xuất hiện của các thuộc tính trong thẻ

Trong phương pháp này, ý tưởng sử dụng khóa động thay vì phương pháp truyền thống cũng được đề xuất để áp dụng tương tự như phương pháp đã đề cập trước trong bài báo này. Bảng dưới đây mô tả khóa động cho phương pháp "Thay đổi thứ tự xuất hiện của các thuộc tính trong thẻ".

Bảng 3. *Khóa động cho phương pháp "Thay đổi thứ tự xuất hiện của các thuộc tính trong thẻ"*

Thuộc tính thứ nhất	**Thuộc tính thứ hai**	**If ((N mod 2) = 1)**	**If ((N mod 2) = 0)**
Cellpadding	Cellspacing	Biểu diễn bit 1	Biểu diễn bit 0
Cellspacing	Cellpadding	Biểu diễn bit 0	Biểu diễn bit 1

Align	Valign	Biểu diễn bit 1	Biểu diễn bit 0
Valign	Align	Biểu diễn bit 0	Biểu diễn bit 1
Month	Date	Biểu diễn bit 1	Biểu diễn bit 0
Date	Month	Biểu diễn bit 0	Biểu diễn bit 1

Gọi N là số bit "1" trong chuỗi kết quả của thuật toán XOR giữa mật khẩu và tên thẻ chúng ta có:

If ("Thuộc tính thứ nhất" trước "Thuộc tính thứ hai")

{

If ((N mod 2) = 1)

Bit "1" được mã hóa bởi thẻ

Else

Bit "0" được mã hóa bởi thẻ

}

Else {

If ((N mod 2) = 1))

Bit "0" được mã hóa bởi thẻ

Else

Bit "1" được mã hóa bởi thẻ

}

Ví dụ 6: Dữ liệu ngụy trang HTML với phương pháp khóa động

Giả sử, chúng ta cần nhúng ký tự B = 01000010

Khóa bí mật (*stego-key*): pass

Dữ liệu ngụy trang (*cover data*):

<table border="0" cellpadding ="0"cellspacing ="0">

<tr>

<td align="left" valign="bottom">

<img src=" http://www.microsoft.com/ssdis100.gif" alt="click" align="right" valign="top"

border="0" width="100" height="100"></td>

<td align="top“ valign="left" >

</td></tr>

<tr align="center" valign="middle">

<td align="center" valign="middle" nowrap="nowrap">

<td align="center" valign="middle" nowrap="nowrap">

<td align="center" valign="middle" nowrap="nowrap">

ssd news</td> <td></td>

</tr>

</table>

Tên thẻ: table, tr, td, img

table XOR passp cho chúng ta kết quả:

01110100 01100001 01100010 01101100 01101101

01110000 01100001 01110011 01110011 01110000

00000100 00000000 00010001 00011111 00011101 (có 12 bit “1” trong chuỗi kết quả)

Do đó N = 12

Với các tên thẻ còn lại chúng ta có:

td XOR pa sẽ cho kết quả N = 3

img XOR pas sẽ cho kết quả N = 7

tr XOR pa sẽ cho kết quả N = 4

Thông tin bí mật cần che giấu: B = 01000010

<table border="0" cellpadding ="0“cellspacing ="0">…0

<tr>

<td align="left" valign="bottom">…1

<img src=" http://www.microsoft.com/ssdis100.gif" alt="click" valign=“right“ align=“top” …0

