A Novel Text Steganography Technique Based on Html Documents

Mohit Garg

Department of Computer Engineering, Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi 110042, India
mohit_270488@yahoo.co.in

Abstract

The internet has grown rapidly in recent years. This growth has increased the demand for techniques that can ensure information security. In this paper, we propose a text steganography technique that uses html documents as the cover medium to hide secret messages. The use of html documents has a benefit that the existence of a secret message will not be suspicious as html documents are fundamental elements of the web and are used very commonly on the internet. We have implemented our technique using C#.net technology. The technique proposed by us also integrates cryptography with steganography by first encrypting the secret message and then hiding the encrypted secret message in html cover medium. The integration of cryptography with steganography provides an extra layer of security that ensures the safe and secure delivery of message to the intended recipient.

Keywords: *Text Steganography, Cryptography, Html documents, Html attributes, Data Hiding*

1. Introduction

The internet is a huge collection of networks. It is a super-highway that connects places all over the world. Internet is one of the rapidly growing technologies in the present era. This growth has focused attention on one of the most important aspect of internet viz. information security. Since internet is a public network, securing the information on internet is very important. Various techniques including cryptography, steganography etc. are used to secure information on the internet. Cryptography is the science of converting the messages that are intended to be secret into some other form, such that it is not understandable to anyone other than the intended sender and recipients. Steganography is a technique for securing information by hiding it in some other medium, such that the existence of information is concealed to everyone except for the intended sender and receiver.

Steganography refers to the art and science of hiding secret information in some other media. The information to be hided is called the secret message and the medium in which the information is hided is called the cover document. The cover document containing hidden message is called stego-document. The algorithms employed for hiding the message in the cover medium at the sender end and extracting the hidden message from the stego-document at the receiver end is called stego system.

Steganography can be broadly classified into three types on the basis of the type of the cover media used viz. text steganography, image steganography, and audio steganography. A steganography technique that uses text as the cover media is called a text steganography. It is one of the most difficult types of the steganography technique. This is because text files have a very small amount of redundant data to hide a secret message. A steganography technique

that uses images as the cover media is called an image steganography. Hiding secret messages in digital images is the most widely used method as it can take advantage of the limited power of the human visual system (HVS) and also because images have a large amount of redundant information that can be used to hide a secret message. A steganography technique that uses audio as the cover media is called an audio steganography. It is the most challenging task in steganography. This is because the human auditory system (HAS) has a large dynamic range that it can listen over. Thus, even a minute change in audio quality also can be detected by the human ears.

Of text, image and audio steganography, text steganography is most challenging due to the presence of very less redundant information in text documents as compared to the images and audio [1]. In this paper, we propose a text steganography technique that hides the secret information in html documents. The attributes of the html documents are used to hide the information. The use of html documents has a benefit that the existence of a secret message will not be suspicious as html documents are fundamental elements of the web and are used very commonly on the internet. The proposed technique combines cryptography with steganography by first encrypting the secret message and then hiding the encrypted secret message in html documents. The proposed technique is also implemented using C#.net programming language.

The rest of the paper is organized as follows. Section 2 presents the previous works that have been done in the field of text steganography. Section 3 presents the proposed technique in detail. Section 4 concludes the paper.

2. Previous Works

There has been tremendous research in the field of text steganography. Some of the text steganography works are listed below.

Moerland proposed a text steganography technique by using specific characters from the words. In this method, some specific characters from certain words are selected and are used to hide the secret information. For e.g. the first character of first word of each paragraph can be used to hide a secret message one character at a time such that by placing these characters side by side, we get the whole message [2].

Moerland also discussed about the text steganography approach by using punctuation marks. The idea behind this approach is to utilize the presence of punctuation marks like comma (,), semi colon (:), quotes (‘, “) etc. in the text for encoding a secret message. The use of punctuation marks is quite common in the normal english text and hence it becomes difficult for the intruder to recognize the presence of secret message in the text document. This accounts for the security of the technique [2].

Low, Maxemchuk, Brassil, Gorman [3] and Alattar [4] proposed a text steganography technique by using line shifting method. In this method, the lines of the text are shifted to some degrees say 1/300 inch up or down and then the information is hidden by creating a hidden unique shape of the text.

