

Lời cảm ơn

Trước hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới thầy giáo Thạc sỹ Lê Thụy – giảng viên Bộ môn CNTT, trường Đại học Dân lập Hải Phòng đã tận tình giúp đỡ em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành báo cáo đề án tốt nghiệp.

Em xin chân thành cảm ơn các thầy cô trong bộ môn tin cũng như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành báo cáo.

Cuối cùng, em xin cảm ơn gia đình và các bạn đã động viên, góp ý và trao đổi hỗ trợ cho em trong suốt thời gian vừa qua.

Và mặc dù đã cố gắng nghiên cứu, tìm hiểu và hoàn thành báo cáo trong phạm vi và khả năng cho phép song chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của Thầy cô và các bạn .

Hải phòng, ngày 20 tháng 3 năm 2009
Sinh viên

Nguyễn Thanh Vân

MỤC LỤC

Lời nói đầu.....	3
CHƯƠNG 1 TỔNG QUAN VỀ KỸ THUẬT GIẤU TIN.....	4
1.1 Giới thiệu chung về giấu tin	4
1.1.1 Vài nét về lịch sử giấu tin.....	4
1.1.2 Khái niệm giấu tin.....	4
1.1.3 Một số tính chất giấu tin	6
1.2 Phân loại các kỹ thuật giấu tin.....	6
1.2.1 Giấu tin mật.....	7
1.2.2 Thủy vân số	7
1.3 Một số ứng dụng.....	8
CHƯƠNG 2 VĂN BẢN VÀ GIẤU TIN TRONG VĂN BẢN	10
2.1 Các loại bảng mã tiếng Việt.....	10
2.1.1 Bảng mã TCVN3	10
2.1.2 Bảng mã Unicode.....	11
2.2 Các loại văn bản.....	14
2.2.1 Văn bản text.....	14
2.2.2 Siêu văn bản HTML	16
2.2.3 Văn bản theo chuẩn Microsoft Word (.doc).....	22
2.3 Giấu tin trong văn bản	22
2.3.1 Giấu tin trong HTML	22
2.3.2 Giấu tin trong DOC	28
CHƯƠNG 3 CHƯƠNG TRÌNH MÔ PHỎNG	33
3.1 Mô tả thuật toán:	33
3.2 Quá trình thực hiện của chương trình	34
3.3 Giao diện chương trình giấu tin trong HTML	34
KẾT LUẬN	37
Tài liệu tham khảo	39
PHỤ LỤC	40

Lời nói đầu

Ngày nay, cùng với sự phát triển mạnh mẽ của ngành khoa học công nghệ thông tin, internet đã trở thành một nhu cầu, phương tiện không thể thiếu đối với mọi người, nhu cầu trao đổi thông tin qua mạng ngày càng lớn. Và với lượng thông tin lớn như vậy được truyền qua mạng thì nguy cơ dữ liệu bị truy cập trái phép cũng tăng lên vì vậy vấn đề bảo đảm an toàn và bảo mật thông tin cho dữ liệu truyền trên mạng là rất cần thiết. Nhiều kỹ thuật đã được nghiên cứu nhằm giải quyết vấn đề này. Một trong những kỹ thuật quan trọng nhất là mã hóa thông tin. Tuy nhiên một thông điệp bị mã hóa dễ gây ra sự chú ý và một khi các thông tin mã hóa bị phát hiện thì các tin tặc sẽ tìm mọi cách để giải mã.

Một công nghệ mới phần nào giải quyết được những khó khăn trên là giấu thông tin trong các nguồn đa phương tiện như các nguồn âm thanh, hình ảnh ... Xét theo khía cạnh tổng quát thì giấu thông tin cũng là một dạng mật mã nhằm đảm bảo tính an toàn của thông tin, nhưng phương pháp này ưu điểm ở chỗ là giảm được khả năng phát hiện ra sự tồn tại của thông tin trong các nguồn mang.

Giấu thông tin là một kỹ thuật còn tương đối mới và đang phát triển rất nhanh, thu hút được cả sự quan tâm của giới khoa học và giới công nghiệp và cũng còn nhiều thách thức. *Nội dung của báo cáo này chủ yếu nghiên cứu về kỹ thuật giấu tin nói chung và giấu tin trong văn bản nói riêng*

CHƯƠNG 1 TỔNG QUAN VỀ KỸ THUẬT GIẤU TIN

1.1 Giới thiệu chung về giấu tin .

1.1.1 Vài nét về lịch sử giấu tin

Từ Steganography bắt nguồn từ Hi Lạp và được sử dụng cho tới ngày nay, nó có nghĩa là tài liệu được phủ (covered writing). Có nhiều câu chuyện lịch sử xoay quanh kỹ thuật giấu tin được lưu truyền. Có lẽ những ghi chép sớm nhất về kỹ thuật giấu tin thuộc về sử gia Hy Lạp Herodotus. Khi bạo chúa Hi Lạp Histiaeus bị vua Darius bắt giữ ở Susa vào thế kỷ thứ năm trước Công Nguyên, ông ta đã gửi một thông báo bí mật cho con rể của mình là Aristagoras ở Miletus. Histiaeus đã cạo trọc đầu của một nô lệ tin cậy và xăm một thông báo trên da đầu của người nô lệ ấy. Khi tóc của người nô lệ đủ dài người nô lệ được gửi tới Miletus.

Một câu chuyện khác về thời Hi Lạp cổ đại cũng do Herodotus ghi lại. Demeratus, một người Hi Lạp, cần thông báo cho Sparta rằng Xerxes định xâm chiếm Hi Lạp. Để tránh bị phát hiện, anh ta đã bóc lớp sáp ra khỏi các viên thuốc và khắc thông báo lên bề mặt các viên thuốc này, sau đó bọc lại viên thuốc bằng một lớp sáp mới.

Ý tưởng về che giấu thông tin đã có từ hàng ngàn năm về trước nhưng kỹ thuật này được dùng chủ yếu trong quân đội và trong các cơ quan tình báo. Ngày nay, kỹ thuật giấu tin được nghiên cứu để phục vụ các mục đích như bảo vệ bản quyền, hay giấu tin mật về quân sự và kinh tế.

1.1.2 Khái niệm giấu tin

- Giấu tin là kỹ thuật nhúng một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác.

- Trong quá trình giấu tin để tăng bảo mật, có thể phải dùng khóa viết mật. Đó là loại giấu tin có xử lý. Nếu không dùng khóa viết mật để Giấu tin, tức là chỉ dấu tin đơn thuần vào môi trường phủ. Đó là loại Giấu tin đơn thuần.

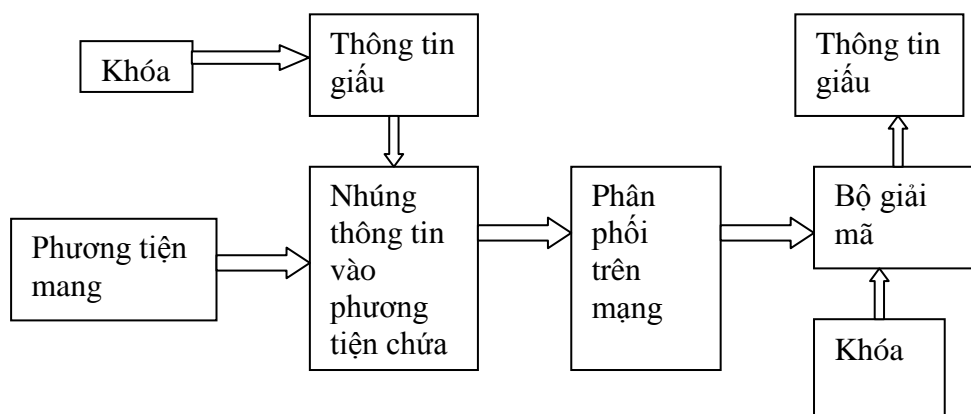
- Yêu cầu cơ bản của giấu tin là đảm bảo tính chất ẩn của thông tin được giấu đồng thời không ảnh hưởng đến chất lượng của dữ liệu gốc.

- Sự phát triển của công nghệ thông tin đã tạo ra những môi trường giấu tin mới rất tiện lợi và phong phú như văn bản, hình ảnh, âm thanh, các phần mềm tiện ích hay cũng có thể giấu tin ngay trong các khoảng trống, phân vùng ẩn của đĩa cứng, đĩa mềm.

Các phương pháp giấu tin được tiến hành theo nhiều cách khác nhau tùy vào mục đích và môi trường giấu tin. Mỗi kỹ thuật giấu tin gồm :

- Thuật toán giấu tin
- Bộ giải mã thông tin

Thuật toán giấu tin được dùng để giấu thông tin vào một phương tiện mang bằng cách sử dụng một khóa bí mật được dùng chung bởi người mã và người giải mã.



Lược đồ chung cho giấu tin

Hình vẽ trên biểu diễn quá trình giấu tin cơ bản. Phương tiện chứa bao gồm các đối tượng được dùng làm môi trường để giấu tin như text, audio, video, ảnh,... thông tin giấu là mục đích của người sử dụng. Thông tin giấu là một lượng thông tin mang một ý nghĩa nào đó như ảnh, logo, đoạn văn bản ... tùy thuộc vào mục đích của người sử dụng. Thông tin sẽ được giấu vào trong phương tiện chứa thông qua chương trình. Sau khi giấu tin ta thu được phương tiện chứa bản tin đã giấu và phân phối trên mạng. Sau khi nhận được đối tượng phương tiện có giấu tin, quá trình giải mã được thực hiện thông qua chương trình giải mã tương ứng với chương trình mã hóa cùng với khóa của quá trình mã hóa. Kết quả thu được gồm phương tiện chứa gốc và

thông tin đã giấu. Bước tiếp theo thông tin giấu sẽ được xử lý kiểm định so sánh với thông tin giấu ban đầu.

