Multimed Tools Appl (2014) 71:1893–1911
DOI 10.1007/s11042-012-1313-8

Linguistic steganalysis using the features derived from synonym frequency

Lingyun Xiang · Xingming Sun · Gang Luo · Bin Xia

Published online: 18 December 2012

Abstract A linguistic steganalysis method is proposed to detect synonym substitution-based steganography, which embeds secret message into a text by substituting words with their synonyms. First, attribute pair of a synonym is introduced to represent its position in an ordered synonym set sorting in descending frequency order and the number of its synonyms. As a result of synonym substitutions, the number of high frequency attribute pairs may be reduced while the number of low frequency attribute pairs would be increased. By theoretically analyzing the changes of the statistical characteristics of attribute pairs caused by SS steganography, a feature vector based on the difference of the relative frequencies of different attribute pairs is utilized to detect the secret message. Finally, the impact on the extracted feature vector caused by synonym coding strategies is analyzed. Experimental results demonstrate that the proposed linguistic steganalysis method can achieve better detection performance than previous methods.

Keywords Steganalysis · Steganography · Linguistic steganalysis · Support vector machine (SVM) · Synonym substitution (SS)

L. Xiang
College of Computer and Communication Engineering, Changsha University of Science & Technology,
Changsha, Hunan 410004, China
e-mail: suhong210@yahoo.com.cn

X. Sun (✉)
Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science
and Technology, Nanjing 210044, China
e-mail: sunnudt@163.com

G. Luo · B. Xia
College of Information Science and Engineering, Hunan University, Changsha 410082, China

G. Luo
e-mail: luog@yahoo.cn

B. Xia
e-mail: xnby@foxmail.com

1 Introduction

Linguistic steganography is the technology to hide secret messages into an innocuous-looking cover text by using natural language processing techniques to achieve the goal of covert communication. With the prevalence of digital texts, including novels, news, office documents, blogs, etc., linguistic steganography has attracted increasing interest during the past few years. Accordingly, the opposite technology—linguistic steganalysis also has attracted the attentions of researchers. Linguistic steganalysis aims at discovering the presence of secret messages in digital texts. It can be used to find and even prevent covert communication established by terrorists or illegal groups.

Linguistic steganography mainly falls into two categories. The first category is directly to generate a new natural-looking text [4, 11] by certain rules based on mimicking technique. The generated texts are usually not only obscurely readable and understood, but also easy to be distinguished from natural texts by statistical analysis [5]. The other embeds messages by approximately meaning-preserving linguistic modifications, such as synonym substitution (SS) [2, 6, 10, 14, 15, 17, 19, 22], syntactic transformation [1, 13], etc., to the natural texts. Among the existing linguistic steganography methods, SS steganography is an attractive one because of its simplicity, high embedding capacity and good imperceptibility. And some improved variants of SS steganography have been proposed to prevent possible wrong syntax and semantics caused by SS [2, 10, 14, 15, 17]. For example, Bolshakov [2] previously tested relative synonyms for semantic compatibility with collocations to determine whether synonym substitutions were correct, and replaced absolute synonyms directly. Topkara et al. [17] just chose one alternative for every synonym to be replaced to hide one bit message. Liu et al. [10] used the disambiguation function to determine which synonym was right to substitute the original word. Muhammad et al. [14] only adopted two absolute synonyms of each synset[1] for embedding messages so that the replaceable synonyms are limited in a narrow scope to obtain good imperceptibility. Shirali-Shahreza et al. [15] used the English words which have the same meanings but different spellings in British and American English to camouflage data. In addition, Yang et al. [22] and Chiang et al. [6] cooperated the SS with some auxiliary techniques to design text watermarking schemes.

In recent years, some works on linguistic steganalysis have been done for breaking the SS steganography [12, 16, 23]. The first work on linguistic steganalysis against SS steganography was published by Taskiran et al. [16]. In this work, a feature vector extracted from a 3-gram language model is used to distinguish the steganographically modified sentences from the unmodified ones by SVM. However, it neither accurately detected stego sentences nor determined whether a text was a cover or stego one. In addition, Yu et al. [23] constructed a detector based on characteristics of the evaluated suitability of synonyms for their context in a text. This detector provided reliable results when the embedding rates of stego texts were very high. However, it has to access Google frequently, which leads to a very low running speed because Google does not allow automated frequent queries [7]. Luo et al. [12] extracted a statistical feature from the synonym sequences, each of which was composed of synonymous words appearing in the text, to detect the stego texts generated by SS steganography. In spite of running fast, the detection accuracy was not desirable especially when detecting stego texts with lower embedding capacity. Additionally, above steganalysis methods all do not consider the two critical factors affecting detection accuracy for SS steganography—the embedding rate and the synonym database.

[1] Synset is defined as a set of words with identical or similar meanings

In order to improve the performance of previous steganalysis for SS steganography, a new linguistic steganalysis method is developed in this paper. As a result of SS steganography, the number of high frequency synonyms may be reduced while that of low frequency synonyms would be increased. These changes can produce convincing evidences to reveal the existence of the hidden message. Thus, attribute pairs defined based on synonyms' frequencies and synsets' sizes are used to capture these changes, and some statistical features are extracted from the distribution of attribute pairs to form a vector for SVM to discriminate stego and cover texts. One advantage of employing the attribute pair in this paper is that relationships between statistical characteristics in cover and corresponding stego texts can be formulized to correlate with the embedding rate. The experimental results verify the efficacy of the proposed steganalysis method for different embedding rates, SS steganographic tools, and synonym databases, and demonstrate that the proposed method has more significant advantages than the existing methods.