border="0" width="100" height="100"></td>…0

<td valign="top“align="left" > …0

</td></tr>

<tr align="center" valign="middle"> ….0

<td align="center" valign="middle" nowrap="nowrap">...1

<td valign="center" align="middle" nowrap="nowrap">…0

<td valign="center" align="middle" nowrap="nowrap">

ssd news</td> <td></td>

</tr>

</table>

Dữ liệu đã được giấu đi với khóa động: 01000010

3. Kết luận

Che giấu thông tin trong các tài liệu thẻ là một kỹ thuật khá phức tạp so với các loại dữ liệu “ngụy trang” khác vì cấu trúc của tài liệu thẻ cực kỳ khó để thêm hay xóa đi các dữ liệu từ tài liệu gốc mà không làm ảnh hưởng tới ý nghĩa và hiển thị của tài liệu trên trình duyệt. Tuy nhiên, bằng cách sử dụng các đặc tính của tài liệu thẻ, chúng ta có thể thay đổi cấu trúc của tài liệu để che giấu dữ liệu bằng nhiều phương pháp. Trong bài báo này, chúng tôi đã đề xuất phương pháp dùng khóa động nhằm tăng cường tính bảo mật của phương pháp truyền thống. Nhờ đó, đối tượng tấn công khó có thể phát hiện và đánh cắp các thông tin được che giấu.

Trong tương lai, các phương pháp này có thể kết hợp với các kỹ thuật mã hóa như DES, 3DES, RSA để tạo ra sự “bảo mật kép” cho việc che giấu thông tin [2], [8], [9], [10]. Đối tượng tấn công có thể phát hiện khóa hoặc có thể biết kỹ thuật che giấu nhưng khó có thể tìm ra hai thông tin trên cùng một lúc để giải mã ra các thông tin bí mật đã được che giấu. Điều này gây khó khăn hơn rất nhiều cho đối tượng tấn công, và gia tăng tính bảo mật cho việc trao đổi thông tin.

TÀI LIỆU THAM KHẢO

1. W. Bender et al. *Techniques for Data Hiding,* IBM Systems Journal, 1996
2. Neil F. Johnson, Sushil Jajodia. *Steganalysis: The Detection of Hidden Information*. Proceedings of the IEEE Information Technology Conference, Sep 1998
3. R.Anderson, F.Petitcolas: *On the limits of the steganography*, IEEE Journal Selected Areas in Communications, VoL .16, No. 4, May 1998.

4. Shingo Inoue, Kyoko Makino, Ichiro Murase, Osamu Takizawa, Tsutomu Matsumoto, Hiroshi Nakagawa. *A proposal on Information Hiding Method using XML*, Mitshubishi Research Institute, Inc, Yokohama National University, 2001.
5. Donovan Artz. *Digital Steganography: Hiding Data within Data*, IEEE, 2001.
6. M. M Amin, M. Salleh, S. Ibrahim, M.R.Katmin and M.Z.I. *Shamsuddin: Information Hiding using Steganograph*, National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia , IEEE 2003.
7. Mohamed Lahcen BenSaad, Sun XingMing. *Techniques with Statistics for Web Page Watermarking*, 2005.
8. Aasma Ghani Memon, Sumbul Khawaja, Asadullah Shah. *Steganography: A New Horizon for Safe Communication through XML*, 2005
9. Marc Smeets, Matthijs Koot Covert Channels - Research Report. 2006
10. Shirali-Shahreza, *Advanced Communication Technology*, 2008. ICACT 2008. 10th International Conference on Volume 3. Text Steganography by Changing Words Spelling, 2008.

IMPROVEMENT METHOD FOR INFORMATION HIDING IN TAGGED DOCUMENT BASED ON STEGANOGRAPHY APPROACH

Nguyen Thi Huong Giang
College of Pedagogy, Hue University
Nguyen Xuan Linh
Thua Thien Hue Center of Information Technology

SUMMARY

Information hiding is a technique to embed secret information into a covered data in the way that keeps the secret information invisible. Nowadays, tagged documents such as HTML, XML, XHTML and WML are known as the universal format for preserving structured documents and data as well as presenting data on web browsers. They are used as basic languages for exchanging information on Web. As compared to the information hiding methods intended for images and sounds, there are few methods for hiding information into text, especially on tagged documents. Furthermore, one of the limitations of the traditional method is that it is easy to break if the attacker knows the method with static stego-key. In this paper, we propose improvement methods which enhance the security level of the traditional method by using the dynamic stego-key concept to hide information in tagged documents.