Tóm lại, giấu thông tin là nghệ thuật và khoa học của truyền thông, mục đích của Steganography là che giấu những thông báo bên trong những thông báo khác mà không làm ảnh hưởng đáng kể đến thông báo này, và bằng một cách thức nào đó sao cho người không có thẩm quyền không thể phát hiện hoặc không thể phá hủy chúng.

1.1.3 Một số tính chất giấu tin

Một kỹ thuật giấu tin được đánh giá dựa trên một số đặc điểm sau :

- Tính vô hình của thông tin được giấu
- Dung lượng giấu
- Tính bền vững

Tính vô hình : thể hiện mức độ biến đổi môi trường giấu tin. Một phương pháp tốt sẽ làm cho thông tin mật trở nên vô hình trên môi trường giấu tin, người dùng không thể phát hiện trong đó có ẩn chứa thông tin. Tuy nhiên không phải lúc nào người ta cũng cố gắng để đạt được tính vô hình cao nhất. Ví dụ trong bảo vệ bản quyền.

Dung lượng giấu : dung lượng giấu được tính bằng tỷ lệ của lượng tin giấu so với kích thước môi trường giấu. Vì tin mật được gửi cùng với môi trường giấu mang qua mạng nên đây cũng là một chỉ tiêu quan trọng. Các phương pháp đều cố làm sao giấu được nhiều tin 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 tính vô hình, tính bền vững.

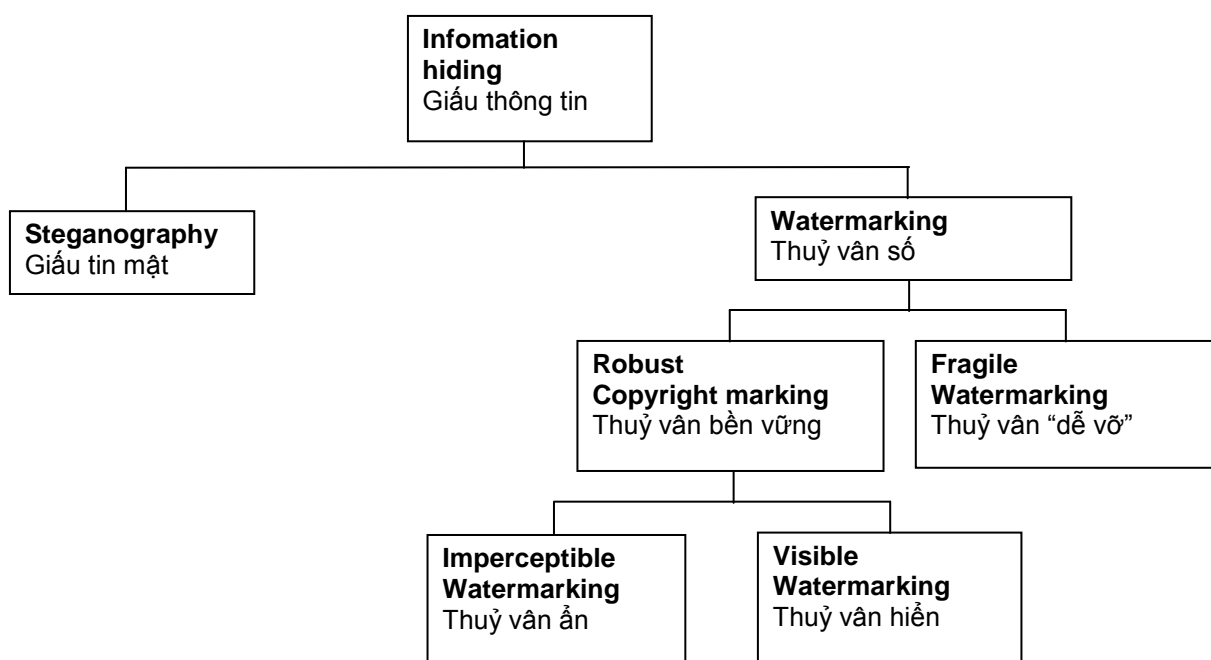
Tính bền vững : Sau khi giấu tin vào môi trường giấu, bản thân chính những môi trường giấu đó có thể phải trải qua các khâu biến đổi khác nhau. Tính bền vững là thước đo sự nguyên vẹn của thông tin mật sau những biến đổi.

1.2 Phân loại các kỹ thuật giấu tin.

Mục đích của việc giấu tin là đảm bảo an toàn và bảo mật thông tin. Có 2 khía cạnh cần được quan tâm đó là:

- + Bảo mật cho dữ liệu được đem giấu .
- + Bảo mật cho chính đối tượng được đem giấu thông tin .

Hai khía cạnh khác nhau này dẫn đến 2 khuynh hướng kỹ thuật chủ yếu của giấu tin : steganography và watermarking.



1.2.1 Giấu tin mật

(*Steganography*) quan tâm tới việc giấu các tin sao cho thông tin giấu được càng nhiều càng tốt và quan trọng là người khác khó phát hiện được một đối tượng có bị giấu tin bên trong hay không bằng kỹ thuật thông thường.

1.2.2 Thủy vân số

(*Watermaking*) đánh dấu vào đối tượng nhằm khẳng định bản quyền sở hữu hay phát hiện xuyên tạc thông tin. Thủy vân số được phân thành 2 loại thủy vân bền vững và thủy vân dễ vỡ.

- *Thủy vân bền vững*: thường được ứng dụng trong các ứng dụng bảo vệ bản quyền. Thủy vân được nhúng trong sản phẩm như một hình thức dán tem bản quyền. Trong trường hợp này, thủy vân phải tồn tại bền vững cùng với sản phẩm nhằm chống việc tẩy xóa, làm giả hay biến đổi phá hủy thủy vân.

+ *Thủy vân ẩn*: cũng giống như giấu tin, bằng mắt thường không thể nhìn thấy thủy vân.

+ *Thủy vân hiện*: là loại thủy vân được hiện ngay trên sản phẩm và người dùng có thể nhìn thấy được.

- *Thủy vân dễ vỡ*: là kỹ thuật nhúng thủy vân vào trong ảnh sao cho khi phân bố sản phẩm trong môi trường mở nếu có bất cứ một phép biến đổi nào làm thay đổi đối tượng sản phẩm gốc thì thủy vân đã được giấu trong đối tượng sẽ không còn nguyên vẹn như trước khi dấu nữa (dễ vỡ).

So sánh giữa steganography và watermarking

	Steganography	Watermaking
Mục đích	- Che giấu sự hiện hữu của thông điệp - Thông tin che giấu độc lập với vỏ bọc	-Thêm vào thông tin bản quyền -Che giấu thông tin gắn với đối tượng vỏ bọc
Yêu cầu	Không phát hiện được thông điệp bị che giấu Dung lượng tin được dấu	Tiêu chuẩn bền vững
Tấn công thành công	Phát hiện ra thông điệp bí mật bị che giấu	Watermaking bị phá vỡ

1.3 Một số ứng dụng.

- *Liên lạc bí mật* : trong nhiều trường hợp sử dụng mật mã có thể gây ra sự chú ý ngoài mong muốn. Ngoài ra việc sử dụng công nghệ mã hóa có thể bị hạn chế hoặc cấm sử dụng. Ngược lại việc giấu tin trong môi trường nào đó rồi gửi đi trên mạng ít gây sự chú ý. Có thể dùng nó để gửi đi một bí mật thương mại, một bản vẽ hoặc các thông tin nhạy cảm khác.

- *Bảo vệ bản quyền* : một biểu tượng bí mật được nhúng vào trong ảnh để xác nhận quyền sở hữu. Biểu tượng thủy ấn (Watermark) có thể có cấu trúc phức tạp được đính lên ảnh khi bán hoặc phân phối, thêm vào đó có thể gán một nhãn thời gian (time stamp) để chống giả mạo. Một Watermark cũng được dùng để phát hiện xem các ảnh có bị sửa đổi hay không? Việc phát hiện các Watermark được thực hiện bởi phép

thống kê, so sánh độ tương quan bằng cách đo đạc xác định chất lượng của Watermark trong ảnh phủ.

- *Gán nhãn* : tiêu đề, chú giải và nhãn thời gian cũng như các minh họa khác có thể được nhúng vào ảnh, ví dụ đính tên người lên ảnh của họ hoặc đính tên vùng địa phương lên bảng đồ. Khi đó nếu sao chép ảnh thì cũng sẽ sao chép cả các dữ liệu nhúng trong nó. Và chỉ khi có chủ sở hữu của tác phẩm, người có được khóa mật (Stego-Key) mới có thể tách ra và xem các chú giải này. Trong một cơ sở dữ liệu ảnh, người ta có thể nhúng các từ khóa để các động cơ tìm kiếm có thể tìm nhanh một bức ảnh. Nếu ảnh là một khung ảnh cho cả một đoạn phim, người ta có thể gán cả thời điểm diễn ra sự kiện (timing) để đồng bộ hình ảnh với âm thanh. Người ta cũng có thể gán số lần ảnh được xem để tính tiền thanh toán theo số lần xem.