This paper is organized as follows. Section 2 briefly reviews the general process of the SS steganography, followed by introducing the main synonym coding strategies. In Section 3, the characteristics of the attribute pairs in cover texts and stego texts are analyzed, and the linguistic steganalysis against SS steganography based on attribute pairs is described. The impact on the extracted feature vector caused by coding strategy is investigated in Section 4. Experimental results are presented in Section 5. Section 6 concludes the paper.

2 Overview of the SS steganography

Early SS steganography just simply replaces words with their synonyms to embed a message in a known text, so that the meaning of the cover text, in theory, should not be significantly changed. The stego text looks like innocuous one in appearance. In general, the synsets in a prepared synonym database for SS steganography are pre-encoded by a coding strategy. Each used synonym must be encoded into unique values in order to represent different information.

In the embedding process, firstly, SS steganography recognizes the words with synonyms, and locates the corresponding synsets; then, according the embedded message, it chooses the synonym with appointed code to substitute the original word. Figure 1 illustrates how to embed messages into two given sentences by SS steganography. In the first sentence, "rebuke" is recognized as a synonym, and it is located in the synset {rebuke, reproof}, which

$$\textit{A hungry lion will not stick at a trifle, whereas a full one will flee}$$

$$\textit{at a very small} \left\{ \begin{array}{l} \textit{rebuke} \\ 0:\ \textit{rebuke} \\ 1:\ \textit{reproof} \end{array} \right\}.$$

$$A \left\{ \begin{array}{l} \textit{onetime} \\ 0:\ \textit{erstwhile} \\ 1:\ \textit{onetime} \\ ?:\ \textit{quondam} \end{array} \right\} \textit{college drinking game has turned into some serious business.}$$

Fig. 1 Embedding messages into two example sentences by SS steganography

is encodes as {0: rebuke, 1: reproof}. If the current embedded message is the bit “1”, then “rebuke” will be replaced by “reproof”, otherwise, no substitution will be made. In the second sentence, “onetime” is located in a synset which contains three synonyms. Different coding strategies may lead to completely different coding results, for example, “quondam” can be encoded as “01” or nothing. Three typical coding strategies in previous SS steganography will be introduced.

1) The basic coding strategy. This is the simplest coding strategy, and can be described as follows: assuming a synset has 2^q+m (integers $q>0$, $m\geq 0$) synonyms, then arbitrarily 2^q synonyms from it can be selected to be encoded as q-bit strings.
2) The multi-base coding strategy [2, 19]. It sets the synsets containing the orderly appearing synonyms in a text to $sn_0, sn_1, \ldots, sn_n$ with sizes $k_0, k_1, \ldots, k_n$, respectively, then codes the secret message by the following steps:

   Step1. Convert the secret message into an integer M;
   Step2. Encode synonyms in synset sn_i as integer from 0 to k_i-1;
   Step3. For $i=0$ to n, repeatedly calculate $t_i = M \bmod k_i$, $M' = M/k_i$, $M = M'$, and select the synonym with codeword t_i from sn_i to replace the original synonym in the cover text.

3) The binary tree-based coding strategy [6]. It constructs a binary tree for each synset with the synonyms as the leaves. Different bits (“0” or “1”) are assigned to different branches, and then each synonym in the same synset will obtain a unique codeword. A complete binary tree, Huffman tree, or a normal binary tree may be constructed. The construction principle of the binary tree is determined by concrete coding strategy.

In fact, for most synsets, their synonyms have only part of the same senses, and then substitutions may lead to meaning distortions, sentence syntactic structure errors. The SS steganography proceeds with the works on how to select a perfect word to replace the original synonym for messages embedding [2, 10, 14, 15, 17]. These works used collocations, context information, etc., to measure the suitability of synonym substitutions.

3 Linguistic steganalysis against SS steganography by attribute pairs analysis

3.1 Synonym attributes representation

Synonym substitutions would keep semantic attributes of synonyms in a text almost unchanged. However, some statistical attributes will undoubtedly be modified. Thus, it is necessary to present an outstanding representation method to describe these statistical attributes for steganalysis.

In the existing natural language processing area, synonymous words always have different frequencies in a huge corpus. With these in mind, synsets in a synonym database could be preprocessed into synonym vectors according to the frequencies of the inside synonyms. The definition of synonym vector is given as the definition 1.

Definition 1 A synonym vector is defined as an ordered synset sorted in descending order of the frequencies of the inside synonyms.

A synonym vector $(w_0, \ldots, w_{k-1})$ satisfies the conditions: $F(w_{i-1}) \geq F(w_i)$, $M(w_{i-1}) \approx M(w_i)$, where $i = 1, \ldots, k-1$, $F(w)$ denotes the frequency of synonym w derived from a huge corpus, $M(w)$ represents a lexical concept of w. Each synonym will have a unique position in the synonym vector, whose dimension is definite. Thus, some inherent attributes of a synonym can be obtained, and a new term named attribute pair is introduced to represent them.

