Steganography in HTML Based on Substitution of Script Codes

Fei Wang[1 2], Liusheng Huang[1 2], Haibo Miao[1 2], Zhili Chen[1 2] and Wei Yang[1 2]

[1]*NHPCC, Depart. of CS. & Tech., USTC, Hefei 230027, China*
[2]*Suzhou Institute for Advanced Study, USTC, Suzhou, 215123, China*

scuwf@mail.ustc.edu.cn

Abstract

The HTML document is a main kind of public resources in the Internet, while the research in the steganography and steganalysis of them are in the infancy. In this paper, we propose a novel steganography in HTML documents. Different from natural language text, HTML documents have plentiful script codes which can be utilized for covert transmission more easily than the common text. The syntax structures of programming languages are strict and precise, so to generate scripts that are different in text but strictly equivalent in lexeme is possible. In the steganography, based on the substitution of variables and function naming, covert bits are successfully embedded with no script errors. Because of the transparency and flexibility of script codes, the well-disguised stego-HTML document has a legitimate appearance that is imperceptible. In the experiment, the embedding efficiency of the technique is considerable, proving that the steganography is feasible and promising.

Keywords: *HTML document, script, steganography, substitution, text*

1. Introduction

Research in steganographic techniques is becoming a hot area in information security. The techniques intend to conceal the existence of covert messages by embedding bits into legitimate data implicitly. Traditional steganographic methods mainly focus on static media, such as text files, image files and audio files, and the text steganography is the oldest and the most common one. As text-based media, like e-mail, microblog and text message, are in skyrocketing rise in the Internet today, the amount of text data is increasing at an accelerating pace. Hence, the ubiquity of the digital text in people's lives in turn motivates the increased concerns on utilizing text media as covert channels for secret communications.

Early text steganography is based on the textual format. But such methods can easily be attacked by text format detection and unification. In recent years, many advanced tools exploiting the linguistic steganography have been worked out. These tools embed covert bits into text media in a proper manner in order to help the resulting stego-text have a natural-like appearance, so that the embedded messages can't be easily discovered at the first look except the intended receivers. With the help of many language elements, the text data generated by these tools can successfully evade logical detection techniques which only concentrate on the textual format. However, these artful tools also have some potential drawbacks that are accurately captured by some NLP-based (natural language processing) statistical detection techniques that can find out the statistical difference between the really natural text and the stego-text. Nowadays, with the development of social network service, browser-based game and electronic commerce, HTML documents are becoming attractive carriers for text steganography. Similar to the common text, many HTML documents contain natural language segments. But because HTML is a tag language, HTML files have tags and script

codes which can't be found in natural language text. Therefore, these additional elements can be regarded as particular features in HTML for steganography. At present, research in HTML steganography and steganalysis is in its infancy and the existing HTML steganographic methods are very few. Consequently, to devise a novel steganography in HTML is promising.

In this paper, we propose a steganography in HTML based on substitution of script codes. Different from natural language text, HTML documents have many script codes which can be utilized. The script is a kind of programming language text that has strict and precise syntax structures, which means that the substitution of scripts is much easier than natural language. Exploiting the JavaScript grammar in the variable declaration and function naming, covert bits are successfully encoded. In the steganography, the original HTML document and the stego-HTML document are strictly equivalent in lexeme, so there are no abnormal script errors caused by this technique. Owing to the transparency and the flexibility of the script codes, the well-disguised stego-HTML document has a legitimate appearance, achieving strong concealment.

The rest of this paper is organized as follows: We summarize the related work in Section 2. Section 3 introduces some important preliminaries. In Section 4, we discuss the steganography scheme. In Section 5, we show the transmission efficiency of the technique. Section 6 is the conclusion of this paper.

2. Related Work

HTML is similar to text, so HTML steganography can derive from text steganography. Text steganography can be divided into two categories: format-based steganography and linguistic steganography. In general, format-based steganography is easier to be implemented while linguistic steganography is more undetectable.