CHƯƠNG 2 VĂN BẢN VÀ GIẤU TIN TRONG VĂN BẢN

Trước kia để an toàn thông tin truyền đi người ta chỉ mã hóa thông tin điều này là khá an toàn tuy nhiên dễ bị bọ tin tặc nghi ngờ, nhưng giờ đây để an toàn hơn người ta kết hợp việc mã hóa thông tin rồi sau đó giấu tin vào trong ảnh việc giấu tin vào trong ảnh nhằm mục đích là không nhìn thấy đối với những kẻ không là người nhận hợp pháp.

Bạn hãy tưởng tượng xem trong các kỳ thi tốt nghiệp, kỳ thi đại học của quốc gia để gửi đề thi về cho các đơn vị bằng đường bưu điện thì nhanh cũng phải mất 24 giờ nhưng nếu đề thi bị lộ thì sẽ ảnh hưởng tới thời gian thi của đơn vị bị lộ đề đó và sẽ gây ra rất nhiều tổn kém vậy tại sao ta không gửi qua Internet điều này vừa nhanh lại vừa an toàn.

Để có thể làm được như trên thì chúng ta cần phải *hiểu rõ về các loại văn bản và bảng mã được sử dụng trong văn bản* đó để có thể thực hiện giấu tin trong đó.

2.1 Các loại bảng mã tiếng Việt

2.1.1 Bảng mã TCVN3

Trước hết ta tìm hiểu các khái niệm về điểm mã, đơn vị mã, bảng mã.

- Về bản chất, máy tính chỉ làm việc với các con số, do đó để biểu diễn các ký tự trên máy tính cần phải có một quy ước nhất quán giữa các ký tự cần biểu diễn và các con số tương ứng mà máy tính xử lý. Quy ước này được thể hiện qua các bước:

+ Chọn tập các ký tự cần mã hóa.

+ Gán cho mỗi ký tự cần mã hóa một giá trị nguyên không âm, gọi là điểm mã.

+ Chuyển các điểm mã thành dãy các đơn vị mã để phục vụ cho việc lưu trữ và mã hóa. Các điểm mã không nhất thiết phải có cùng số đơn vị mã.

- Tập hợp những điểm mã của một tập các ký tự được gọi là *bảng mã*. Như vậy khi nói về một bảng mã, chúng ta quan tâm đến hai điều số lượng các ký tự được mã hóa, và cách mã hóa chúng thành các đơn vị mã.

VD : bảng mã ASCII, tập ký tự cần mã hóa có 128 ký tự. Các điểm mã có giá trị nằm trong khoảng từ 0-127. Mỗi điểm mã được mã hóa bằng đúng một đơn vị mã 8 bit

- TCVN là bảng mã dùng 16 bit để biểu diễn các điểm mã, có giá trị nằm trong khoảng từ 0000 đến FFFF, biểu diễn các tập ký tự cơ bản và tập ký tự mở rộng bao gồm các nhóm chữ cái, dấu thanh, dấu nguyên âm, chữ số, ký hiệu, dấu câu và ký tự nguyên âm mang dấu thanh.

2.1.2 Bảng mã Unicode

- Unicode ra đời nhằm thống nhất chung các ký tự của mọi ngôn ngữ trong một bảng mã duy nhất. Phiên bản đầu tiên xuất bản vào năm 1991 bởi hiệp hội Unicode. Các phiên bản mới được viết dựa trên các phiên bản đã có, nhờ vậy đảm bảo được tính tương thích.

- Khi đề cập đến thuật ngữ Unicode thì người ta thường quan tâm hai vấn đề :

+ Tập ký tự mà Unicode biểu diễn: tập ký tự và cách ánh xạ các ký tự bằng các điểm mã tương ứng.

+ Cách mã hóa các điểm mã thành các đơn vị mã.

- Unicode dùng 16 bit để biểu diễn các điểm mã, do đó nó có thể biểu diễn được đến 65536 ký tự.

- Mã cho phép từ 0 đến FFFF. Mã nhỏ nhất 0 dùng cho NULL.

- Do hầu hết các máy tính hiện nay vẫn còn dùng bộ mã ASCII nên chúng không nhận ra các mã lớn hơn 7F. Mặt khác hệ ASCII vốn chỉ xử lý dữ liệu theo từng chuỗi 8bit, mà Unicode được mã hóa dưới dạng 16bit như vậy sẽ gây ra nhầm lẫn khi xử lý các ký tự . Vậy để giải quyết vấn đề thì người ta đã sáng chế ra các cách mã hóa. Mỗi cách mã hóa như vậy được gọi là một dạng biến đổi của Unicode (UTF – Unicode Transformation Format). Thông dụng nhất hiện nay là UTF_16, UTF-8 dùng dãy các đơn vị mã có độ dài khác nhau để mã hóa các điểm mã.

+ UTF-16 : là 1 kiểu mã hóa Unicode trong đó mỗi ký tự được biểu diễn dưới dạng 1 chuỗi 16bit tương đương với giá trị của mã. UTF-16 được dùng trong các hệ thống như Windows 2000/XP, Java, ...

VD : trong UTF-16 chữ “à” được mã hóa thành chuỗi 16 bit 0001 1110 1010 0111(=1EA7) tương đương với mã gốc của “à” trong bảng Unicode.

+ UTF-8 : là 1 kiểu mã hóa khác cho các ký tự Unicode, trong đó mỗi ký tự được biểu diễn dưới dạng 1 hay nhiều chuỗi 8bit để hệ ASCII có thể nhận diện, có thể ko tương đương với mã gốc. UTF-8 thường được dùng trong các ứng dụng web

Cách mã hóa của UTF-8

- 128 ký tự đầu tiên của Unicode từ điểm mã U+0000 đến U+007F, được mã hóa thành 1 byte.

- Từ điểm mã U+0080 đến U+07FF, được mã hóa thành 2 byte.

- Từ điểm mã U+0800 đến U+FFFF, được mã hóa thành 3 byte.

VD : trong UTF-8 chữ “à” được mã thành 3 chuỗi 8 bit
1110 0001 1011 1010 1010 0111(=E1BAA7)

- Cách hoán chuyển UTF-16 qua UTF-8 và ngược lại được làm theo bảng.

Mã	UTF-16		UTF-8		
0-7F	byte 1 00000000	byte 2 0xxxxxxx	0xxxxxxx		
80-7FF	byte 1 00000yyy	byte 2 yyxxxxxx	byte 1 110yyyyy	byte 2 10xxxxxx	
800-FFFF	byte 1 zzzzyyyy	byte 2 yyxxxxxx	byte 1 1110zzzz	byte 2 10yyyyyy	byte 3 10xxxxxx

VD : Mã hóa ký tự Unicode tiếng Việt “à” (mã = 1EA7) dùng UTF-8:

- Viết mã thành 1 chuỗi 16 bit (UTF-16): 0001 1110 1010 0111 tương đương với 1EA7.

Chia chuỗi 16 bit thành 2 byte : byte 1 là: 0001 1110

byte 2 là : 1010 0111.

$7FF < 1EA7 < FFFF$. Theo bảng trên, chuyển đổi cho ký tự “à” sẽ có 3 chuỗi 8 bit.

Đối chiếu với byte 1 và byte 2 trong cột UTF-16, ta sẽ có:
 $zzzz = 0001$; $yyyyyy = 111010$; và $xxxxxx = 100111$.

Đối chiếu với byte 1, byte 2, byte 3 trong cột UTF-8, ta sẽ có dạng UTF-8:

Byte 1 là : 1110zzzz = 11100001. (=E1)
Byte 2 là : 10yyyyyy = 10111010. (=BA)
Byte 3 là : 10xxxxxx = 10100111. (=A7)

Tổng hợp lại, ký tự “à” đã được mã hóa dưới dạng UTF-8 là:

E1BAA7

Lưu ý : bây giờ ta có 3 byte cho ký tự “à” , khác với ký tự gốc là 2 byte

Hầu hết các trang web tiếng Việt sử dụng cách mã hóa UTF-8 để đảm bảo tính tương thích, tuy nhiên một số trang web vẫn còn giữ cách mã hóa theo chuẩn cũ. Các trình duyệt hiện nay như Mozilla Firefox có chức năng tự động chọn cách mã hóa thích hợp nếu như máy tính đã được cài đặt một font thích hợp.

2.2 Các loại văn bản

2.2.1 Văn bản text

- ASCII (American Standard Code for Information Interchange) là bộ mã đầu tiên lúc máy tính được phát minh. ASCII quy định mối tương quan giữa các kiểu bit số với ký hiệu/ biểu tượng trong ngôn ngữ viết, vì vậy cho phép các thiết bị số liên lạc với nhau và xử lý, lưu trữ, trao đổi thông tin hướng ký tự.

- ASCII được công bố làm tiêu chuẩn lần đầu tiên vào năm 1963 bởi Hiệp hội tiêu chuẩn Hoa Kỳ.

- ASCII là mã 7 bit, tức là nó dùng 7bit biểu diễn 128 mã , phần lớn là các ký số, ký tự tiếng Anh, những ký tự đặc biệt và thông dụng khác như các dấu cộng, trừ, phần trăm ... Mặc dù đơn vị dữ liệu là 8 bit nhưng chỉ có 7 bit được dùng, bit thứ tám thường được dùng bit chẵn-lẻ để kiểm tra lỗi trên các đường thông tin hoặc kiểm tra chức năng đặc hiệu theo thiết bị. Các máy tính không dùng chẵn-lẻ thường thiết lập bit thứ tám là zero, nhưng một số thiết bị như máy PRIME chạy PRIMOS thiết lập bit thứ tám là một.