Definition 2 Attribute pair of a synonym is defined as its position in a synonym vector and the dimension of the synonym vector, denoted as an ordered pair <*pos*, *dim*>, where $pos \in \{0, 1, \ldots, dim - 1\}$.

Suppose that synonym w is located in a synonym vector $(w_0, \ldots, w_{k-1})$, if $w = w_j$, $j \in \{0, 1, \ldots, k - 1\}$, then its attribute pair is <*j*, *k*>.

3.2 Statistical characteristics of attribute pairs

Give an attribute pair <*j*, *k*>, its relative frequency $p(j, k)$in a text is given by

$$p(j,k) = \frac{f(j,k)}{\sum\limits_{i=0}^{k-1} f(i,k)} \tag{1}$$

where $f(j, k)$ is the number of total occurrences of <*j*, *k*> in the text, $\sum\limits_{i=0}^{k-1} f(i,k)$ represents the number of total occurrences of all attribute pairs whose second components equal to k in the text.

According to definition 1, for any synonym vector $(w_0, \ldots, w_{k-1})$, when $j < h$, $h \in \{1, \ldots, k - 1\}$, the frequency of w_j is larger than or equal to that of w_h. Thus, usually the probability of w_j occurring in a cover text is not less than that of w_h. In fact, in most synonym vectors, synonyms with different attribute pairs have unequal frequencies. The cover text would contain more synonyms with attribute pair <*j*,*k*> than the ones with attribute pair <*h*, *k*>. Consequently,

$$f_c(j,k) > f_c(h,k), \quad j < h \tag{2}$$

$$p_c(j,k) - p_c(h,k) > 0, \quad j < h \tag{3}$$

where $f_c(j, k)$ and $p_c(j, k)$ represent the number of total occurrences and relative frequency of <*j*, *k*> in cover text, respectively.

Therefore,

$$Max(f_c(j,k)) = f_c(0,k) \tag{4}$$

$$Min(f_c(j,k)) = f_c(k-1,k) \tag{5}$$

In existing steganographic algorithms, synonyms in a synset are always stored or encoded in alphabetical order, just like the tool Tyrannosaurus lex [20] (Tlex for short). However, after processing synsets used by SS steganography into synonym vectors, the order of synonyms in a synonym vector is not always synchronous with the encoding order. Thus, for arbitrary synonyms being encoded as the same specified codeword, the values of their attribute pairs are random. In other words, in a stego text, if a synonym *w* contains a secret message and its attribute pair is <*pos*, *k*>, *pos* may be a random value varying from 0 to *k*−1. For all synonyms with secret messages, whose attribute pairs are <*pos*, *k*>, the proportion of synonyms with attribute pair <*j*, *k*> is 1/*k*. Since both a synonym and its substituted one come from the same synset, SS steganography just causes the transitions between attribute pairs having the same second components, leading to $\sum_{i=0}^{k-1} f_s(i,k) = \sum_{i=0}^{k-1} f_c(i,k)$, where $f_s(j, k)$ represents the number of total occurrences of attribute pair <*j*, *k*> in stego text. Finally, the following equations can be deduced:

$$f_s(j,k) = (1-r)f_c(j,k) + \frac{1}{k} r \sum_{i=0}^{k-1} f_c(i,k) \tag{6}$$

$$p_s(j,k) - p_s(h,k) = \frac{f_s(j,k) - f_s(h,k)}{\sum_{i=0}^{k-1} f_s(i,k)} = (1-r)(p_c(j,k) - p_c(h,k)) \tag{7}$$

where $p_s(j, k)$ represents the relative frequency of attribute pair <*j*, *k*> in stego text, *r* is the embedding rate. As a matter of convenience, *r* is measured by the ratio of total number of synonyms containing secret messages to total number of synonyms appearing in a text.

It is noteworthy that Eq. (7) is obtained under the assumption that the above analysis used the same synonym database with the one in SS steganography.

As $0<r\leq 1$, thus

$$p_s(j,k) - p_s(h,k) < p_c(j,k) - p_c(h,k), \quad j < h \tag{8}$$

The difference between $p_s(j, k)–p_s(h, k)$ and $p_c(j, k)–p_c(h, k)$ depends on the value of *r* and $p_c(j, k)–p_c(h, k)$. The larger $p_c(j, k)–p_c(h, k)$ is, the larger the difference is for the same *r*. *r* becomes larger, $p_s(j, k)–p_s(h, k)$ becomes closer to 0, the difference becomes larger. When *r*=1, the difference achieves the largest value.

Figure 2 shows the statistical distributions of *p*(0, 2)–*p*(1, 2), *p*(0, 3)–*p*(1, 3), *p*(0, 3)–*p*(2, 3) of cover and stego texts. 5,622 cover texts were used and denoted as the ones with embedding rate 0% in Fig. 2. For each specified embedding rate, 5,622 stego texts were generated by the SS steganographic tool Tlex, whose source code was slightly modified so that message can be embedded with multi-base coding strategy for any embedding rate.