Low, Maxemchuk, Brassil, Gorman [3] and Kim, Moon, Oh [5] proposed a text steganography technique using word shifting method. In this method, the information is hidden by shifting the words horizontally or by changing the distance between the words.

Niimi, Minewaki, Noda, Kawaguchi proposed a technique that uses synonyms of certain words to hide the message in the english text. In this method, certain words from the text are selected, their synonyms are identified and then the words along with their synonyms are used to hide the secret message in the text [6].

Huang, Yan proposed a technique for hiding information by adding extra white-spaces in the text. These white spaces can be placed at the end of each line, at the end of each paragraph or between the words [7].

Shirali-Shahreza [8, 9] and Memon, Khowaja, Kazi [10] proposed a steganography method on Arabic, Persian and Urdu text. One of the characteristics of these languages is the abundance of points in its letters. One point letters can be used to hide the information by shifting the position of a point a little bit vertically high with respect to the standard point position in the text.

Alla, Prasad proposed a hindi text steganography technique. This technique is based on the fact that each language has its own characteristics. Every language is formed of combinations of one or more vowels and consonants. These vowels and consonants and the combination of the two, forms the basis of this hindi text steganography technique. This technique makes use of two elements viz. simple letters (pure vowels and pure consonants) and compound letters (combinations of vowels, consonants, vowels and consonants) [11].

Shirali-Shahreza proposed a text steganography technique that hides secret message in the English text by using different spellings of the words. In English some words have different spelling in UK and US. For example "dialog" has different terms in UK (dialogue) and US (dialog). This difference in spellings forms the basis of steganography [12].

Wang, Chang, Kieu, Li proposed an emoticon based text steganography technique. Emoticons are emotional icons that are used in online chatting. These emoticons express the feeling or mood of the persons communicating with each other. The use of emoticons in steganography is quite interesting [13].

3. The Proposed Technique

In this section, we present the proposed technique. We propose a text steganography technique that uses html documents as the cover medium to hide the secret messages.

3.1. Overview of the Proposed Technique

The proposed technique uses the html tags and their attributes to hide the secret message. It is based on the fact that the ordering of the attributes in the html tags has no impact on the appearance of the document. This ordering can be used to hide the secret messages efficiently. The proposed technique essentially has three components viz. key file generation, hiding process and extracting process. Hiding process is used to hide a message in html documents and extracting process is used to extract the hidden message from the html documents. The key component of the technique is the generation of key file.

The key file is essentially a collection of key combinations stored in the form of rows and columns. These combinations are generated by thorough scanning of the html documents. The attributes combinations used in the html tags are used to generate a key file.

The key file contains two types of attributes, corresponding to two columns:

- Primary Attribute
- Secondary Attribute

The primary attribute is in the first column and secondary attribute is in the second column. These attribute combinations aids in the hiding process.

Basic Procedure

- The hiding process scans each attribute of each html tag, and checks to see whether that attribute exists in the primary attribute field of the key file.
- If **yes**, its corresponding secondary attribute is searched in the corresponding html tag. If found, then this combination of attribute is used to hide a bit. If not, skip this attribute.
- The hiding of a bit is determined by the order of the attributes in the attribute combination. If primary attribute is followed by a secondary attribute, it can hide a bit 1; else it can hide a bit 0.
- The extractor program extracts the message from stego text by first identifying the attribute combinations that hides a bit and then finding the bit corresponding to the order of those attributes.
- If primary attribute is followed by secondary attribute, a bit 1 is detected, else a 0 is detected.

3.2. Framework of the Proposed Technique

The proposed technique essentially has the following schema. It depicts the basic flow of the both the hiding and extracting process of the proposed technique as shown in figure 1 and figure 2 respectively.