The format-based steganography has many typical methods. D. Zou and S. H. Low proposed text steganographic methods based on word or line shifting [2, 10]. By shifting two consecutive words, lines or paragraphs at proper distance, covert messages can be encoded. These methods were successfully detected by a statistical detection technique designed by L. Li [7]. The technique utilized two self-defined concepts, "neighbor difference" and "environmental equal", and three statistical classifiers to distinguish the covert from the normal, achieving a successful detection rate at 93.3% in PDF documents. In English, the case of the letters and the abbreviation of words can also be exploited for covert embedding. For instance, "TEXT" can be written as "text" and "I'm" can be written as "I am". Stegparty is a steganographic tool encoding secrets by the substitution of abbreviation and the modulation of letters' case. But this kind of tool was thwarted by statistical steganalysis invented by H. Miao [3]. The technique operates on a characteristic dictionary collected from natural text files in advance and the detection accuracy can be up to 90%.

The linguistic steganography is an artistic technique. The original linguistic steganography is called the acrostic poetry that just splits covert messages at the first letter of words or sentences. Another method substitutes selected words with synonyms for information hiding [6], and it is improved by X. Zheng according to the context elements [12]. Among the numerous linguistic techniques, NICETEXT, TEXTO, Markov-Chain-Based (MCB) and Translation-Based Steganography (TBS) are the most famous [13]. NICETEXT generates cover text by the mixture of word substitution and probabilistic context-free grammars (PCFG) [8].The system has a dictionary table and a template of writing style. The PCFG is used to produce the template. In addition, the template can also be created by simulating a sample text. Under the template, the dictionary is employed to generate artificial sentences that are similar to natural ones randomly [13]. TEXTO is a system for transforming pure ASCII data into English sentences [5]. It runs like a substitution-

based encryption process which replaces each ASCII symbol by an English word. For all the words in the stego-text, only the nouns, verbs, adjectives, and adverbs are utilized to fill in the preset sentence structures while others are ignored [13]. MCB treats cover text generation as a signal transmission process from a Markov signal source [11]. From a sample natural text, the approach establishes a state transfer chart for the Markov signal source. A part of state transfer chart with branches denoting covert bits is shown in Figure 1 [13]. According to covert messages, the system can produce proper natural-like text sentences. TBS is a new steganography in text, developing with the Machine Translation techniques. The system deploys several translation machines and a cover text. Each sentence in the text can be translated by these machines into the same language. This approach just chooses the translated sentence from a proper translation machine to encode covert bits and LiT is the first TBS implementation proposed by C. Grothoff [1]. Nevertheless, because the stego-text doesn't have natural, coherent and complete sense, which can be detected by a human warden, the tools listed above are vulnerable to advanced statistical steganalysis. Z. Yu proposed a detection method using context information to attack the substitution of synonyms [14]. The steganalysis exploits the inverse document frequency (IDF) to evaluate the suitability of words in order to distinguish common words and rare ones. The detection accuracy in this approach is above 90.0%. Z. Chen devised a steganalysis to frustrate NICETEXT, TEXTO and MCB [13]. The approach utilizes the statistical characteristics of correlations between words and the N Window Mutual Information to separate stego-text. The accuracies against NICETEXT, TEXTO and MCB are 98.48%, 97.96% and 94.01% respectively. P. Meng designed a steganalysis based on the statistical language model, which successfully discovered TBS [9]. The approach employs a 12-dimension feature vector to evaluate the stego-text, achieving an accuracy rate above 85%.

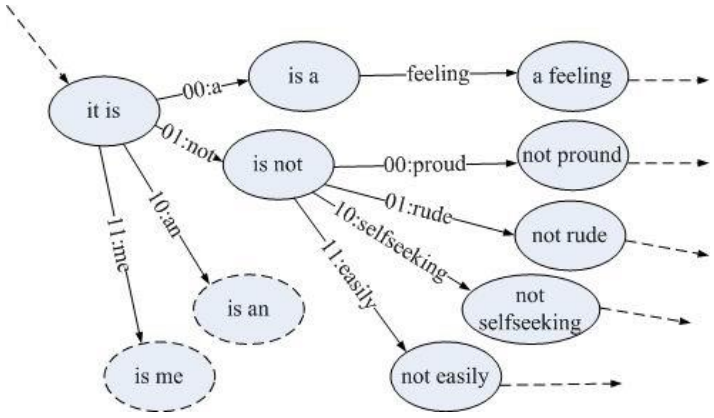


Figure 1. A part of Markov Transfer Chart

So far, research in HTML steganography is very limited. Since WEB applications are insensitive to the format of attributes in tags, like the case, the shifting, and the order, most HTML steganographic tools are format-based, which derives from existing text steganography. Deogol is a typical one which just reorders the attributes in tags for covert encoding [4]. But as above arguments expound, the format-based steganography and the linguistic steganography are actually vulnerable and detectable. What's more, as a kind of public resources, HTML documents should not be unreadable, which means it can't have meaningless natural language segments. Therefore, current text steganography is not very suitable for HTML documents.

