A Novel Approach of Secret Hiding in Webpage by Bit Grouping Technology

Xiaoming Zhang, Guoqing Zhao, Pengfei Niu
Department of Computer, Beijing Institute of Petro-chemical Technology, Beijing, China
Email: {zhangxiaoming, zhaoguoqing, niupengfei}@bipt.edu.cn

Abstract—Web pages are used popularly and frequently today because the message in the pages can be fast spreading all over the world in real-time. A kind of new approach is constructed for the webpage information hiding. The hiding data is composed of the length of secret, the checksum and the secret data. The secret message is required for modification using forward transform technology before hiding. A kind of grouping 5-bit scheme and algorithm are proposed to convert the hiding data into some invisible characters. It can hide 10 bits in each end tag of page line. In the extracting process, the message is recomposed as the coupling relationship. The hiding process and algorithms are stated in detail. The experiments are implemented with some well-known homepages of portal websites. The results show that the new approach can not only keep the hiding effects, but also increase the hiding capacity largely. It will be suitable for large-scale message transmission through the Internet.

Index Terms—Webpage, Information hiding, Invisible character, Capacity, Algorithm, Network Security

I. INTRODUCTION

Webpage information hiding (WIH) is a kind of technology to hide important message into the web pages. Owing to the global distribution of webpage in real time, the secret message can be transmitted and spread quickly without any other channels. Meanwhile, if some watermarks are embedded into the web pages, these pages can be protected by temper-proof technology. Both of the sides become important study contents of webpage information security.

There exist several approaches for WIH to aim at the HTML documents. Firstly, the tag-based hiding is used as the popular approach of WIH. There are three typical methods including insensitiveness of upper and lower case letters [1], order of tag attributes [2-3] and white space inserting [4] among the tag-based hiding scheme. All the methods are based on HTML specifications and web browsing effects. By changing the page letters as its upper or lower cases, Zhao [1] proposed a PCA-based approach to embed small image watermarks into web pages. Moreover, Sun [5] used m-sequence to generate the watermark and embed them into the tags of the source code of web pages. It can decrease the time consuming in embedding and extracting operation stages. However, the different letter cases throughout the page can be identified easily as a strange page.

For the method of order of tag attributes, it has good transparency in source code. The more the number of attributes increases, the bigger the hiding capacity becomes. However, with the cascade style (CSS) technique popularly used in webpage creation, many tag’s attributes are only defined in CSS. Because there leaves a few attributes for the initial webpage, the hiding effect by attribute order method is not satisfied in many cases. Owing to the specification of CSS, Sun [6] applied some class selectors of CSS in hiding algorithm. The living Class Selectors of some objects are caught and repeatedly imported to embed information. It has preferable imperceptibility, good ability to contradict with detection and auto filtration.

The space inserting method is easily realized in the end tag. One can not quickly identify the white space among the lots of webpage lines. The method can be presented as bit inserting idea since each inserting can show the state of bit 1. The hiding capacity relies on the number of page lines with end tag.

There exist several creative algorithms in WIH, such as tabled-based hiding [7-8] and Flash-based hiding [9]. Many web pages have table styles to present information in normalization. Zhang [7] proposed a new approach for WIH by table specification with parity of the consecutive line or row values. The approach has good transparent performance in source code checking. The approach was updated to increase the capacity by multiple table hiding [8]. On the other hand, multimedia information-based hiding approach is still attractive for WIH based on the existed multimedia hiding algorithms. Most of web pages present multimedia information including images, audios, videos and animations except the text message. These messages can be applied for information hiding, especially the popular logo information in the homepage. Through analyzing the data format in Flash developing environment, Zhang [9] successfully created a kind of method for flash information hiding by the Flash logo.

Meanwhile, Xie [10] proposed a novel method by means of the composition of tag, attribute and quotes. Because the representation formats of the tag attribute values can form 8 kinds within the commonly writing...
habits of people, it could be derived from this phenomenon that the 8 formats of the tag attributes could represent 8 values. It can increase the hiding capacity, and will not increase the size of cover-webpage. Similarly, Yang [11] adopted the tag attribute with single quotes and double quotes to represent the two hiding bit, 0 and 1. Besides, Ren [12-13] proposed ideas of tag dictionary and method integration. With more tag composition, the hiding capacity can be enlarged.