In Fig. 2, it can be observed that cover texts have higher values of *p*(*j*, *k*)–*p*(*h*, *k*), in comparison to stego texts. Especially, it is easy to differentiate stego texts with high

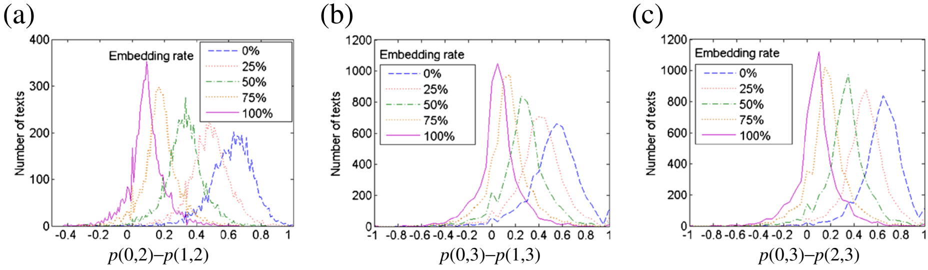

Fig. 2 The statistical distributions of *p*(*j*,*k*)–*p*(*h*,*k*) for different attribute pairs

embedding rates from cover texts by the values of $p(j, k)–p(h, k)$, which are concentrated in different ranges. Hence, the statistics $p(j, k)–p(h, k)$ extracted from the relative frequencies of attribute pairs can be taken as a clue to detect secret message existing in synonyms of texts.

3.3 The proposed linguistic steganalysis

The details of the proposed steganalysis using the statistical features derived from distributions of attribute pairs are as follows.

Step1. Preprocess the collected synonym database to convert synsets into synonym vectors.
Step2. Traverse every synonym in the text to obtain the value of its attribute pair.
Step3. Calculate the values of $p(j, k)–p(h, k), j < h, j, h \in \{0, 1, \ldots, k–1\}$ to form a feature vector.
Step4. Train an SVM with the above feature vector as an input.
Step5. Use the trained SVM to classify stego texts from cover texts.

In general, only synonymous words with the closest senses could be replaced with each other, and each synonym can only exist in a synset for SS steganography. Thus, not all the synonyms of a word listed in every sense are used by steganography. The synset size is always small. The mean synset size of the synonym database used in Tlex is 2.32 words after deleting repeated synsets. And for the synonym database extracted from Wordnet 2.1 [21] and used in our experiments, the mean synset size is 2.53 words. Synonyms in a text recognized by these synonym databases are always coming from small synsets rather than large synsets. In other words, the larger k is, the smaller $f(j, k)$ is. What is more, for a short text, the value of $f(j, k)$ may be very small and even zero when k is large. Under this condition, only small k should be selected to extract informative feature vector. As mentioned above, a larger $p_c(j, k)–p_c(h, k)$ is beneficial for differentiating $p_s(j, k)–p_s(h, k)$ from $p_c(j, k)–p_c(h, k)$. In order to maximize $p_c(j, k)–p_c(h, k)$, j should be set to zero, since $Max(f_c(j,k)) = f_c(0,k)$. In conclusion, only the statistical characteristics $p(j, k)–p(h, k)$ whose k=2, 3, 4, j=0, h=1, …, k–1, are extracted to form a feature vector in this work. The final feature vector with six elements is $(p(0, 2)–p(1, 2), p(0, 3)–p(1, 3), p(0, 3)–p(2, 3), p(0, 4)–p(1, 4), p(0, 4)–p(2, 4), p(0, 4)–p(3, 4))$.

4 Analyzing the impact caused by synonym coding strategies

The analysis in Section 3 is based on the fact that synonyms are encoded independently of their frequencies in existing SS steganographic methods. But, if one considers the synonym frequencies when encoding the synonyms, then the stego texts may be less probable to be detected by the proposed steganalysis methods.

If the synonyms are encoded in frequency order, a determined attribute pair will map into a fixed codeword. For different synsets with the same size, their synonyms having the same attribute pair are encoded as the same codeword. Different coding strategies may encode a synonym into codewords of different lengths. Different codeword lengths would make the probabilities of the corresponding attribute pairs occurring in a text different. Therefore, different coding strategies would result in different statistical characteristics of the attribute pairs in stego texts. In the following, the impact caused by different synonym coding strategies on the extracted feature vector will be analyzed. And the analysis uses the same synonym database with the one in SS steganography.

4.1 Basic coding strategy

For a synset of size k, set $k=2^q+m$, (integer $q>0$, $m\geq 0$), then only 2^q number of synonyms in it are encoded to embed message, and their codewords have the same length. Let the first component of attribute pairs of these 2^q selected synonyms form a set denoted as S_{stego}. The remainders form a set S_{cover}. Given a synonym w with attribute pair <pos, k> in a stego text, if w has been embedded message, then pos is one of the integers in S_{stego} with the same probability $1/2^q$ (i.e. $\frac{1}{k-m}$), as the codeword length of w is q bits. If $pos \in S_{cover}$, then w must not be embedded message. Therefore,

$$f_s(j,k)=\begin{cases}(1-r)f_c(j,k), & j\in S_{cover}\\ (1-r)f_c(j,k)+\frac{1}{k-m}r\sum\limits_{i=0}^{k-1}f_c(i,k), & j\in S_{stego}\end{cases} \tag{9}$$

$$p_s(j,k)-p_s(h,k)=\begin{cases}(1-r)(p_c(j,k)-p_c(h,k)), & j<h,\ j\in S_{stego},h\in S_{stego}\\ (1-r)(p_c(j,k)-p_c(h,k))+\frac{1}{k-m}r, & j<h,\ j\in S_{stego},h\in S_{cover}\\ (1-r)(p_c(j,k)-p_c(h,k))-\frac{1}{k-m}r, & j<h,\ j\in S_{cover},h\in S_{stego}\\ (1-r)(p_c(j,k)-p_c(h,k)), & j<h,\ j\in S_{cover},h\in S_{cover}\end{cases} \tag{10}$$