- Mã cho phép từ 0 đến 7F

Mã nhỏ nhất 0 dùng cho ký tự NULL, mã lớn nhất 7F dùng cho phím DEL

32 ký tự đầu tiên là các ký tự điều khiển không in được như ký tự Enter (mã 13), ký tự ESC (mã 27).

Các mã ASCII 32-47, 58-64, 91-96 và 123-127 là các ký tự đặc biệt như dấu chấm, dấu phẩy, dấu cách, dấu ngoặc, dấu móc, dấu hỏi, ...

Các mã ASCII 48-57 là 10 chữ số

Các mã ASCII 65-90 là các chữ cái hoa từ A đến Z

Các mã ASCII 97-122 là các chữ cái thường từ a đến z

1 ký tự được mã hóa thành 1 byte

Bảng mã ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	>	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Bảng mã ANSI là bảng mã ASCII mở rộng, dùng 8 bit để biểu diễn 256 mã. Mã cho phép từ 0 đến FF. Mã nhỏ nhất là 0, dùng cho ký tự NULL, mã lớn nhất là FF. ASCII là tập con của ANSI.

Bảng mã ASCII mở rộng

128	Ç	144	É	161	í	177	⌘	193	⌞	209	ƒ	225	β	241	±
129	ü	145	æ	162	ó	178	⌘	194	⌞	210	π	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⌞	211	⌞	227	π	243	≤
131	â	147	ô	164	ñ	180	†	196	—	212	⌞	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	‡	197	+	213	ƒ	229	σ	245	∫
133	à	149	ò	166	ª	182	‡	198	†	214	π	230	μ	246	÷
134	â	150	û	167	º	183	π	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	168	¿	184	‡	200	⌞	216	‡	232	Φ	248	°
136	ê	152	—	169	—	185	‡	201	π	217	∫	233	⊙	249	.
137	ë	153	Ö	170	¬	186	‡	202	⌞	218	∫	234	Ω	250	.
138	è	154	Û	171	½	187	‡	203	ƒ	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	‡	204	‡	220	■	236	∞	252	—
140	î	157	¥	173	¡	189	‡	205	=	221	■	237	φ	253	²
141	ì	158	—	174	«	190	‡	206	‡	222	■	238	e	254	■
142	Ä	159	f	175	»	191	‡	207	±	223	■	239	∩	255	
143	Å	160	á	176	⌘	192	L	208	⌞	224	α	240	≡		

Source: www.LookupTables.com

2.2.2 Siêu văn bản HTML

- Khái niệm HTML (HyperText Markup Language): là ngôn ngữ đánh dấu siêu văn bản được thiết kế ra để tạo nên các trang web. file HTML là 1 văn bản chứa các thẻ, các thẻ này giúp các trình duyệt web hiểu được cách trình bày và hiển thị 1 trang web.

- Trong file HTML các phần tử được đánh dấu bằng các thẻ HTML. Các thẻ này được bao bởi dấu < và dấu >. Thông thường các thẻ HTML được dùng theo một cặp <tên thẻ> (thẻ bắt đầu) và </tên thẻ> (thẻ kết thúc), văn bản nằm giữa cặp thẻ này là nội dung của phần tử. Các thẻ HTML không phân biệt chữ hoa chữ thường.

- Có thể tạo tài liệu HTML bằng bất kỳ trình soạn thảo nào Notepad, Worpap, Prontpage, DreamWeaver, Jereator.

- Ưu và nhược điểm của trang web khi viết bằng HTML

+ Ưu điểm : - thiết kế đồ họa đẹp

- Tốc độ truy cập nhanh vì không mất thời gian truy vấn CSDL.

- Chi phí đầu tư thấp vì không phải xây dựng CSDL, chi phí cho thuê chỗ, chi phí yêu cầu hệ điều hành tương đối ít.

+ Nhược điểm : - khó khăn trong việc thay đổi và cập nhật thông tin

- thông tin không có tính linh hoạt, không thân thiện với người dùng.

- khó thích hợp nâng cấp mở rộng.

- Các trang Web khác nhau, tế nhạ hay sinh động, vô nghĩa hoặc rất cần thiết, thì tất cả đều có cùng một cấu trúc cơ sở.

- *Cấu trúc cơ bản của 1 trang web*

```
< HTML >  
  < Head > < Title > tiêu đề của trang web < / Title >  
  < / head >  
  < Body >  
    Nội dung của trang web  
  < / Body >  
< / HTML >
```

Các tập tin HTML luôn bắt đầu bằng thẻ <HTML>. Thẻ này làm nhiệm vụ báo cho trình duyệt Web biết rằng nó đang đọc một tài liệu có chứa các mã HTML. Tương tự, dòng cuối trong tài liệu luôn luôn là thẻ </HTML>, tương đương như Hết.

Để xác định phần đầu, đưa thêm thẻ <HEAD> và thẻ </HEAD> vào ngay sau thẻ <HTML>. Mặc dù có thể đặt một số chi tiết bên trong phạm vi phần đầu này, nhưng phổ biến nhất là tên trang. Nếu có ai đó xem trang này trong browser, thì tên trang sẽ xuất hiện trong dải tên của cửa sổ browser.

Để xác định tên trang, đặt đoạn văn bản tên đó giữa các thẻ <TITLE> và </TITLE>.

Phần thân là nơi nhập vào các văn bản sẽ xuất hiện trên trang Web lẫn các thẻ khác quy định dáng vẻ của trang. Nội dung sẽ được đặt trong các thẻ <BODY> và </BODY> sau phần đầu (dưới </HEAD>).

- Một số thẻ định dạng

+ định dạng font chữ : chọn kiểu chữ hiển thị

văn bản hoặc đoạn văn bản

+Các thuộc tính : face = " tên font "

Size = " n "

Color = màu

+ thẻ xuống dòng : < Br > Thẻ này không có thẻ đóng, nó có tác dụng chuyển sang dòng mới.

+ thẻ định dạng in ký tự : các thẻ này quy định thuộc tính in đậm, in nghiêng, gạch chân, ... cho ký tự, văn bản khi được thể hiện trên trình duyệt.

thẻ in chữ đậm : < B > chuỗi văn bản < / B >

thẻ in chữ nghiêng < I > chuỗi văn bản < / I >

thẻ in chữ gạch chân < U > chuỗi văn bản < / U >

thẻ in chữ bị gạch ngang < S > chuỗi văn bản < / S >

+ căn lề văn bản : muốn có một trang web với bố cục đẹp thì phải chú ý tới việc căn lề.

< P align="kiểu căn chỉnh"> đoạn văn bản < /P>

Kiểu căn chỉnh : left : căn lề trái

Center : căn giữa trang

Right : căn lề phải

Ngoài ra, có thể dùng < CENTER >đoạn văn bản</CENTER>

+ thẻ tạo bảng

```
<table các thuộc tính ><Tr><Td>...</Td></Tr></table>
```

Các thuộc tính : BgColor =màu

Border=n :độ dày đường viền

BorderColor = màu

< Tr > < / Tr > tạo dòng trong bảng

< Td > < / Td > tạo ô trong dòng

+ thẻ chèn âm thanh

```
<Embed src = “Tên file âm thanh”
```

```
AutoStart=True/False With=n Hieght=m Loop>
```

+ thẻ chèn hình ảnh

src là viết tắt của source (nguồn) và tên tập tin là tên và đường dẫn của tập tin ảnh mà bạn muốn hiển thị (dùng dạng GIF hoặc JPG)

+ thẻ tạo liên kết < a href =”URL” > Nhãn định danh </ a>

URL : địa chỉ của 1 website hoặc đường dẫn của 1 trang cần liên kết đến

href là viết tắt của hypertext reference (tham chiếu siêu văn bản). Chỉ cần thay từ địa chỉ bằng địa chỉ thực của trang Web mà bạn muốn dùng để liên kết

Thẻ tạo form

<Form *các thuộc tính* > Các loại Form </Form>

+ Các thuộc tính :

Action =URL cho biết các thông tin đưa vào Form sẽ gửi đi đâu

Method : Post/Get phương thức nhận dữ liệu

+ Các loại Form :

Hộp văn bản :

Nhãn định danh <Input type="text" name=tên hộp

size =n value = "giá trị ban đầu (nếu có)">

Hộp Password :

<Input type ="Password" name=tên hộp size=n value= "giá trị ban đầu">

Hộp văn bản ẩn :

<Input type = "hidden" name =tên hộp size=n value = "giá trị ban đầu">

Hộp văn bản lớn :

<Textarea name= tên hộp rows=n

cols=m wrap>giá trị ban đầu(nếu có) </Textarea>

Hộp lựa chọn :

<Input type="radio" name=tên hộp value=giá trị>nhãn định danh

Hộp kiểm :

```
<Input type="checkbox" name=tên hộp
```

```
Value =giá trị checked>nhãn định danh
```

Danh sách lựa chọn :

```
<Select name=tên hộp size=1 multiple>
```

```
<Option value=giá trị 1> Nhãn định danh 1
```

...

```
<Option value=giá trị n> Nhãn định danh n
```

```
</ Select>
```

Trong đó : size=1 :combo box không có multiple

Size >=2 listbox, sd multiple ko cho phép chọn nhiều giá trị cùng lúc

Hộp đệ trình :

```
<Input type "submit" size=n value=giá trị>
```

Hộp nút lệnh :

```
<Input type="button" size=n value =giá trị>
```

Hộp chọn file :

```
<Input type="file" name =tên hộp>
```

- Màu sắc trong thiết kế web

Màu sắc có thể được xác định qua thuộc tính *bgcolor=* hay *color=* . Sau dấu “ = ” có thể là giá trị RGB hay tên tiếng Anh của màu (với tên tiếng Anh có 16 màu, giá trị RGB có 256 màu).