A new kind of approach by inserting invisible characters is proposed here for WIH. Through the bit separation and composition algorithm, the secret message with any character can be embeded into the webpage. Because the inserting information is not bit information but special characters, the hiding capacity increases largely. Next, the hiding idea will be stated firstly. Then, the new algorithm is presented with flow chart and process description. The experiments are carried out for the ordinary portal homepage, while several typical attacks on the hiding algorithm are designed and tested with good effects.

This paper is organized as follows. In Section 2, we analyze the novel webpage hiding idea. Section 3 explains the design solution and algorithm in detail. Experiments and discussion are shown in Section 4. Finally, this paper is concluded in Section 5.

II. ANALYSIS OF HIDING IDEA

There are three considerations for the hiding approach design. Firstly, the browsing web page is always different from the initial page by the source code. It means that the hiding carrier should keep consistence. Secondly, the hiding capacity is required to increase much more than that of bit inserting method. Finally, for the web site construction, the file format in the server side is always different from that in the webpage browsing side. For example, the ActiveX Server Page (ASP) documents will be transformed to HTML format. Therefore, the hiding approach is needed for different source documents.

From the view of the above design ideas, the end tag-based hiding method is adopted for the characters code hiding.

A. Analysis of Invisible Characters set

Definition 1 Invisible character (IC): it means the character inserting into the web page but without display sign when page browsing. The invisible character shows special symbol or white space in the page source code.

There are 33 characters in the ASCII table for IC with ASCII coding values of 0-32. The codes of 1-31 show special symbols. While the code 0 is null, and the ASCII code 32 is white space. All these characters are chosen as invisible characters to form a set \( S_p \).

\[
S_p = \{ s_i | [0 - 32] \} \quad . \tag{1}
\]

Evidently, if the ordinary characters can be translated as one of SC or its composition, then secret message can be hidden into the web pages.

From the view of ASCII coding values, the invisible characters have code less than 32. The ordinary characters are much larger than the ICs in ASCII code. For example, the letter of 0 has code of 48, while the letter of z has 122 coding value. Therefore, translation between the secret characters and ICs are required.

B. Analysis of HTML Tag End Mark

Next, where can these ICs be hidden into the webpage? As it knows, each tag of web page has its end mark. Even though there are different coding forms, it can have the same page browsing effects. Some researchers attached the white spaces in the end of webpage lines. Although this method can hide large information, the hiding state can be easily identified and cleaned by the attacker. Besides, each white space can only represent one bit.

A new kind of hiding position is discovered as follows: If one of the ICs is inserted into the tag end, the character will not be displayed in the webpage. For example, the following inserting forms can be still effective for page browsing normally:

White space inserting state of tag end with three forms: 
\(</table> \), or \(<table > \), or \(<\ table > \)

So, there exist two inserting positions for two ICs: after mark of “<” and before “>”.

C. Five Bit Grouping Scheme

The third question is how to enlarge the hiding capacity. As stated above, how many bits appointed by user can be hidden in one IC?

In order to hide any character from ASCII table, all the secret data have to be converted to the IC. Considering the IC value from 0 to 31, it only need the lower 5 bits in an 8-bit data to represent all the IC value. Therefore, a kind of group scheme is created for the hiding data, as shown in Figure 1. The initial data are grouped as 5 bits, and then each group is added with 3 bits of 0 to its higher position to form a new 8-bit data. For example, one group with 5 bits of 10110 is chosen for IC data creation. After three bits of 0 are inserted to the head of group, the IC data will have value of 22. Its ASCII character belongs to an IC of device control.
D. Analysis of Hiding Contents

In order to assure the extraction correctness, the length of secret should be hidden into the webpage along with the secret data.

Besides, the interrupt of network communication is always taken place. As the network protocol idea, the checksum is very useful to check the composed message in the IP layer and the transmission layer.

Therefore, the hiding content includes three kinds of data as following:

1. Length of secret with 16 bits: $N_s$;
2. Secret data: $D_s$;
3. Checksum calculation to the $N_s$ and $D_s$ with 16 bits.

In the extraction process, the checksum is calculated again for the extracted $N_s$ and $D_s$. If the extracted checksum and the calculation result are matched well, then extraction data are correct.

E. Scrambling Technology

Before the secret is embedded into the web pages, it is required for pre-processing. One choice is encrypting to the secret to form some strange message, and the other is scrambling the secret to modify its content sequence. The former will create some control characters or invisible characters which is not suitable for checking. Therefore, the scrambling technology is adopted here to modify the secret content.