3. Preliminaries

3.1. Important Notions

HTML Document. As mentioned above, HTML is a kind of tag language. Additionally, script codes can also be embedded into it. Consequently, HTML documents consist of pure literal text, tag text and script text. The following is a fragment of an HTML document. Line 2 to Line 4 is a piece of script code. The string "This is the homepage of ABC company" in Line 7 is the pure literal text and all the segments in the angle brackets belong to the tag text.

```
1  <script type="text/javascript">
2  function test () {
3  var str="This is a sample for script codes";
4  alert (str); }
5  </script>
6  <input type="button" value="A sample" onclick="test ()">
7  <a href="http://www.abc.com">This is the homepage of ABC company</a>
```

Script Legitimacy. To WEB clients, script codes in HTML documents are almost all written in JavaScript language. The codes can help the webpage do some front-end processing for user data and bring friendly user experience to visitors. Actually, script codes are not directly visible to normal visitors. People who are using WEB applications, such as Internet browsers, can't know what the codes exactly do and what commands the codes exactly execute. Behaviors of script codes are transparent and they can't be observed from the appearance of HTML documents. Hence, script is a common vulnerability of HTML and the codes are scrutinized by WEB applications to dig potential attacks. Although scripts are under surveillance, if there are no malicious intrusion features in them, they will not be intercepted and be considered as legitimate ones. In practical, nearly all the scripts in HTML documents from normal websites are well-meaning and legitimate.

Script Flexibility. Scripts are written by different programmers, so script characteristics differ from document to document. The coding styles are flexible in different script segments. Even in the same HTML document, scripts change frequently, especially in some minisites under construction. We collect the same 1000 HTML documents in 10 days, once a day, and compare the scripts. The result is in Figure 2. As the figure shows, the proportion of the documents whose scripts are invariable is less than 35%, which fully proves the flexibility of scripts. Therefore, because the modification in script codes is not suspicious, the flexibility can be utilized for steganography.

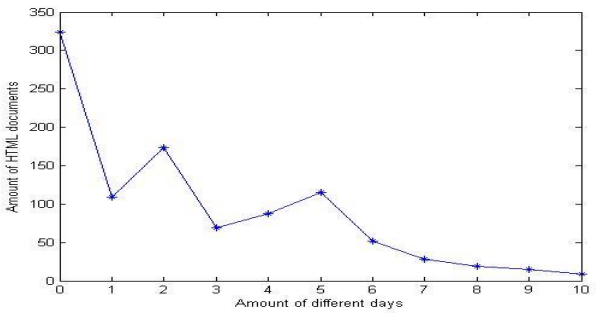


Figure 2. Script Flexibility in the same HTML Document

3.2. JavaScript Grammar

The flexibility of scripts is partly caused by the diversity of the language grammar. Hence, we'll introduce three important JavaScript grammar points which will be exploited in the HTML steganography.

Action Scope. The concept is designed to restrict the scopes of names used in codes. Each name has its own action scope where it is declared. The scope is denoted by a pair of braces "{" and "}". Especially, the scope outside all the brace pairs is the global scope and the names in it are valid in all the scopes. It should be noted that, in programming codes, the relationship between two scopes has only two classes: inclusive and disjoint, as the following expression.

$$\text{scope relationship}\begin{cases} scope_1 \subset scope_2 \ \text{ OR } \ scope_1 \supset scope_2 \\ scope_1 \cap scope_2 = \varnothing \end{cases} \tag{1}$$