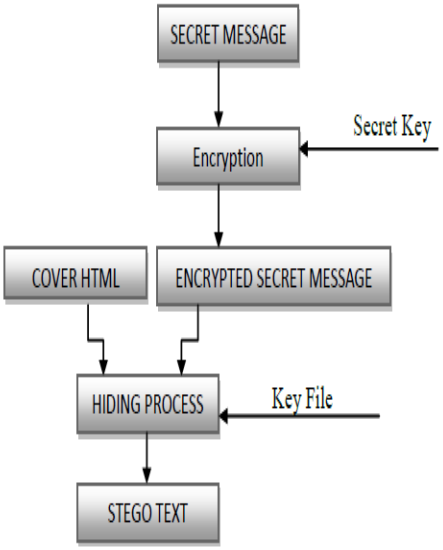

Figure 1. Hiding Process

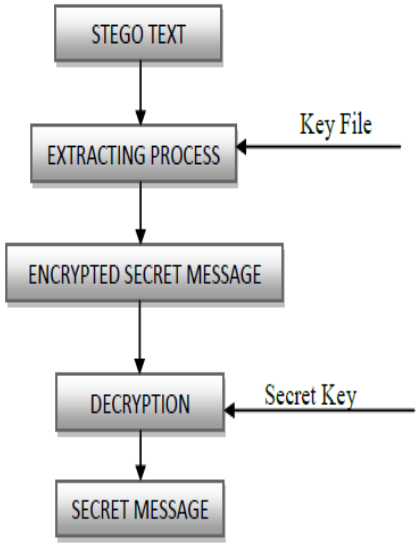

Figure 2. Extracting Process

Figure 1 shows the process of hiding the secret message in the html documents. Figure 2 shows the process of extracting the secret message from the html documents. Encryption and decryption mechanism shown in figure 1 and figure 2 respectively shows the use of cryptography in the proposed text steganography technique. The cryptography scheme used is play fair cipher. Secret message is the message to be hided. Cover html is the document

chosen to be used as the cover medium. Stego text is generated after hiding the secret message in the cover html. Hiding process, extracting process and key file generation of the proposed steganography technique shown in figure 1 and figure 2 respectively are explained in detail in subsequent section.

3.3. Proposed Technique Description

In this section, we describe the proposed technique in detail. We have implemented the proposed technique in C#.net language. It essentially consists of three main components:

1. Key file generation
2. Hiding the message
3. Extracting the message

3.3.1. Key file generation: This is the most important component of the technique. The key fie is essentially the collection of key combinations that are stored in the form of rows and columns. Each combination is actually an attribute pair that is candidate towards hiding a bit. These combinations are derived from the html document.

The key file contains two types of attributes corresponding to two columns; primary attribute and secondary attribute. The format of key file is shown in table 1. The primary attribute is something that represents a bit 1 or 0, depending on its order relative to the secondary attribute.

Table 1. Key File Format

First Attribute (Primary)	Second Attribute (Secondary)

Procedure for key file generation:

Scan HTML document. Analyze each tag. Corresponding to each tag, identify the combination or pair of attributes that can be used. Preferably, select the pair that is used quite frequently in the tags of the html documents. Designate one of the attributes in the selected pair as primary attribute while another as secondary attribute. The primary attribute should follow the uniqueness constraint i.e. it should be unique. There is no special requirement for choosing the primary attribute. You can choose any attribute of the identified pair of attributes as primary as long as the constraint is satisfied. The terminology of primary and secondary is only used to differentiate between the ordering of attribute to represent bits 0 or1, i.e.; primary followed by secondary represents a bit 1 and secondary followed by primary represents a bit 0.

For e.g.; {style, class}, {width, height}, {cellspacing, cellpadding}, {border, width}, {name, value} etc.

Example:

1. Consider the following html code

```
<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="#FFFFFF" 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>
```

2. Following attribute pairs can be identified by scanning the html document as shown in table 2:

Table 2. An Example Key File

First (Primary) Attribute	Second (Secondary) Attribute
name	Content
text	bgcolor
alink	Vlink
align	Width
class	Style

3.3.2. Hiding the message: To hide a message, first convert it in the binary fashion, in terms of bit stream. Then scan the html document to find the attribute combinations that can be used to hide a bit.

Procedure for hiding the message:

1. Encrypt the secret message using play fair cipher encryption mechanism and convert the message in binary format.

2. Scan the html document. Analyze each attribute of each tag of the html document.

3. For each attribute:

3.1. If this attribute is found in the primary attribute field of the key file:

3.1.1. Check if its corresponding secondary attribute is present in the currently being processed tag. If yes, then this pair of attribute can hide a bit. To hide a bit, read one bit of secret message.

