Review of Information Hiding in Tagged Documents using Steganography

Kriti Saroha\textsuperscript{1} and Nguyen Xuan Linh\textsuperscript{2}
\textsuperscript{1}School of IT, CDAC, Noida, India.
\textsuperscript{2}Thua Thien Hue Center of Information Technology, Hue City, Vietnam.
\textsuperscript{1}kritisaroha@cdacnoida.in and \textsuperscript{2}nxlinh@gmail.com

ABSTRACT
Information hiding is the technology to embed the secret information into a cover data in a way that keeps the secret information invisible. Nowadays, Tagged documents such as HTML, XML, XHTML and WML are known as the universal format for structured documents and data as well as present data on browser. They are used as the basic language for exchanging information on the Web. As compared to the information hiding methods intended for images and sounds, there are very few methods for hiding information into text, especially on tagged documents. In this paper, various techniques are analyzed for hiding information in tagged documents by exploiting the features of the tag language.

KEYWORDS
Information hiding, Steganography, tagged documents.

1. INTRODUCTION
Steganography or hiding data within data, coming from Greek words stegos, meaning covered and graphia, which means writing, is the art and science of hiding the information within information. With the rapid development of Internet technologies, the amount of information sent and received electronically is increasing rapidly. As the technology of transmitting information on network must be secure, the importance of information hiding came to be recognized widely.

We use a cover data to hide secret information which, later can be extracted when needed. Figure 1 shows the general model of information hiding [4]. For different kinds of data, we need different methods of information hiding. For example, for hiding data in images, we can work with the least significant bits of the pixels in the image to hide information. After embedding the information, the changes to the image are difficult to be recognized by the naked eye. When working with text, changing the location of the punctuations, the choice between synonyms, or spacing between words are simple ways to add information without changing the value of the original text. Similarly, tagged documents can be used for information hiding by changing the structure of the document without affecting their display in the web browser [3, 5, 6].

2. TECHNIQUES FOR INFORMATION HIDING IN TAGGED DOCUMENTS

2.1. Representation of Empty Elements
In this method, we can utilize the feature of tagged documents, where to represent an empty element we can use either a start-tag immediately followed by an end-tag, or an empty-element tag. For example, we can use both formats <img> </img> or <img/> with the same meaning on tagged documents.

Suppose, we define a stego key/rule as following:

- <img> </img> to represent bit “0”
- <img/> to represent bit “1”

Once the stego key is defined in this way, we can embed the data by switching between the two forms to represent “0” and “1” without affecting the meaning of the original document. For instance, if we want to hide character “A” in ASCII code, we need eight tags as discussed below:

\textbf{Example 1:}

\begin{verbatim}
A = 01000001
Stego data: <img src="foo1.jpg"></img><img src="foo2.jpg"></img>
<img src="foo3.jpg"></img><img src="foo4.jpg"></img>
<img src="foo5.jpg"></img><img src="foo6.jpg"></img>
<img src="foo7.jpg"></img><img src="foo8.jpg/>
Embedded data: 01000001
\end{verbatim}
In the above example, for hiding one character we required eight \texttt{<img>} tags. Thus, the amount of data that can be hidden is usually small because it requires a lot of empty element tags in the cover file [4, 8].

2.2. White Spaces in Tags

According to W3C, a tag can include some white spaces before close bracket, or no white space. By inserting or deleting spaces, we can embed the data while preserving all meaning of the original document. For instance, we can define a rule as the following example:

\textbf{Example 2:}

Stego key:
\texttt{<tag>, </tag>, or <tag/>} to represent bit ‘0’
\texttt{<tag>, </tag>, or <tag/>} to represent bit ‘1’

Stego data:
\texttt{<user><name>Alice</name><id>01</id></user>}
\texttt{<user><name>Bob</name><id>02</id></user>}

Embedded data:
101100 010011

This method can embed one bit of data on each tag and does not depend on type of the tag or attributes present in it. Therefore, we can use all of the tags in the cover document to hide the data [4, 8].

2.3. Appearing Order of the Elements

We can also embed secret data in tagged documents by changing the appearing order of the elements. Following is the example of this method.

\textbf{Example 3:}

Stego key:
\texttt{<user><name>NAME</name><id>ID</id></user>} for bit ‘0’
\texttt{<user><id>ID</id><name>NAME</name></user>} for bit ‘1’

Stego data:
\texttt{<user><name>Alice</name><id>01</id></user>}
\texttt{<user><id>02</id><name>Bob</name></user>}

Embedded data:
0110

This method can embed one bit of data on each tag and does not depend on type of the tag or attributes present in it. Therefore, we can use all of the tags in the cover document to hide the data [4, 8].