Variable Declaration. The same as other programming languages, there are two types of variables in JavaScript: global variables and local variables. For a scope, the variables outside it are the global and in it are the local. All the variables should be declared before defined. In JavaScript, local variables should be declared definitely after the keyword *var* while global ones are not necessary. However, in practical HTML documents, because of the requirement in coding standardization, all the global variables are declared after *var*. Additionally, there is another particular format of the local: formal parameters. They are declared in the brackets of a function declaration, and there mustn't be any keyword before them. The duplication of names between the two types is allowed. If the duplication happens in an action scope, the priority relationship of them follows the proximity principle:

$$P_{local} > P_{global} \tag{2}$$

In the following fragment, the global variables *a*, *b* and *y* are covered by the local variables *a*, *b*, and *y*, so the values of the global variables are not changed and the result is 70.

```
1   <script type="text/javascript">
2   var a = 100;
3   var b = 50;
4   var y = 20;
5    function f ( a, b ) {
6         var y = 10;
7         a = a * y;
8         b = b * y; }
9    f ( a, b );
10  alert (a – b + y);
11  </script>
```

Function Naming. The function is another important element in JavaScript. Functions can be divided into two categories: named functions and unnamed functions. The following Fragment 1 defines a named function and Fragment 2 defines an unnamed function assigned to a variable. However, Fragment 2 is incorrect because an unnamed function can't be used before defined while a named one is permitted. If the function is used after its definition, there will be no difference between the two fragments. In general, functions in HTML documents are mainly used in some controls outside script domains and script domains are relatively independent of each other. Therefore, in order to manage the documents effectively and

maintain the script readability, no functions are used before defined in practical HTML webpages. In other words, the methods of function definition in Fragment 1 and Fragment 2 are strictly equivalent and error-free in HTML documents.

	Fragment 1	Fragment 2
1	<script type="text/javascript">	<script type="text/javascript">
2	a ();	a ();
3	function a () {	var a = function () {
4	var b = 2;	var b = 2;
5	alert (b); }	alert (b); }
6	a ();	a ();
7	</script>	</script>

3.3. Adversary Model

The imaginary adversary of the HTML steganography is a WEB filter. The filter is a passive detection system, so it only monitors the traffic but doesn't take any attacks by alteration. The covert receiver is in a LAN and all the ingoing HTML documents will be detected by the filter. The filter only checks scripts primarily in order to discover threatening codes. Because of the flexibility of scripts, the filter won't take an in-depth inspection of the characteristics in programming styles. Since, as mentioned above, scripts change frequently, the modification of script codes in the same HTML document is not suspicious. Altogether, as long as the scripts have no malicious behaviors, they will be identified as legitimate and bypass the filter.

4. HTML Steganography Scheme

Although an adversary won't inspect script codes deeply, it is unwise to attach secret information to scripts directly. For this reason, in the HTML steganography, we utilize the grammar pointed out in Section 3.2 to design two advanced techniques based on the substitution of script codes.

4.1. Substitution of Variables

There must be many variables in scripts and the names of them are multifarious. The names of the variables in codes have no semantic meaning and the naming conventions vary with programmers. Therefore, the difference in names can be modulated for steganography. The following is a sample. In others' sight, there is no difference between the codes, but the variables are modulated to encode covert bits.

	Original Script	Stego-Script
1	<script type="text/javascript">	<script type="text/javascript">
3	function f () {	function f () {
4	var a = 2;	var x = 2;
5	var b = 3;	var y = 3;
6	alert (a + b); }	alert (x + y); }
7	f ();	f ();
8	</script>	</script>

Specifically, first, we collect massive HTML documents in public and obtain the variables' names in the scripts. Then, because each name is a string, we utilize a string hash function to split the names into two classes, one with the odd hashed values and the other with the even.

Then, for a covert binary bit 1, a selected variable's name is substituted by an odd-hashed name, and an even-hashed one for a bit 0. Last, covert messages are extracted by the same string hash function. In this technique, the sender and the receiver only need to share a string hash function, which is easily implemented. The outline of the algorithm is as follows.

Step1. The sender recognizes all the variables in scripts with the keyword *var* and *function*. Then, the sender ascertains the action scope of each variable with the scope marks "{" and "}". The variables' names are stored in order. Exceptionally, the variables recognized do not include the variables pointing to functions shown in Section 3.2 because the functions are usually used outside the script domains.

Step2. For each variable, in order, if current cover bit is 1, the name of the variable will be randomly substituted by an odd-hashed one collected in advance and by an even-hashed one for a bit 0. If the original name of the variable meets the requirement of the hashed value, the variable won't be substituted. If variables are in the same action scope, the new names of them should be different from each other. If all the covert bits have been encoded, the sender will encode 8 consecutive 0 bits to denote the end.