Màu sắc	Giá trị	Tên tiếng anh
Đỏ	#FF0000	RED
Đỏ sẫm	#8B0000	DARKRED
Xanh lá cây	#00FF00	GREEN
Xanh nhạt	#90EE90	LIGHTGREEN
Xanh nước biển	#0000FF	BLUE
Vàng	#FFFF00	YELLOW
Vàng nhạt	#FFFFE0	LIGHTYELLOW
Trắng	#FFFFFF	WHITE
Đen	#000000	BLACK
Xám	#808080	GRAY
Nâu	#A52A2A	BROWN
Tím	#FF00FF	MAGENTA
Tím nhạt	#EE82EE	VIOLET
Hồng	#FFC0CB	PINK
Da cam	#FFA500	ORANGE

2.2.3 Văn bản theo chuẩn Microsoft Word (.doc)

Microsoft Word, còn được biết đến với tên khác là **Winword**, là một công cụ soạn thảo văn bản khá phổ biến hiện nay của công ty phần mềm nổi tiếng Microsoft. Nó cho phép người dùng làm việc với văn bản thô (text), các hiệu ứng như phông chữ, màu sắc, cùng với hình ảnh đồ họa và nhiều hiệu ứng đa phương tiện khác như âm thanh, video khiến cho việc soạn thảo văn bản được thuận tiện hơn. Ngoài ra cũng có các công cụ như kiểm tra chính tả, ngữ pháp của nhiều ngôn ngữ khác nhau để hỗ trợ người sử dụng.

Các phiên bản của Word thường lưu tên tệp tin với đuôi là **.doc** hay **.docx** đối với Word 2007.

Phiên bản Word đầu tiên chạy trên hệ điều hành MS-DOS được giới thiệu vào 11/1983. Và phiên bản gần đây nhất được đưa ra giới thiệu vào năm 2006 là Office Word 2007

2.3 Giấu tin trong văn bản

2.3.1 Giấu tin trong HTML

Muốn giấu thông tin trên một trang HTML ta không thể chèn thêm thông tin vào chính trang HTML bởi có thể hoặc thông tin sẽ được hiện lên trong trình duyệt hoặc được nhìn thấy trong mã nguồn.

Tuy nhiên, có thể thông qua sự thay đổi thứ tự của các thuộc tính để có thể giấu thông tin mà không làm thay đổi, ảnh hưởng đến tệp HTML

Ví dụ :

```
<Span class = "bigText" style = "color: # 0088ff">  
</ Span>  
<Span style = "color: # 0088ff" class = "bigText">  
</ Span>
```

Với mỗi sự kết hợp của các **class** và **style** ta có thể quy ước và giấu thông tin qua đó. Như vậy cần 80 cặp thẻ Spans để ẩn 10 ký tự của một bí mật văn bản. Ngoài cặp thẻ Spans, HTML còn cung cấp rất nhiều định dạng thẻ. Dưới đây là một vài ví dụ. Thuộc tính khóa đầu tiên có thể có nghĩa là "1", tương ứng thuộc tính đầu tiên có thể có nghĩa là "0".

Cặp thuộc tính	Bit dấu	Cặp thuộc tính	Bit dấu
<code>width - height</code>	1	<code>Height - width</code>	0
<code>src - alt</code>	1	<code>Alt - src</code>	0
<code>align - valign</code>	1	<code>Valign - align</code>	0
<code>href - target</code>	1	<code>Target - href</code>	0

Để hiểu rõ hơn vấn đề này ta xem xét ví dụ sau


Canaries

The Finches who got their Name from Islands which got their Name from Dogs




The most canaries are yellow, even though they can have all thinkable patterns of **white, red and grey**. [click here to see photos.](#)

Male birds are great singers. [click here to listen to a sample.](#)



You cannot keep canaries in a cage all day long. They can get sick, if you don't let them fly.

Another big mistake is to keep one canary alone. Every bird needs at least one partner, loneliness can cause bad disorders.



To be continued with more well known stuff about well known birdies...

Ở trên, ta sẽ thấy đó là một trang web giới thiệu về loài chim fanatic.

```
<Html>
<Head>
  <Title> Canary Birds </ title>
  <Meta name = "author" content = "Peter Miller">

  <Style>
    . bigText (font-size: 14px; font-weight: bold;)
  </ Style>
</ Head>
<body text = "# 000000" bgcolor = "# FFFFFFF" link = "# FF0000"
      alink = "# FF0000" vlink = "# FF0000">
  <Div align = "center" width = "50%">
    <H1> Canaries </ h1>
    <Span class = "bigText" style = "color: # 0088ff">
      The Finches who got their Name from Islands
      which got their Name from Dogs
    </ Span>
  </ Div>
```


Có năm các cặp thuộc tính hữu ích:

Cặp thuộc tính	Bit dấu	Cặp thuộc tính	Bit dấu
name - content	1	Content - name	0
text - bgcolor	1	Bgcolor - text	0
alink - vlink	1	Vlink - alink	0
align - width	1	Width - align	0
class - style	1	Style - class	0

Mỗi cặp thuộc tính chỉ xảy ra một lần, do đó, phân đầu của tài liệu chỉ có thể ẩn năm bit. Ta xét tiếp phần còn lại của trang web:

```

<Table width = "60%" height = "100" cellpadding = "4" cellspacing = "0"
      bgcolor = "white" align = "center">

  <Tr>
    <Td align = "right" valign = "middle">
      <Img src = "exampleImage.jpg" width = "164" height = "116"
          alt = "Yellow Bird" title = "Yellow Bird" border = "0">
    </ Td>
    <Td align = "left" valign = "top">
      The most canaries are yellow, even though they can have
      all thinkable patterns of
      <Span class = "bigText"
        style = "color: # ffffff; nền: # 000000">white</ span>,
      <Span class = "bigText" style = "color: # bb0000">red</ span>
and
      <Span class = "bigText" style = "color: # 888888">grey</
span>.
      <A href = "#" target = "_blank">click here to photos </ A>
    </ Td>
  </ Tr>
  <Tr>
    <Td align = "right" valign = "top">
      Male birds are great singers.
      <A href = "#" target = "_blank">click here to listen to a
sample. </ A>
    </ Td>
    <Td align = "left" valign = "middle">
      <Img src = "exampleImage2.jpg" width = "164" height = "176"
          alt = "Singing Bird" title = "A Canary is singing"
border = "0">
    </ Td>
  </ Tr>
  <Tr>
    <Td align = "left" valign = "top">
      You cannot keep canaries in a cage all day long.
      They can get sick, if you don't let them fly.
    </ Td>
    <Td align = "left" valign = "top">
      Another big mistake is to keep one canary alone.
      Every birds need at least one partner,
      Loneliness can lead to bad disorders.
    </ Td>
  </ Tr>
</ Tr>

```

```

        <Td colspan = "2">
            <Img src = "exampleImage3.jpg" width = "194" height = "35"
                alt = "feather" title ="A Canary Feather" border = "0">
        </ Td>
    </ Tr>
</ Table>
</ Body>
</ Html>

```

Trong phần này của các tài liệu, ta thêm được các cặp thuộc tính :

Cặp thuộc tính	Bit dấu	Cặp thuộc tính	Bit dấu
width - height	1	Height - width	0
src - alt	1	Alt - src	0
title - border	1	Border - title	0
Cellspacing-cellpadding	1	Cellpadding -cellspacing	0
bgcolor - align	1	Align - bgcolor	0
align - valign	1	Valign - align	0
href - target	1	Target - href	0

Sự kết hợp của **width** và **height** xảy ra bốn lần, do đó có thể ẩn tới 4 bit. **src** và **alt** xuất hiện 3 lần, do đó có thể ẩn tới 3 bit. Tương tự 3 bit tiếp từ **title** và **border**, **cellpadding** / **cellspacing** 1 bit, **bgcolor** / **align** 2 bit. **align** / **valign** 6 bit, **href** / **target** cho biết thêm 3 bit. Cùng với 5 bit từ ở trên, tài liệu đã có thể tới ẩn 26 bit, tương đương với 3 ký tự .

Ba ký tự không phải là đủ cho một lá thư dài, nhưng đủ để nói "no!", trong bảng mã ASCII thì có giá trị ", 110 111 033" ("01101110 01101111 00100001" ~ 6E 6F 21). Thực hiện ẩn thông tin (nhị phân) tại các cặp từ khóa.

```

<Meta name = "author" content = "Peter Miller">

Quy ước : name/content là "1", content/name là "0".
Đổi lại thứ tự các thuộc tính, để ẩn một giá trị "0":

<Meta content = "Peter Miller" name = "author">

```

Một bit đã được ẩn. bit tiếp theo, ...

```

<body text = "# 000000" bgcolor = "# FFFFFFFF" link = "# FF0000"
    alink = "# FF0000" vlink = "# FF0000">

text/bgcolor là "1", bgcolor/text là "0".
alink/vlink là "1", vlink/alink là "0".
"1" và "1" là mã cần ẩn, vì thế không có thay đổi đối với các dòng .

< body text = "# 000000" bgcolor = "# FFFFFFFF" link = "# FF0000"
    alink = "# FF0000" vlink = "# FF0000">

```

```
<Img src = "exampleImage.jpg" width = "164" height = "116" alt = " Yellow Bird "
      title = " Yellow Bird " border = "0">
```

Mã cần ẩn "010".