2.4. Appearing Order of the Attributes

Secret data can also be embedded in the tagged documents by changing the appearing order of the attributes within the element. Following is the example of this method. One bit of data can be hidden per change of two elements [4, 8].

\textbf{Example 4:}

Stego key:
\texttt{<event month=“MONTH” date=“DATE”>EVENT</event>}

This order represents bit ‘0’
\texttt{<event date=“DATE” month=“MONTH”>EVENT</event>}

This order represents bit ‘1’

Stego data:
\texttt{<event month=“JUL” date=“4”>Independence day</event>}
\texttt{<event date=“24” month=“DEC”>Christmas</event>}
\texttt{<event month=“JAN” date=“1”>New Years</event>}
\texttt{<event date=“8” month=“March”>Women Day</event>}

Embedded data:
0101

This method, the order of the attributes works as the secret key for embedding and extracting. We can also define many couple of attributes to enhance the amount of data that can be hidden in the cover document [4, 8].

2.5. White Space between the Elements’s Name and the First Attribute

In tagged documents, we can insert a white space directly after the element’s name and before the first attribute without changing the display properties of the original document.

\textbf{Example 5:}

Stego key:
\texttt{<font face=“verdana” size=“3”> for bit ‘0’}
\texttt{<font face=“verdana” size=“3”> for bit ‘1’}

Stego data:
\texttt{<font face=“verdana” size=“3”> 0}
\texttt{<font face=“Time New Roman” size=“2”> 1}
\texttt{<font face=“Tahoma” size=“3”> 1}
\texttt{<font face=“Arial” size=“3”> 0}

Embedded data: 0110

This method can only work with tags that have attributes, therefore the amount of data that can be hidden is limited as compared with “White space in tag” method [4, 8].

2.6. White Space between Attributes

We can also insert spaces between the attributes of an element to hide data as explained in the following example:

\textbf{Example 6:}

Stego key:
\texttt{<font face=“verdana” size=“3”> for representing bit ‘0’}
\texttt{<font face=“verdana” size=“3”> for representing bit ‘1’}

Stego data:
\texttt{<font face=“verdana” size=“3”> 0}
\texttt{<font face=“Time New Roman” size=“2”> 1}
\texttt{<font face=“Tahoma” size=“3”> 1}
\texttt{<font face=“Arial” size=“3”> 0}

Embedded data: 0110

This method also limits the amount of data that can be hidden as it can only work with tags that have at least two attributes [4, 8].

2.7. White Space between Tags

This method can be used when there are more than one tags appearing on the same line in the tagged document. We can add a space between any two of the attributes as illustrated in the following example:

\textbf{Example 7:}

Stego key:
Stego data:
<table>
<thead>
<tr>
<th>Column1</th>
<th>Column2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Embedded data: 0110

This method does not depend on the type of tag or attribute present in it and therefore can be used to hide a lot of information. However, it requires two or more consecutive tags to be present on the same line [4, 8].

### 2.8. White Space After “=”

The attributes of an element have values affected by the symbol “=” and if we insert a space after “=” it does not change the rendering of the document, as explained with the help of the following example:

**Example 8:**

Stego key:
- `<font size="3">` to represent bit ‘0’
- `<font size= "3">` to represent bit ‘1’

Stego data:
- `<font size="3">` 0
- `<font size="3">` 1
- `<font size="3">` 0
- `<font size="3">` 0

Embedded data: 0100

Using this method, we can utilize many attributes in each tag to enhance the amount of secret data that can be hidden in the cover tagged document [4, 8].

### 3. CONCLUSION

Hiding information in tagged documents is the most challenging job because the structure of the tagged document makes it extremely hard to add or remove data from the original document without affecting the meaning and display of the document. However, by utilizing the features of the tagged documents, we can make small alterations to the structure of the document to hide data using various methods such as the order of attributes in each tag, the order of elements, white spaces in tags and white space between tags. These methods can be used separately or in combination to enhance the amount of data to be hidden. Further, these methods can be incorporated with cryptography to create a double security covert channel. Thus, making it more difficult for an attacker to break or steal the hidden message.

### 4. FUTURE SCOPE

In the future, these methods can also incorporate with cryptography such as Data Encryption Standard, Triple Data Encryption Standard, RSA to create a double security for information hiding. An attacker can break the key for cryptographic or can find out steganographic techniques but breaking the combination of both can be nearly impossible.

### REFERENCES