Assume the data of secret is $A[0,1,\ldots,M-1]$, and $P$ and $M$ are designed to be coprime integers. Let $B$ as the temporary data array with the same length of $A$ to store the operating results. Then, the scrambling technology is designed as follows:

1. Forward transform

\[
\begin{align*}
B(i) &= A(i \times P \mod M), i = 0,1,\ldots,M - 1 \\
A &= B
\end{align*}
\]  

(2)

2. Inverse transform

\[
B(i \times P) \mod M = A(i), i = 0,1,\ldots,M - 1
\]

\[
A = B
\]

(3)

F. Adoption of Checksum Calculation

In the famous OSI model in computer network architecture, there exist checksum calculation in two layers of the network layer and transmission layer. The checksum is applied mainly to check the data from the upper layer of OSI. If the calculation result is not consistent with the extracted value of checksum, then the data will be rejected by UDP protocol or retransmitted by TCP protocol.

For the webpage hiding application, the checksum can be used to increase the reliability of secret transmission, and its computation solution is different from that in OSI model.

**Definition 2 Hiding Checksum (HC):** In the secret hiding process, the hiding checksum is applied to check two kinds of data together. One is the length of secret, and the other is the data of secret.

**Definition 3 Hiding Data (HD):** It refers to the data which is actually embedded into the webpage. In order to increase the reliability of web information transmission, the hiding data is composed of three kinds of data, that is, the length of secret, the HC and the data of secret, as shown in Figure 2.

![Figure 2. Composition of hiding data composition in webpage](image)

The length is used to decide the actual number of secret data, while the checksum is for checking the network error and the data consistence. Both of the length and the checksum are 16 bits. Therefore, the final length of hiding data will be as following:

\[
L_{sd} = 32 + 8N_s
\]

III. DESIGN OF HIDING ALGORITHM BASED ON IC

It is composed of two stage of the webpage hiding process. The page is redistributed after secret is embedded into it, while the secret is extracted from the document when browsing the page through the network.

A. Design of Secret Hiding Process Model

The process is as shown in Figure 3.

After the hiding data is embedded into the appointed webpage, the new created webpage $P_2$ is distributed through the web environment. Then, the receiver can obtain the important message by browsing the website with a series of extraction algorithm. The browsed webpage is often different from that existed in the web server.

In the embedding process, there are three key

\[
B(i \times P) \mod M = A(i), i = 0,1,\ldots,M - 1
\]

\[
A = B
\]
Input the secret message $S$

Change $S$ to $X$ by forward scrambling technology

Form the hiding data

Call the embedding algorithm to embed hiding data into the web page $P_1$ to form new page $P_2$

Distribution of page $P_2$

Browsing web page and get its source code document $P_3$

Output the obtained message $S'$

Change $X'$ to the secret $S'$ by inverse scrambling technology

Translate the data parts of $Y'$ into $X'$ as the character forms

Call the extraction algorithm to obtain the bit array $Y$
Browsed webpage P3
  Read all lines from P3
  Search for all the hiding lines as the embedding position scheme
  Extract two characters as IC[0] and IC[1] from the end tag data in hiding line Li
  Extract the lower 5 bits from IC[0] and IC[1] respectively, and then form a bit string Si

Search to the end of P3?

Yes
  Merge all the bit string Si into the final bit array S
  Read 8 bits from S to form an 8-bit data array D_i

No
  Extracted the hiding data D_i

Figure 4. Extraction flow-chart diagram

Input: the hiding data D_i with length of m, and the initial webpage P1 with hiding lines of n (m<n).
Output: the new webpage P2 with secret information.

Embedding procedure:
(1) Convert Dh to the bit stream H_b;
(2) Get the IC data array DataArray as the bit inserting algorithm;
(3) Input P1 to get the hiding capacity and its hiding positioning scheme;
(4) As the end tag format, search for all the hiding lines as Li, i=0,1,…,n;
(5) i←0;
(6) Read one hiding line Li from P1;
(7) Read two IC data from DataArray and convert them to two IC characters.
(8) Renew the end tag in Li from </Ti> to </IC[0]⋅Ti⋅IC[1]>. Here, Ti is a kind of string as HTML specification;
(9) i←(i+1);
(10) Turn to (6) if i≤m. Otherwise, to the next;
(11) Obtain the new webpage P2.

E. Design of Extraction Algorithm
As stated in Figure 3, the extraction process is applied to extract hidden ICs from the appointed webpage to form hiding data as bit stream. The concrete process is described as shown in Figure 5.
Owing to the difference between the initial webpage and the browsed webpage through network, the hiding interval is difficult to set as a fixed value. The extraction software will search automatically for the end tag.