Step3. For each substituted variable, replace all the original names with the new names in its action scope. The replacement is very important and tricky because the duplication of names in different scopes may cause unexpected script errors. As the formula (1) shows, the relationship between two different scopes is inclusive or disjoint. Therefore, we build a multifurcating tree to denote the correlation in scopes, illustrated in Figure 3. All the child nodes of the same father node are disjoint and included by their father nodes. We adopt a bottom-up approach in the replacement, based on the formula (2). First, we replace the variables' names in the leaf nodes. Then, do the replacement up layer-by-layer, to the root, global scope. The replacement in any non-leaf nodes shouldn't be executed until all of its child nodes are finished. Utilizing this method which exactly matches the proximity principle, the names can be substituted with no script errors. During the process, we additionally maintain a signal at each position which should be replaced. By the signals, if a name is replaced, it won't be replaced again. In this way, even if the duplication of names happens in the replacement of different scopes, script errors will be avoided.

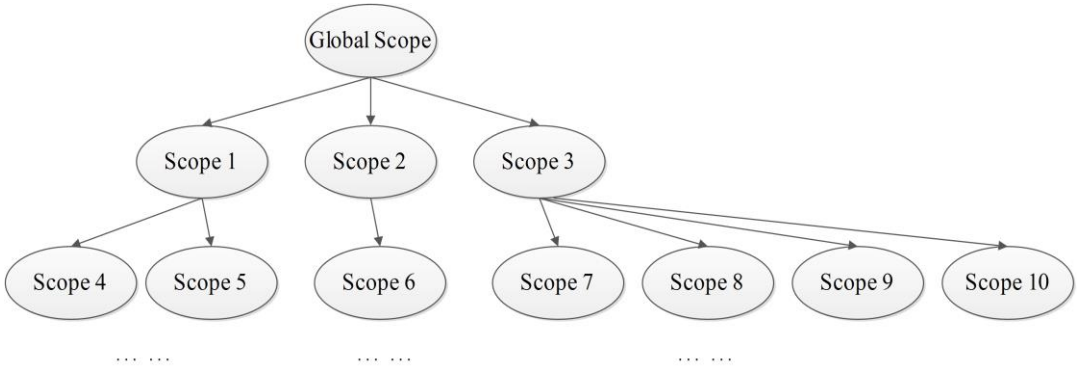


Figure 3. Tree Structure of the Scopes

Step4. The receiver recognizes all the variables and extracts the covert message by byte using the same string hash function. If the receiver obtains a byte whose value is 0, the extraction process will be terminated and the covert message will be integrated.

4.2. Substitution of Function Naming

As mentioned in Section 3.2, the two naming methods for functions are strictly equivalent in HTML documents. Hence, the difference of them can be exploited for steganography. Specifically, first, we recognize all the functions by the keyword *function*. Then, for a covert bit 1, the selected function is defined as a named function, and an unnamed function for a bit 0. The outline of the algorithm is in the following.

Step1. The sender recognizes all the functions in scripts with the keyword *function*. The functions are stored in order.

Step2. For each function, in order, if current covert bit is 1, the function will be defined as a named one and as an unnamed one for a bit 0. If all the covert bits have been encoded, the sender will encode 8 consecutive 0 bits to denote the end.

Step3. The receiver recognizes all the functions and extracts the covert message by byte using the same rule. If the receiver obtains a byte whose value is 0, the extraction process will be terminated and the covert message will be integrated.

4.3. Implementation Scheme

There are two kinds of steganography: active and passive. In general the passive are more undetectable, but less efficient. The active generate additional data for encoding while the passive exploit the existing. The HTML steganography proposed in this paper can be both active and passive. In the HTML steganography, there are a covert sender and receiver. The receiver is disguised in an HTTP client and the sender in a WEB server. The server can be a public one which can be visited by other normal HTTP users. Figure 4 illustrates the working processes of the scheme.