If $m=0$, then $S_{cover}=\varnothing$. For any $j,h\in\{0,1,\ldots,k-1\}$, $j \in S_{stego}$, $h \in S_{stego}$, and $p_s(j,k)-p_s(h,k)=(1-r)(p_c(j,k)-p_c(h,k))$, which is in accordance with the Eq. (7).

If $m\neq 0$, $S_{cover}\neq\varnothing$, when $j \in S_{stego}$, $h \in S_{stego}$ or $j \in S_{cover}$, $h \in S_{cover}$, then $p_s(j,k)-p_s(h,k)=(1-r)(p_c(j,k)-p_c(h,k))$, which is consistent with Eq. (7); when $j \in S_{stego}$, $h \in S_{cover}$, then the difference between $p_s(j, k)-p_s(h, k)$ and $(p_c(j, k)-p_c(h, k))$ is reduced compared to Eq. (7). Conversely, when $j \in S_{cover}$, $h \in S_{stego}$, then the difference is increased. Thus, just taking $p(j, k)-p(h, k)$ with $j \in S_{stego}$, $h \in S_{cover}$ as features may increase the difficulty of detecting message embedded by steganography encoding synonyms in frequency order. In the proposed steganalysis, only the components $p(0, 3)-p(1, 3)$, $p(0, 3)-p(2, 3)$ of the feature vector satisfy the condition $m\neq 0$. Two integers of 0, 1, and 2 must belong to S_{stego}, so in the worst case, only one of $p(0, 3)-p(1, 3,)$, $p(0, 3)-p(2, 3)$ would satisfy $j \in S_{stego}$, $h \in S_{cover}$.

Based on the above analysis, just one component of the proposed feature vector fails. Therefore, although the detection accuracy of the proposed steganalysis would be slightly declined, our steganalysis would still be effective in this case.

4.2 Multi-base coding strategy

Since the big integer M converted from the secret message can be regarded as a random integer, then the value of t_i is random between 0 and k_i-1, when $t_i=M$ mod k_i. When SS steganography utilizes t_i to select a synonym, the selected synonym is random. Thus if a synonym with attribute pair <pos, k> contains secret message, the probability of pos being arbitrary integer between 0 and $k-1$ is $1/k$. In this case,

$$f_s(j,k)=(1-r)f_c(j,k)+\frac{1}{k}r\sum_{i=0}^{k-1}f_c(i,k) \tag{11}$$

$$p_s(j,k)-p_s(h,k)=(1-r)(p_c(j,k)-p_c(h,k)) \tag{12}$$

Equation (12) is in accordance with the Eq. (7). The detection performance of the proposed method is not affected by the order of encoding synonyms using multi-base coding strategy in SS steganography.

4.3 Binary tree-based coding strategy

For each synset, a binary tree is constructed. The synonyms in different levels of a binary tree have different codeword lengths. Denote the codeword length of the synonym with attribute pair <*j*, *k*> as l_{jk}, $l_{jk} \in \{1, \dots, k-1\}$. If a synonym *w* with attribute pair <*pos*, *k*> contains secret message, then the probability of *pos* being arbitrary integer *j* between 0 and *k*−1 is known to be $1/2^{l_{jk}}$, and $\sum_{j=0}^{k-1} 1/2^{l_{jk}} = 1$. Therefore,

$$f_s(j,k) = (1-r)f_c(j,k) + \frac{1}{2^{l_{jk}}} r \sum_{i=0}^{k-1} f_c(i,k) \tag{13}$$

$$p_s(j,k) - p_s(h,k) = (1-r)(p_c(j,k) - p_c(h,k)) + \left(\frac{1}{2^{l_{jk}}} - \frac{1}{2^{l_{hk}}}\right) r \tag{14}$$

If $l_{jk} > l_{hk}$, $\left(\frac{1}{2^{l_{jk}}} - \frac{1}{2^{l_{hk}}}\right) r < 0$, $p_s(j,k) - p_s(h,k)$, difference between $p_s(j, k)$−$p_s(h, k) < (1-r)$ ($p_c(j, k)$−$p_c(h, k)$) is enlarged. This is beneficial for the proposed method to detect this type of stego texts. Thus, no experiments on this case are conducted.