Function of extraction algorithm: drawing the hidden data from the appointed web page to form bit stream.
Input: browsed webpage P3 within secret message.
Output: the hiding data array D_x as bit stream.

Extraction procedure:
(1) Read all hiding lines from P3;

Hiding data
  Convert all the hiding data to bit array
  Convert the bit stream to the IC data array DataArray[i] as the bit insertion algorithm
  Read two data from DataArray[i], then convert them to two IC: IC[0], IC[1]

No
  Scan the whole page to get the hiding lines as the end tag format
  Read one hiding line Li from P1
  Extract the end tag data Ti from Li
  Extract two characters as data IC[0] and IC[1] from the IC data array DataArray[i] as the bit insertion algorithm
  Read two data from DataArray[i], then convert them to two IC
  Compose to new end tag data T_i: T_i=IC[0]T_iIC[1]
  Write T_i back to the hiding line Li in P1

Yes
  Secret embedded webpage P2
  Hiding over?

Figure 5. Embedding flow-chart diagram
(8) Compose all the bit string to \( S = \sum_{i=0}^{n} S_i \).
(9) Convert \( S \) to the 8-bit data array \( D_x \);
(10) Output \( D_x \).

IV. EXPERIMENT DESIGN AND RESULT ANALYSIS

The proposed hiding approach is implemented with C# programming language in VS.NET. It is tested under lots of typical web site by browsing.

A. Performance calculation

The calculation aims to character not the bit as the proposed algorithms.

(1) Calculation of character hiding effects

**Definition 4 Character Error Rate (CER):** Suppose the secret \( S \) with ASCII code \( X = x_1x_2...x_m \) and the extraction message \( X' = x'_1x'_2...x'_m \), the character error rate is defined as the hiding error:

\[
CER = \frac{\sum_{i=1}^{m} |x_i - x'_i|}{\sum_{i=1}^{m} x_i} \times 100\% \quad (9)
\]

(2) Calculation of hiding capacity

As the IC inserting approach, each usable end tag can hide two ICs. As (4) and (5), the required number of hiding lines is concluded as:

\[
N_h = \frac{N_s \times 8 + m}{10}, (0 \leq m < 5) \quad (10)
\]

Where, \( N_s \) is the length of hiding data.

On the other hand, each character value has an 8-bit length. As the existed bit inserting method, two bits can be hidden into one end tag. Then, for the same hiding data, the required number of hiding lines for bit inserting method is:

\[
N_{bit} = \frac{N_s \times 8}{2} = 4N_s \quad (11)
\]

To compare their hiding capacities for the two typical inserting approaches, a kind of new parameter is defined as following:

**Definition 5 Hiding Capacity Ratio:** It refers to ratio of two kinds of required number of hiding lines in webpage. Here, to compare the two approaches of bit inserting and IC inserting, the ratio is defined as:

\[
R_b = \frac{N_{bit}}{N_h} = \frac{40N_s}{8N_s + m} \quad (11)
\]

Because the value of \( m \) is so small that it can be omitted. Then, the ratio is approximately:

\[
R_b = 5 \quad (12)
\]

Furthermore, the relationship between the number of hiding characters and that of webpage hiding lines as (10) can be described as following:

\[
N_s = 1.25N_h \quad (13)
\]

Therefore, the propose approach has good ability with large hiding capacity. It is nearly five times of that in the bit inserting method.

B. Experimenting Design

Some typical web page is tested by getting the HTML source code after browsing the page. These pages are chosen as shown in Table I.

For instance, the secret is selected from one of the news title in web site www.w3.org as following string:

\[\text{W3C Leads Discussion at TypeCon 2010 on New Open Web Font Format}\]

The above string is composed of 64 characters including white space. So, the number of the hiding data is 68 totally.