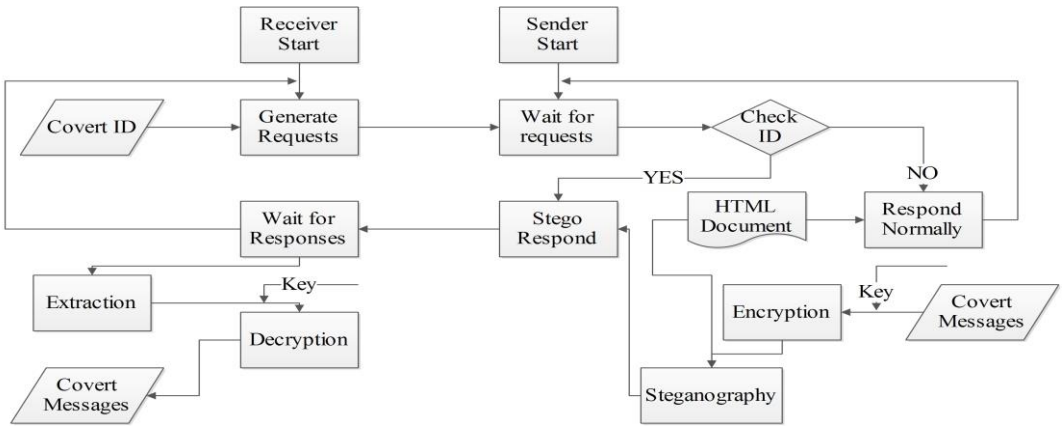


Figure 4. The Flow Diagram of the Implementation Scheme

As the figure shows, the HTML steganography can be completed in the following 3 steps.

Step1. The receiver sends an HTTP request with its authentication ID to the sender.

Step2. The sender receives the request and checks the ID. If the receiver is identified as covert, the sender will respond the request with the stego-HTML document which encodes the encrypted covert messages. Otherwise, the sender will respond the request with the original HTML document as a normal WEB server.

Step3. The receiver extracts data from the stego-HTML document and decrypts the data to obtain covert messages. By the way, if the encryption and decryption of covert messages are symmetric, the key will be shared between the sender and the receiver, but if they are not, the public key and the private key will be allocated to the sender and receiver respectively.

5. Experiment

In the experiment, employing the two substitution techniques, we test the embedding capacity in the scripts of several homepages of famous international websites. Table 1 shows the results. The results illustrate that the embedding capacity is considerable, which proves that the steganography is feasible and promising.

Table 1. Embedding Capacity of the Steganography

Homepage	Embedding Quantity (Byte)
www.google.com.hk	353
www.youku.com	196
www.sina.com.cn	117
www.microsoft.com	568
www.ebay.com	331
www.baidu.com	184

Besides, in our campus network, we transmit three trivial files: a 10.21 KB TEXT file, a 9.63KB WMV file and a 15.84KB PNG file, for a further experiment. The receiver is a common personal computer in our department LAN and the sender is a mirror of a WEB server in our college website. The transmission rates are listed in Table 2. The unit of the rate is Bph (Byte per HTML document). As the results show, the transmission rates are much lower than that of famous websites' homepage, because the webpages of our college website are very simple and there are not abundant script resources. Therefore, if we want to improve the steganographic performance, we should utilize webpages with big size of script codes.

Table 2. Transmission Rates for Files

File	Transmission Rate (Bph)
TEXT	29.3
WMV	18.7
PNG	22.8

6. Conclusion

In this paper, we devise a HTML steganography based on the substitution of script codes. Exploiting the JavaScript grammar in the variable declaration and function naming, covert bits are successfully encoded. In the steganography, the original HTML document and the stego-HTML document are strictly equivalent, so there are no abnormal script errors caused by this technique. Due to the transparency and the flexibility of the script codes, the well-disguised stego-HTML document has a legitimate appearance, achieving strong imperceptibility. Because the existing research

in the steganography and steganalysis of HTML documents is very limited, we can't guarantee that it mustn't be defeated by some statistical detection systems in the future.

The script is a peculiar element of HTML documents. Different from normal natural language text, the script is a kind of programming language text which has strict and precise syntax structures, which means that the equivalent substitution of scripts is much easier than natural language. Therefore, some more intelligent techniques, based on the semantically equivalent code generation, for HTML steganography are promising in the following work.

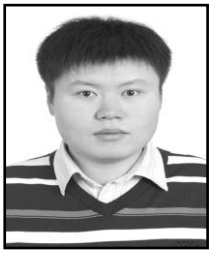
Acknowledgements

This paper was partially supported by the National Natural Science Foundation of China (Nos.60903217 & 61202407), the Fundamental Research Funds for the Central Universities (Nos.WK0110000027 & WK0110000033), the Natural Science Foundation of Jiangsu Province of China (No.BK2011357), the Guangdong Province Strategic Cooperation Project with the Chinese Academy of Sciences (No.2012B090400013), and the Scientific and Technical Plan of Suzhou (No. SYG201010).