3.1.2. If it is 1, then compare the actual order of this pair of attributes in the tag with the desired order according to key file. If primary attribute lies before the secondary in the tag, then order is retained, else it is reversed. Mark both attributes as processed.

3.1.3. If it is 0, then retain the order if secondary attribute lies before the primary, else reverse the order. Mark both attributes as processed.

3.2. If the attribute is not found in the primary attribute field of the key file or if it is marked as processed, then skip this attribute and move to another attribute.

3.3.3. Extracting the message: The extraction component is quite simple. Firstly, it scans the document to find the attribute pairs that hides a bit, using a similar procedure as used for hiding. Once it finds the attribute pair, it compares the positions of the attributes according to the key file.

If primary attribute lies before (as determined by the positions) the secondary attribute, then a bit 1 is recorded else a bit 0 is recorded.

Procedure for extracting the message

1. Scan the html document. Analyze each attribute of each tag of the html document.

2. For each attribute:

2.1. If this attribute is found in the primary attribute field of the key file:

2.1.1. Check if its corresponding secondary attribute is present in the currently being processed tag. If yes, then this pair of attribute hides a bit. To retrieve the hidden bit, check the ordering of attribute.

2.1.2. If the primary attribute is followed by secondary attribute, record a bit 1, else record a bit 0. Mark the attributes as processed after retrieving the bit.

2.2. If the attribute is not found in the primary attribute field of the key file or if it is marked as processed, then skip this attribute and move to another attribute.

3. Convert the bit stream obtained after the completion of step2 into stream of characters. This is the extracted secret message in its encrypted form.

4. Decrypt the encrypted secret message using play fair cipher decryption mechanism to recover the original secret message.

3.4. Proposed technique explained through example

Let us consider a sample key file as shown in table 3:

Table 3. A Sample Key File

Key Attribute	Corresponding Attribute
width	height
src	alt
title	border
cellspacing	cellpadding
bgcolor	align
align	valign
href	target

Let us consider a sample html tag of the cover document:

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

3.4.1. Hiding the message: Suppose we want to hide "010". We follow the hiding process of section 3.3.2 to hide the message.

1. The first key attribute in this tag is "src", so we take the corresponding attribute "alt" from table 3. The first bit to hide is "0". The ordering of this combination of attributes for a bit "0" is alt/src according to table 3. So we place the "alt"-attribute before the "src"-attribute. Mark alt and src as processed.

2.The next key attribute in the sample tag is "width". The corresponding attribute is "height" from table 3. Now, the second bit to hide is "1", so we put "height" after "width". Mark width and height as processed.

3.The third key attribute is "title", and its corresponding attribute is "border" from table 3. To hide a "0", we move "title" behind "border". Mark title and border as processed.

Resulting Tag: <img **alt**="Yellow Bird" **src**="exampleImage.jpg" **width**="164" **height**="116" **border**="0" **title**="Yellow Bird">

3.4.2. Extracting the message: For the resulting tag of section 3.4.1, the message can be extracted using the process described in section 3.3.3 as follows.

Tag: <img **alt**="Yellow Bird" **src**="exampleImage.jpg" **width**="164" **height**="116" **border**="0" **title**="Yellow Bird">

1. Examine the alt attribute. It is not present in the primary field in table 3. Skip it.

2. Examine src attribute. This attribute is present in the primary field in table 3. Its corresponding secondary attribute is also present in the above tag. According to key file in table 3, src/alt=1 and alt/src=0;

→Since; the order is alt/src in the above tag, **we record a bit 0**. Mark both attributes as processed.

3. Examine width attribute. It is found in the primary field of the key file in table 3. Its corresponding secondary attribute is also present in the above tag, so this pair hides a bit. According to table 3, width/height=1 and height/width=0.

→Since order in above tag is width/height, **we record a bit 1**. Mark both attributes as processed.

4. Examine height attribute. Skip it, as it is marked as processed.

5. Examine border attribute. It is not found in primary field of the key file in table 3. Skip it.

6. Examine title attribute. It is found in the primary field of the key file in table 3. Its corresponding secondary attribute is also present in the above tag. So, this pair hides a bit. According to table 3, title/border=1 and border/title=0.

→Since order in above tag is border/title, **we record a bit 0**. Mark both attributes as processed.