The 64 secret data will be modified as the forward transform technology before they are merged with the length of secret and the checksum. As described in (2), there are two parameters of \( M \) and \( P \). Here, \( M \) is 64. Now let \( P = 3 \), then the result of forward transform is list below:

\[\text{wassoaTen0 wp bortCesiui po20neOneFtoa3LdDcsntyC 1oN eW nFm}\]

<table>
<thead>
<tr>
<th>TABLE I.</th>
<th>SOME HTML DOCUMENTS FOR EXPERIMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Homepages</strong></td>
<td><strong>Size(KB)</strong></td>
</tr>
<tr>
<td><a href="http://www.163.com">www.163.com</a></td>
<td>365</td>
</tr>
<tr>
<td><a href="http://www.sohu.com">www.sohu.com</a></td>
<td>181</td>
</tr>
<tr>
<td><a href="http://www.bipt.edu.cn">www.bipt.edu.cn</a></td>
<td>22</td>
</tr>
<tr>
<td><a href="http://www.w3.org/Protocols/">www.w3.org/Protocols/</a></td>
<td>24</td>
</tr>
<tr>
<td><a href="http://www.w3.org">www.w3.org</a></td>
<td>29</td>
</tr>
<tr>
<td><a href="http://www.tianwang.com">www.tianwang.com</a></td>
<td>9</td>
</tr>
<tr>
<td><a href="http://www.sogou.com">www.sogou.com</a></td>
<td>5</td>
</tr>
<tr>
<td><a href="http://www.baidu.com">www.baidu.com</a></td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II.</th>
<th>EXPERIMENT RESULTS FOR INFORMATION HIDING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Homepages</strong></td>
<td><strong>Initial size</strong></td>
</tr>
<tr>
<td><a href="http://www.163.com">www.163.com</a></td>
<td>365</td>
</tr>
<tr>
<td><a href="http://www.sohu.com">www.sohu.com</a></td>
<td>181</td>
</tr>
<tr>
<td><a href="http://www.bipt.edu.cn">www.bipt.edu.cn</a></td>
<td>22</td>
</tr>
<tr>
<td><a href="http://www.w3.org/Protocols/">www.w3.org/Protocols/</a></td>
<td>24</td>
</tr>
<tr>
<td><a href="http://www.w3.org">www.w3.org</a></td>
<td>29</td>
</tr>
</tbody>
</table>

The transform result shows secure and effective to make
the secret difficult for understanding.

The secret hiding system for webpage is developed with C# for many functions, such as webpage selection, calculation of hiding capacity, scrambling process for secret, grouping 5-bit technology, and the whole embedding and extraction algorithm. One of the operation interfaces is shown in Figure 6. The grouping result is list in detail to form corresponding ICs. The chosen web page for hiding secret is the default page of web site www.w3.org/Protocols.

After embedding the selected string, the extraction results are shown in Table II. The hidden size of selected documents has small change, while the CER value show zero for all hiding web pages.

The grouping approach has good security, capacity and robustness. The common methods for webpage hiding are compared as shown in Table III. Meanwhile, the source codes in different states are shown in Figure 7.

Even if a few strange characters can be observed throughout the webpage source code, the hidden result is not affected because of its target for secret transmission. It is clearly that the IC inserting approach has good hiding performance with large hiding capacity.

V. Conclusion

Based on the browsing property of invisible character in web pages, a kind of invisible character inserting approach for secret hiding is created into the web pages. The hiding data consist of three kinds of data as the length of secret, the checksum and the secret data. The secret data will be modified using forward transform technology. All the hiding data will be translated into invisible characters by grouping 5-bit method. It increases the hiding capacity with 5 times as the ordinary bit inserting approach. Therefore, the approach will be available for large amount message in the web page hiding.

REFERENCES


© 2012 ACADEMY PUBLISHER
He was a lecturer and associate professor in the Department of Computer of Beijing Institute of Petro-chemical Technology in 1996 and 1998, respectively. He was a visiting scholar of Narvik Institute of Technology in Norway during 2000-2001. Since 2008, he has been a Professor in the Department of Computer of Beijing Institute of Petro-chemical Technology. His current research interests include multimedia information hiding and network system security.

Guoqing Zhao received the M.S. degree in computer testing and measurement from Beijing Institute of Technology in 2000. Since 2001, he has been an associate professor in the Department of Computer of Beijing Institute of Petro-chemical Technology. His research interests include network intrusion detection and system security.

Pengfei Niu received the B.S. degree in computer science and technology from Beijing Institute of Petro-chemical Technology in 2008. Since 2008, he has been studying M.S. degree in Beijing Institute of Petro-chemical Technology. His research interests are in information hiding and watermark in stream media.

Xiaoming Zhang was born in Jiangxi, China on January 25, 1968. He received the B.S. degree in agricultural mechanic automation from the Jiangxi Agricultural University in 1989, the M.S. and Ph.D. degrees in mechanical engineering & automation from Dalian University of Technology in 1992 and 1996, respectively.