References

[1] C. Grothoff, K. Grothoff, L. Alkhutova, R. Stutsman and M. J. Atallah, "Translation-based steganography", Information Hiding Conference, Barcelona, Spain, **(2005)** June 6-8.
[2] D. Zou and Y. Shi, "Formatted text document data hiding robust to printing, copying and scanning", IEEE International Symposium on Circuits and Systems, Kobe, Japan, **(2005)** May 23-26.
[3] H. Miao, L. Huang, Z. Chen and W. Yang, "Analysis and detection of text steganographic tool-Stegparty", Journal on Communications of China, vol. 31, no. 9A, **(2010)** September, pp. 251-258.
[4] Introduction to Deogol. http://hord.ca/projects/deogol/intro.html, **(2013)**.
[5] K. Maher, TEXTO. ftp://ftp.funet.fi/pub/crypt/steganography/texto.tar.gz, **(2013)**.
[6] K. Winsten, "Lexical steganography through adaptive modulation of the word choice hash", http://alumni.imsa.edu/~keithw/tlex/lsteg.ps, **(2013)**.
[7] L. Li, L. Huang, X. Zhao, W. Yang and Z. Chen, "A statistical attack on a kind of word-shift text-steganography", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, **(2008)** August 15-17.
[8] M. Chapman, "Hiding the Hidden: A software system for concealing ciphertext as innocuous text", http://www.NICETEXT.com/NICETEXT/doc/thesis.pdf, **(2013)**.
[9] P. Meng, L. Hang, Z. Chen, Y. Hu and W. Yang, "STBS: A statistical algorithm for steganalysis of translation-based steganography", Information Hiding Conference, Calgary, Canada, **(2010)** June 28-30.
[10] S. H. Low, N. F. Maxemchuk, J. T. Brassil and L. O'Gorman, "Document marking and identification using both line and word shifting", IEEE Conference on Computer Communications, Boston, USA, **(1995)** April 2-6.
[11] S. Wu and L. Huang, "Research on information hiding", Master Thesis, University of Science and Technology of China, **(2003)**.
[12] X. Zheng, L. Huang, Z. Chen, Z. Yu and W. Yang, "Hiding information by context-based synonym substitution", International Workshop on Digital Watermarking, Guildford, UK, **(2009)** August 24-26.
[13] Z. Chen, L. Huang, Z. Yu, W. Yang, L. Li, X. Zheng and X. Zhao, "Linguistic steganography detection using statistical characteristics of correlations between words", Information Hiding Conference, Santa Barbara, USA, **(2008)** May 19-21.
[14] Z. Yu, L. Huang, Z. Chen, L. Li, X. Zhao and Y. Zhu, "Steganalysis of synonym-substitution based natural language watermarking", International Journal of Multimedia and Ubiquitous Engineering, vol. 4, no. 2, **(2009)** April, pp. 21-34.

Authors



Fei Wang is now undertaking a Master degree in Computer Science, University of Science and Technology of China, Hefei, Anhui, China. His areas of interests include information hiding, steganography, steganalysis, covert channel and traffic classification. He can be contacted by e-mail at scuwf@mail.ustc.edu.cn.

Liusheng Huang is a professor in the department of Computer Science, University of Science and Technology of China, Hefei, Anhui, China. His research areas include wireless networks, sensor networks, privacy preservation, information hiding, distributed computation, parallel computation and ubiquitous computation.

Haibo Miao is a Ph.D. candidate in Computer Science, University of Science and Technology of China, Anhui, China, Hefei. His areas of interests include information hiding and multimedia security. He can be contacted by e-mail at mhb@mail.ustc.edu.cn.

Zhili Chen received his Ph.D. in Computer Science, University of Science and Technology of China, Anhui, China, Hefei. He is in a postdoctoral study now. His areas of interests include information hiding and privacy preservation.

Wei Yang received his Ph.D. in Computer Science, University of Science and Technology of China, Anhui, China, Hefei. He is in a postdoctoral study now. His areas of interests include quantum cryptography and privacy preservation.