Thuộc tính khóa đầu tiên trong thẻ này là "src",

tìm thuộc tính tương ứng "alt".

Các bit cần ẩn là "0", là sự kết hợp của "0" là alt / src,

do đó ta đặt thuộc tính "alt" trước thuộc tính "src".

```
<Img alt = "Yellow Bird" src = "exampleImage.jpg" width = "164" height = "116"
      title = " Yellow Bird " border = "0">
```

Các thuộc tính khóa tiếp theo là "width", thuộc tính tương ứng là "height".

Bit cần ẩn là "1", do đó, đặt "height" sau "width".

Thuộc tính khóa là "title", và thuộc tính tương ứng của nó là "border".

Để ẩn một "0", thuộc tính "title" đặt phía sau "border".

```
<Img alt = "chim vàng" src = "exampleImage.jpg" width = "164" height = "116"
      border = "0" title = "chim vàng">
```

Đánh giá : Với các cặp thuộc tính trên ta có nhận xét sau :

Trang HTML	Cặp thuộc tính	Số lần lặp	Số bit dấu	Tỉ lệ giấu
Canary	name - content	1	1	1.1 : 1
	text - bgcolor	1	1	
	alink - vlink	1	1	
	align - width	1	1	
	class - style	1	1	
	width - height	4	4	
	src - alt	3	3	
	title - border	3	3	
	Cellspacing-cellpadding	1	1	
	bgcolor - align	2	2	
	align - valign	6	6	
	href - target	3	3	
	30(cặp tt)	12	27	

2.3.2 Giấu tin trong DOC

Giấu tin bằng cách nào?

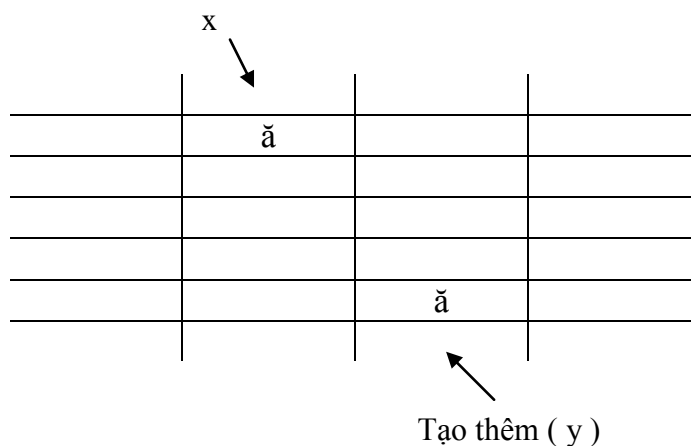
Ta sẽ thay đổi các thuộc tính không nhìn thấy của văn bản. Các thuộc tính này rất nhiều và máy tính không thể theo dõi tất cả. Có rất nhiều phương pháp giấu tin và sẽ lãng phí rất lớn thời gian và tiền vào việc xử lý giấu tin mà lại cho kết quả bấp bênh. Giấu tin sẽ hiệu quả hơn khi được sử dụng trong mã hoá đúng cách. Và chúng tôi sẽ chọn cách thay thế ký tự trong bảng mã Unicode và thay đổi chế độ màu sắc.

thay thế ký tự trong bảng mã Unicode

Trong chế độ này, chúng tôi sẽ thay thế các ký tự trong văn bản được giấu tin. Bit 0 được thể hiện bằng sự xuất hiện của ký tự ban đầu. Bit 1 được thể hiện bằng sự xuất hiện của ký tự thay thế. Trong bảng mã Unicode biểu diễn rất nhiều ký tự của các ngôn ngữ của các dân tộc khác nhau trên thế giới, vì thế việc lấy một ký tự không dùng đến thay đổi và thay thế vào ký tự cần dấu tin là việc không khó.

Không phải tất cả các phong chữ đều hỗ trợ điều này, vì thế chương trình sẽ khuyên bạn nên giải mã tin nhắn của bạn trước khi bạn gửi nó để đảm bảo rằng đó đúng là các thông điệp bạn cần gửi đi. Người nhận phải có cùng một font chữ mà bạn sử dụng cài đặt trong máy tính của mình. Điều này sẽ có lẽ không được coi là một vấn đề cho bạn bởi vì có một loạt các font Unicode phổ biến trong các phiên bản cho hệ điều hành Windows. Người nhận cũng phải biết lựa chọn khóa mà bạn sử dụng cho việc giải mã.

Ví dụ : văn bản để giấu thông điệp sử dụng font Time New Roman



Chữ ã có vị trí (mã) là x -> bit 0, y -> bit 1

Việc tạo thêm này chỉ khi ta xem xét mã của các chữ thì mới thấy có sự khác biệt. Và ta chỉ thực hiện giấu các bit trên các **nguyên âm** và 8 bit đầu sẽ được dùng để dành cho độ dài thông điệp giấu.

Ký tự	Mã	Bit dấu	Ký tự thay thế	Mã	Bit dấu
a	61	0	a	05E4	1
â	00E2	0	â	062A	1
á	00E1	0	á	05D1	1
A	41	0	A	0636	1
Ă	0102	0	Ă	0680	1
e	0065	0	e	069F	1
ê	00EA	0	ê	0663	1
è	00E8	0	è	05EA	1
i	0069	0	i	05E1	1
í	00ED	0	í	0631	1
o	006F	0	o	06D6	1
ô	00F4	0	ô	0692	1
ơ	01A1	0	ơ	06A2	1
õ	00F5	0	õ	FB97	1
u	0075	0	u	FB73	1

Ví dụ : thông tin cần ẩn là "no!", và ẩn trong đoạn văn bản dưới đây

“ Đã bao giờ bạn tự hỏi thành công là gì mà bao kẻ bỏ cả cuộc đời mình theo đuổi? Phải chăng đó là kết quả hoàn hảo trong công việc, sự chính xác đến từng chi tiết? Hay đó là cách nói khác của từ thành đạt, nghĩa là có được một cuộc sống giàu sang, được mọi người nể phục? Vậy thì bạn hãy dành chút thời gian để lặng mình suy ngẫm. Cuộc sống sẽ chỉ cho bạn có những người đạt được thành công theo một cách giản dị đến bất ngờ. “

Chuỗi bit cần ẩn là "00000011 01101110 01101111 00100001". 8 bit đầu là độ dài thông tin ẩn

Đã bao giờ bạn tự hỏi thành công là gì mà bao kẻ bỏ cả cuộc đời mình theo đuổi?

Bit cần dấu "00000011"

Quy ước : a / 61 / 0 , o / 006F / 0 , i / 0069 / 0 , i / 0069 / 0 , ô / 00F4 / 0 , ...

Phải chăng đó là kết quả hoàn hảo trong công việc, sự chính xác đến từng chi tiết?

Bit cần dấu "1110"

Quy ước : i / 05E1 / 1 , o / 06D6 / 1 , o / 06D6 / 1 , ô / 00F4 / 0 , ...

Hay đó là cách nói khác của từ thành đạt, nghĩa là có được một cuộc sống giàu sang, được mọi người nể phục? Vậy thì bạn hãy dành chút thời gian để lặng mình suy ngẫm. Cuộc sống sẽ chỉ cho bạn có những người đạt được thành công theo một cách giản dị đến bất ngờ.

Đánh giá : giả sử với bảng khóa như trên

Trang văn bản	Số lượng từ	Số bit dấu	Tỉ lệ dấu
Khỏ A4, font Time New Roman, cỡ chữ 13	720	370	

Chế độ thay đổi Màu sắc

Chế độ này rất ổn định và an toàn. Trong chế độ này, ta sẽ thay đổi màu sắc của các ký tự trong các văn bản được giấu tin theo chuỗi bit dấu. Ta sẽ tìm kiếm màu sắc gần nhất với màu sắc đã lựa chọn mà với mắt thường thì không thể nhận ra được sự khác biệt. Bit 0 được thể hiện bằng sự xuất hiện của các ký tự có màu sắc ban đầu. Bit 1 được thể hiện bằng sự xuất hiện của các ký tự có màu sắc gần với màu sắc ban đầu nhất. Người nhận phải biết về màu sắc mà bạn đã chọn cho việc mã hóa. Và cũng giống với chế độ thay thế ký tự trong bảng mã Unicode thì chế độ thay đổi màu sắc cũng dành 8 bit đầu cho độ dài thông điệp dấu.

Ví dụ : ta chọn màu **đen** để ẩn thông tin cần dấu

255 255 255 -> bit 0

thì 255 255 254 -> bit 1

Để hiểu rõ hơn về chế độ thay đổi màu sắc thì ta xét ví dụ sau :

thông tin cần ẩn là "no!", và ẩn trong đoạn văn bản dưới đây

“ Đã bao giờ bạn tự hỏi thành công là gì mà bao kẻ bỏ cả cuộc đời mình theo đuổi? Phải chăng đó là kết quả hoàn hảo trong công việc, sự chính xác đến từng chi tiết? Hay đó là cách nói khác của từ thành đạt, nghĩa là có được một cuộc sống giàu sang, được mọi người nể phục? Vậy thì bạn hãy dành chút thời gian để lặng mình suy ngẫm. Cuộc sống sẽ chỉ cho bạn có những người đạt được thành công theo một cách giản dị đến bất ngờ. “

Chuỗi bit cần ẩn là "00000011 01101110 01101111 00100001". 8 bit đầu là độ dài thông tin ẩn

Đã bao giờ bạn tự hỏi thành công là gì mà bao kẻ bỏ cả cuộc đời mình theo đuổi?