If $l_{jk} < l_{hk}$, $\left(\frac{1}{2^{l_{jk}}} - \frac{1}{2^{l_{hk}}}\right) r > 0$, the difference between $p_s(j, k)$−$p_s(h, k)$ and ($p_c(j, k)$−$p_c(h, k)$) is smaller than that in the case of Eq. (7). This will make the detection performance drop. But the impact may be small, since $\max\left(\frac{1}{2^{l_{jk}}} - \frac{1}{2^{l_{hk}}}\right) = \frac{1}{2} - \frac{1}{2^{k-1}} < \frac{1}{2}$. When *k*=2, 3, 4, $\max\left(\frac{1}{2^{l_{jk}}} - \frac{1}{2^{l_{hk}}}\right) = 0, 0.25, 0.375$, respectively. It is worthwhile to note that the coding strategy can only encode to make l_{02}=l_{12}=1 for *k*=2. At this time, Eq. (14) is in accordance with Eq. (7). To make $l_{jk} < l_{hk}$, synonyms with lower frequencies may be encoded as longer codewords while the ones with higher frequencies are encoded as shorter codewords. Based on this consideration, when *k*>2, the worst case for the proposed steganalysis is to construct the special binary tree so that l_{ik}=*i*+1, $l_{(k-1)k} = k-1$, *i*=0, …, *k*−2. The constructed tree is not a complete binary tree any more in this case. In the experiments, a SS steganographic tool called Hsyn, which adopted this kind of special binary tree to encode synonyms in frequency order, has been implemented.

5 Experimental results and discussion

5.1 Experimental setup

A book named "Word Frequencies in Written and Spoken English: based on the British National Corpus" [8] provides several kinds of frequency lists derived from a 100,000,000 word electronic databank. A complete alphabetical frequency list without frequency cut-offs was downloaded from a companion website [9] for this book. With the help of this frequency list, we preprocess the synonym database to obtain an attribute pair sequence for each text.

In experiments, 5,622 texts were used as cover texts. These texts come from one chapter, several chapters, or a whole book, randomly downloaded from the Internet. The embedding capacities of these texts are significantly in a wide range, which helps to objectively evaluate the performance of steganalysis. The cover texts are embedded arbitrary messages by tools Tlex, Bsyn, Ctsyn with four embedding rates, 25%, 50%, 75%, and 100% to generate 3*4 groups of stego texts. The number of stego texts is 5,622*3*4 in total. Tools Bsyn, Tlex and Ctsyn were implemented with basic, multi-base and binary complete tree-based coding strategies respectively to encode synonyms in alphabetical order. In order to train a classifier, 2,000 texts were selected from the cover texts, and 1,000 texts were selected from each groups of stego texts to form a training set. The remainders formed a testing set.

According to the discussions in Section 4, three tools Bsyn-fre, Tlex-fre, Hsyn were implemented, which adopted basic, multi-base, binary tree-based coding strategies respectively to encode synonyms in frequency order. 5,622*3*4 stego texts were generated by these three tools with four embedding rates, 25%, 50%, 75%, and 100%.

For all the above SS steganographic tools, they used the same synonym database as well as in the tool Tlex, denoted as DB#1.

5.2 Detection performance analysis

In the experiments, the SVM with RBF kernel is utilized as the classifier. The software used is LIBSVM2.9 [3]. Receiver operation characteristic (ROC) curve is used to display the detection probability (the fraction of the stego texts that are correctly classified) in terms of the false positive probability (the fraction of the cover texts that are misclassified as stego texts) in this study. And to evaluate the overall goodness of the ROC curve, we use the area under the ROC curve (AUC) [18]. $AUC = \int_0^1 P_D(P_{FP})dP_{FP}$, where P_D is the detection probability, P_{FP} is the false positive probability. AUC is a common summary statistic for the goodness of a predictor in a binary classification task.

The AUCs of our steganalysis with DB#1 are listed in Table 1. Experimental results in Table 1 demonstrate that the proposed steganalysis can effectively detect different SS steganographic tools with various embedding rates. Especially, when the embedding rate is greater than 50%, the detection performance is very good. However, when the embedding rate is 25%, the detection performance is not very good. It is easy to find that the detection performances for some steganographic tools with the same embedding rate are very approximate. In fact, the corresponding ROC curves may be overlapped. Since the difference between detection performances for Bsyn-fre and Bsyn are slightly great, just the ROC curves of our steganalysis for Bsyn-fre and Bsyn are depicted in Fig. 3. It can be seen that the

Table 1 The AUCs of our steganalysis with DB#1

Embedding rate	Steganographic tool					
	Bsyn	Bsyn-fre	Tlex	Tlex-fre	Ctsyn	Hsyn
100%	0.9996	0.9954	0.9994	0.9989	0.9997	0.9978
75%	0.9973	0.9840	0.9974	0.9965	0.9977	0.9883
50%	0.9799	0.9396	0.9749	0.9745	0.9818	0.9463
25%	0.8686	0.7813	0.8445	0.8428	0.8631	0.7889
Average	0.96135	0.925075	0.95405	0.953175	0.960575	0.930325