Recovered message: **010**

4. Conclusion

For the text steganography various methods have been proposed. In this paper, we propose a novel approach of text steganography that uses the html tags and attributes to hide the secret messages. The basic idea of the proposed technique is to hide the messages by changing the order of attributes as the ordering of attributes does not affect the appearance of the html documents. The html documents are fundamental elements of the web. These documents are used very commonly on the internet and hence are less prone to arouse suspicion in the intruder of the existence of the secret message. Moreover, any html document has a considerable number of tags and attributes. Thus the capacity of the hiding process to hide secret messages is also high in the proposed technique.

References

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding”, IBM Systems Journal, vol. 35, Issues 3&4, 1996, pp. 313-336.

[2] T. Moerland, “Steganography and Steganalysis”, www.liacs.nl/home/tmoerlan/privtech.pdf, May 15, 2003.

[3] S.H. Low, N.F. Maxemchuk, J.T. Brassil, and L. O'Gorman, “Document marking and identification using both line and word shifting”, Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '95), vol.2, 2-6 April 1995, pp. 853 - 860.

[4] A.M. Alattar, and O.M. Alattar, “Watermarking electronic text documents containing justified paragraphs and irregular line spacing”, Proceedings of SPIE -- Volume5306, Security, Steganography, and Watermarking of Multimedia Contents VI, June 2004, pp. 685-695.

[5] Y. Kim, K. Moon, and I. Oh, “A Text Watermarking Algorithm based on Word Classification and Inter word Space Statistics”, Proceedings of the Seventh International Conference on Document Analysis and Recognition(ICDAR’03), 2003, pp. 775–779.

[6] M. Niimi, S. Minewaki, H. Noda, and E. Kawaguchi, “A Framework of Text-based Steganography Using SD Form Semantics Model”, Pacific Rim Workshop on Digital Steganography 2003, Kyushu Institute of Technology, Kitakyushu, Japan, July 3-4, 2003.

[7] D. Huang, and H. Yan, "Inter word Distance Changes Represented by Sine Waves for Watermarking Text Images", IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 12, December 2001, pp. 1237-1245.

[8] M. H. Shirali-Shahreza, and S. Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography", Proceedings of 5th IEEE/ACIS international Conference on Computer and Information Science and 1st IEEE/ACIS, June 2006.

[9] M. H. Shirali-Shahreza, and S. Shirali-Shahreza, "A Robust Page Segmentation Method for Persian/Arabic Document", WSEAS Transactions on Computers, vol. 4, Issue 11, Nov. 2005, pp. 1692-1698.

[10] J.A. Memon, K. Khowaja, and H. Kazi, "Evaluation of steganography for Urdu /Arabic text", Journal of Theoretical and Applied Information Technology, pp 232-237.

[11] K. Alla, and R.S.R Prasad, "An Evolution of Hindi Text Steganography", Sixth International Conference on Information Technology New Generations, 2009 (ITNG '09), Digital Object Identifier: 10.1109/ITNG.2009.41**,** 2009, Page(s): 1577 - 1578.

[12] M. Shirali-Shahreza, "Text Steganography by Changing Words Spelling", International Journal of Advanced Communication Technology, 2008 (ICACT 08), Volume: 3, Digital Object Identifier: 10.1109/ICACT.2008.4494159, 2008, Page(s): 1912 - 1913.

[13] Z.H. Wang, C.C. Chang, D. Kieu, and M.C. Li, "Emoticon-based Text Steganography in Chat", Second Asia-Pacific Conference on Computational Intelligence and Industrial applications, 2009.

Authors

Mohit Garg, He received the B.Tech degree in Computer Science from Guru Gobind Singh Indraprastha University, Delhi, India in 2009. He is now undertaking an M.Tech degree course as a research scholar at Delhi Technological University (formerly known as Delhi College of Engineering), Delhi, India. His areas of interests include Software Engineering, Software Testing, Databases, Information Security. His research interests are in Search Based Software Engineering, Automated Test Data Generation using Genetic Algorithms, Model Prediction using Genetic Algorithms, Software Quality Analysis using Quality Metrics'. Mohit can be contacted by e-mail at mohit_270488@yahoo.co.in.