Bit cần dấu "00000011"

Quy ước : Đ / 255 255 255 / 0 , ã / 255 255 255 / 0 , b / 255 255 255 / 0 ...

Phải chăng đó là kết quả hoàn hảo trong công việc, sự chính xác đến từng chi tiết?

Bit cần dấu "001"

Quy ước : P / 255 255 255 / 0 , h / 255 255 255 / 0 , ả / 255 255 254 / 1 ...

Hay đó là cách nói khác của từ thành đạt, nghĩa là có được một cuộc sống giàu sang, được mọi người nể phục? Vậy thì bạn hãy dành chút thời gian để lặng mình suy ngẫm. Cuộc sống sẽ chỉ cho bạn có những người đạt được thành công theo một cách giản dị đến bất ngờ.

Đánh giá : giả sử với bảng khóa như trên

Trang văn bản	Số lượng chữ	Số bit dấu	Tỉ lệ dấu
Khổ A4, font Time New Roman, cỡ chữ 13	2280	2280	1 : 1

CHƯƠNG 3 CHƯƠNG TRÌNH MÔ PHỎNG

Môi trường làm việc : Window XP

Ngôn ngữ thực hiện thuật toán : ngôn ngữ lập trình C# với trình biên dịch Microsoft Visual C#

Chương trình được thiết kế, cài đặt với mục đích thể hiện thuật toán giấu tin trong văn bản

3.1 Mô tả thuật toán:

- Input:

Thông điệp đầu: Một chuỗi các bit .

Một văn bản.

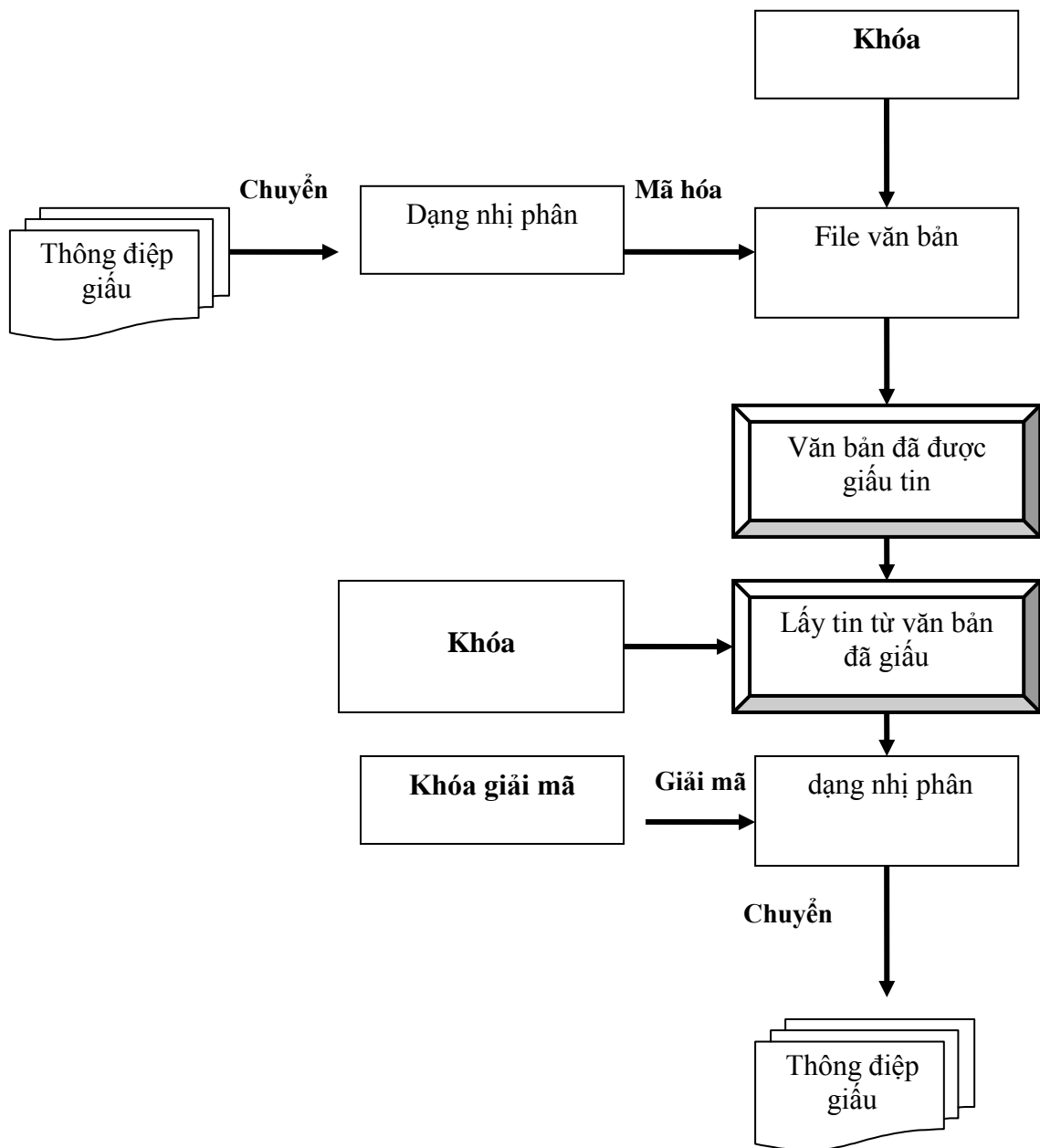
- Output:

Văn bản gốc.

Thông điệp giấu.

Khoá để giải mã.

3.2 Quá trình thực hiện của chương trình



3.3 Giao diện chương trình giấu tin trong HTML

Trang HTML để thực hiện giấu tin

Canary Birds - Microsoft Internet Explorer


File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Print Mail Stop

Address <C:\Documents and Settings\hiepvann\Desktop\canary.htm> Go Links


Canaries

The Finches who got their Name from Islands which got their Name from Dogs



The most canaries are yellow, even though they can have all thinkable patterns of white, red and grey. [click here to photos](#)

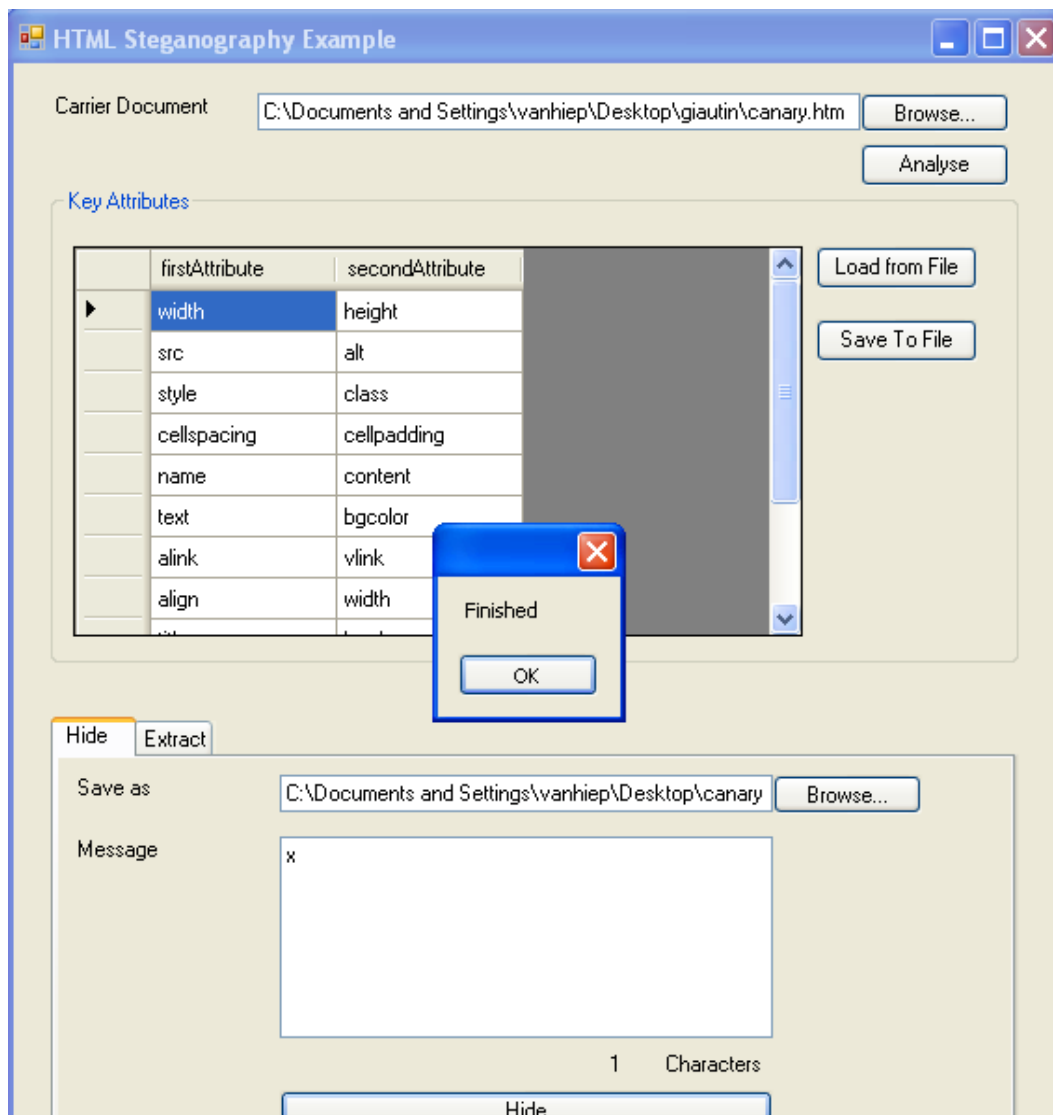
Male birds are great singers. [click here to listen to a sample.](#)