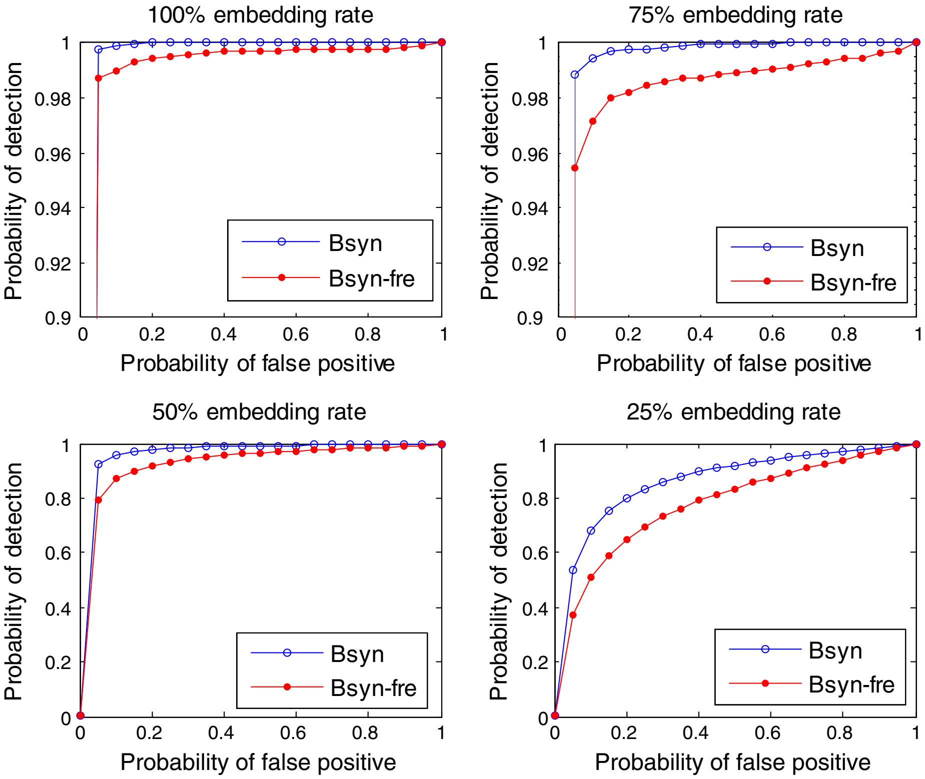

Fig. 3 ROC curves of our steganalysis with DB#1 against part of SS steganographic tools

AUCs for Bsyn-fre and Hsyn are lower than those for Bsyn and Ctsyn, respectively. This indicates that the detection performance of our steganalysis is slightly affected by the frequency order of synonyms in a synset when encoding by basic and binary tree-based coding strategies. But the AUCs for Tlex and Tlex-fre are approximate, namely, the detection performance is nearly influenced for synonyms in any order for multi-base coding strategy.

5.3 Analyzing the impact on detection performance caused by synonym database

Some current SS steganographic algorithms may measure the suitability of a substitution. These behaviors can be regarded as reducing the embedding rate or the number of synonyms in a database. On the other hand, different researchers may extract different synonym databases from different dictionaries or select different synonym sets to form a database. If the synonym databases used in the SS steganography and steganalysis include different synonyms, then for steganalysis, some synonyms with secret message would not be recognized, or some unrelated words are recognized as synonyms, or the same synonym may be located in a synset different from the one used in the steganography. It is intuitive that using different synonym databases will impact the detection performance of linguistic steganalysis.

Considering that the absolute synonyms have the higher probabilities of being used by all SS steganographic methods, we built an absolute synonym database extracted from the

Wordnet 2.1 [21] for windows, called DB #2. Here, the absolute synonyms mean that words are synonymous in any of their senses. The size of DB #2 is smaller than that of DB #1. 24.24% words in DB #2 do not belong to the DB #1, while 53.92% words in DB #1 do not belong to DB #2.

Table 2 gives the AUCs of our steganalysis with DB#2 for the six SS steganographic tools. And the ROC curves of our steganalysis with DB#1 and DB#2 for Bsyn-fre and Bsyn are shown in the Fig. 4. Despite that the detection performance declines trivially compared with the case of using DB #1, the AUCs still maintain relatively high values. When the embedding rate is 50%, the detection performances with DB#2 are still good for Bsyn, Tlex, Tlex-fre, Ctsyn. But when the embedding rate is 25%, the detection performance is worse than that in the case of using DB#1. These results demonstrate that our steganalysis using an absolute synonym database can deliver good performance while lacking the synonym database used in SS steganography.

5.4 Detection performance comparison with other steganalysis methods

One of the existing linguistic steganalysis against SS steganography was presented in [12] (denoted as Luo's method for convenience). This method recognized all synonyms appearing in a text, and made the synonyms from the same synset form a sequence. The sequences containing one word were neglected. In the cover text, a synonym always continuously appeared in a sequence. But in the stego text, synonyms in a sequence would frequently alter to represent different secret message bits. Under this consideration, a statistic characteristic was calculated. This method just extracted statistics from part of the synonyms in a text. If the total occurrences of a word and its synonyms are less than two times, then this word will be ignored. Few sequences containing more than one word will lead to low detection accuracy. This method is not good at detecting the stego text with low embedding capacity.

The AUCs of Luo's method with DB#1 and DB#2 are given in Tables 3 and 4, respectively. In order to compare the detection performance, in Fig. 5, we depict the AUCs of our and Luo's method with both DB#1 and DB#2 in bar charts in terms of the embedding rates used by SS steganographic tools.