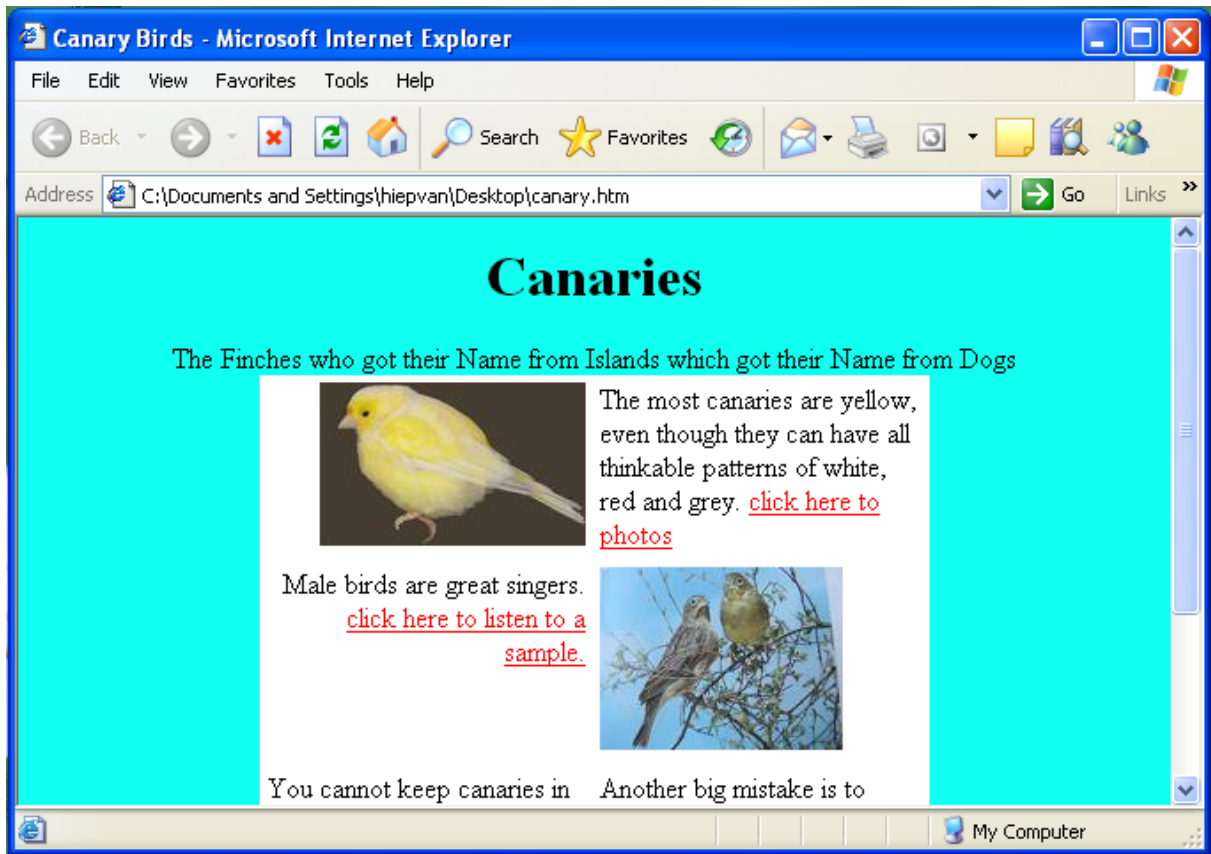
You cannot keep canaries in Another big mistake is to

My Computer

Thực hiện giấu tin



Trang HTML sau khi giấu tin



KẾT LUẬN

Đồ án đã trình bày một số khái niệm liên quan tới việc che giấu thông tin nói chung, cụ thể hơn là nghiên cứu giấu tin trong văn bản.

Với thuật toán giấu tin trong văn bản thì tính vô hình của thông tin sau khi giấu được đảm bảo với những biến đổi không gây ra sự chú ý đáng kể nào. Trang HTML và văn bản DOC sau khi giấu tin sẽ ít có sự thay đổi và quan trọng hơn là với thị giác của con người thì không thể phân biệt được sự khác nhau đó.

Tính bền vững ở đây được hiểu là trang HTML hay văn bản DOC sau khi đã giấu thông tin bởi một hệ thống giấu tin hay một chương trình giấu tin nào đó có chịu được các cuộc tấn công có chủ đích hay không? Đối với cách giấu thông tin ở đây, thì trang HTML có bị lộ mã nguồn hay không thì phụ thuộc vào máy chủ Web có được bảo mật tốt hay không? Còn đối với văn bản DOC thì hoàn toàn không có tính bền vững bởi khi văn bản được giấu tin chuyển đi mà máy tính của người nhận không có font chữ như người gửi thì font chữ sẽ bị hỏng.

Còn dung lượng giấu tỉ lệ với môi trường giấu, trong trang HTML càng nhiều cặp thuộc tính hữu ích thì càng giấu được nhiều thông tin, trong văn bản DOC cũng tương tự. Tuy nhiên trong khi giấu tin thì người ta luôn cân nhắc giữa các chỉ tiêu.

Tài liệu tham khảo

- [1] . Nguyễn Xuân Huy, Trần Quốc Dũng, “ *Giáo trình giấu tin và thủy vân ảnh*”.
- [2] . <http://www.codeproject.com/steganography>
- [3] . Nguyễn Hữu Tuấn “ *Giáo trình thiết kế Web* ”.
- [4] . Dùng Unicode chữ Việt trong .Net

PHỤ LỤC

CÁC THỦ TỤC CHÍNH CỦA CHƯƠNG TRÌNH

Giấu tin trong HTML

Trước tiên ta cần tới class để lưu trữ các thẻ HTML và các thuộc tính của chúng.

```
public class HtmlAttribute (
    private String name;
    private String value;
    private bool handled;

    public String Name (
        get ( return name; )
    )

    public String Value (
        get ( return this.value; )
        set ( this.value = value; )
    )

    public bool Handled {
        get ( return handled; )
        set ( this.handled = value; )
    }

    public HtmlAttribute (String name) (
        this.name = name.ToLower ();
        this.value = String.Empty;
        handled = false;
    )
)
```

Phương thức `Hide` liệt kê tất cả các thẻ HTML, và sau đó lặp trên các thẻ và các thuộc tính. Thuộc tính đó đã được xử lý thì bỏ qua. Nếu một thuộc tính không sử dụng tới, phương thức sẽ lưu nó vào trong bảng khóa ...

```
public void Hide (String SourceFileName,
    String destinationFileName,
    Stream message,
    DataTable keyTable)
(
    // Đọc tài liệu
    StreamReader reader = new StreamReader (Sourcefilename, Encoding.Default);
    String htmlDocument = reader.ReadToEnd ();
    reader.Close ();

    message.Position = 0;

    // Danh sách các thẻ HTML
    HtmlTagCollection tags = FindTags (htmlDocument);

    StringBuilder insertTextBuilder = new StringBuilder ();
    DataRow [] rows;
    HtmlAttribute secondAttribute;
    int offset = 0;
    int bitIndex = 7;
```



```

int messageByte = 0;

foreach (HtmlTag tag in tags) (

    insertTextBuilder.Remove (0, insertTextBuilder.Length);
    insertTextBuilder.AppendFormat ( "<{0}", tag.Name);

    foreach (HtmlAttribute attribute in tag.Attributes) (

        if (! attribute.Handled) (// thuộc tính đã không được sử dụng, nhưng
vẫn

            // Tìm khóa cho các thuộc tính này
            rows =
                keyTable.Select (String.Format ( "firstAttribute = '{0}'",
                    attribute.Name));

```

Lấy lại tin đã ẩn

Đi qua các thẻ và các thuộc tính, tìm một thuộc tính khoá chính, và thuộc tính tương ứng của nó, và so sánh các vị trí.

```

public void Extract (String Sourcefilename, Stream message, DataTable keyTable) (

    // ... đọc tài liệu ...
    // ... danh sách các thẻ HTML ...
    // ... khai báo ...

    foreach (HtmlTag tag in tags) (
        foreach (HtmlAttribute attribute in tag.Attributes) (

            if (! attribute.Handled) (// thuộc tính đã không được sử dụng, nhưng
vẫn

                // Tìm khóa cho các thuộc tính này
                rows =
                    keyTable.Select (String.Format ( "firstAttribute = '{0}'",
                        attribute.Name));
                if (rows.Length > 0) (

                    // Tìm thấy thuộc tính tương ứng
                    secondAttribute = FindAttribute (
                        rows [0] [ "secondAttribute"]. ToString (),
                        tag.Attributes);

                    if (secondAttribute != null) (

                        attributePosition = htmlDocument.IndexOf (
                            attribute.Name,
                            tag.BeginPosition);

                        secondAttributePosition = htmlDocument.IndexOf (
                            secondAttribute.Name,
                            tag.BeginPosition);

                        // So sánh vị trí các thuộc tính
                        messageByte = ExtractBit (
                            attributePosition,
                            secondAttributePosition,
                            messageByte,
                            bitIndex,
                            message);

```