From the experimental results, we can see that our steganalysis greatly outperforms Luo's with both DB #1 and DB #2. The detection performances of our and Luo's methods with DB #1, which is identical to the synonym database in SS steganographic tools, are better than those with DB #2, which is an absolute database and different from DB #1. The performance of Luo's method with DB #2 is poor, while

Table 2 The AUCs of our steganalysis with DB#2

Embedding rate	Steganographic tool					
	Bsyn	Bsyn-fre	Tlex	Tlex-fre	Ctsyn	Hsyn
100%	0.9783	0.9518	0.9839	0.9757	0.9818	0.9217
75%	0.9600	0.9098	0.9692	0.9598	0.9682	0.8649
50%	0.9044	0.8250	0.9016	0.8916	0.9165	0.7721
25%	0.7607	0.6898	0.7420	0.7324	0.7607	0.6501
Average	0.90085	0.8441	0.899175	0.889875	0.9068	0.8022

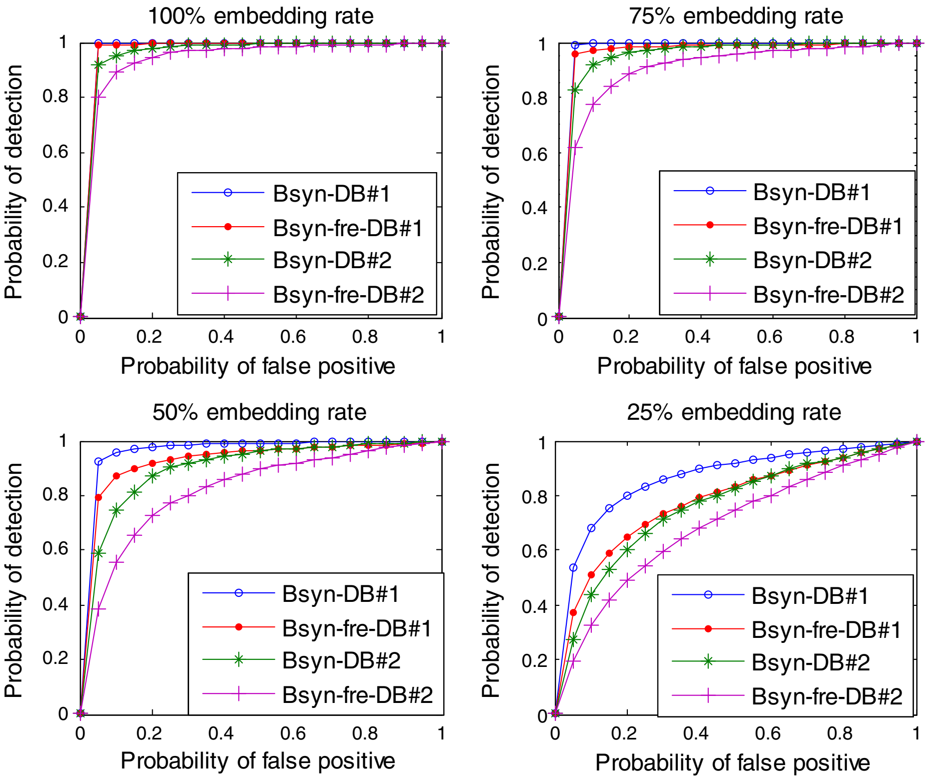

Fig. 4 ROC curves of our steganalysis with DB #1 and DB #2 against part of SS steganographic tools

our method is still effective. In addition to that, in most situations, the performance of our method with DB #2 is better than that of Luo's even if the DB #1 is used. However, with very few exceptions, our method may perform worse than Luo's. Because the performance of our method is affected by some synonym coding strategies and the used synonym database, the AUCs of our method are relatively low while using DB#2 to detect the stego texts generated by Bsyn-fre and Hsyn with embedding rate 25%. In contrast, Luo's method is not affected by the order of encoding synonyms in a synset, and its performance is significantly influenced by

Table 3 The AUCs of Luo's method with DB#1

Embedding rate	Steganographic tool					
	Bsyn	Bsyn-fre	Tlex	Tlex-fre	Ctsyn	Hsyn
100%	0.9033	0.9033	0.8602	0.8660	0.9010	0.9028
75%	0.8967	0.8896	0.8554	0.8581	0.8945	0.8873
50%	0.8640	0.8542	0.8116	0.8112	0.8619	0.8486
25%	0.7729	0.7609	0.7114	0.7126	0.7707	0.7542
Average	0.859225	0.852	0.80965	0.811975	0.857025	0.848225

Table 4 The AUCs of Luo’s method with DB#2

Embedding rate	Steganographic tool					
	Bsyn	Bsyn-fre	Tlex	Tlex-fre	Ctsyn	Hsyn
100%	0.7989	0.8066	0.7882	0.7899	0.8037	0.8040
75%	0.7973	0.7977	0.7818	0.7826	0.7971	0.7919
50%	0.7729	0.7688	0.7482	0.7470	0.7711	0.7603
25%	0.7103	0.7068	0.6780	0.6781	0.7139	0.7000
Average	0.76985	0.769975	0.74905	0.7494	0.77145	0.76405

the length of the embedded secret message rather than the embedding rate. Thus Luo’s method has similar performance for Bsyn and Bsyn-fre, as well as for Ctsyn and Hsyn. When the DB #2 is used for detecting the stego texts with embedding rate 25%, the AUCs of Luo’s method are not very low. Consequently, in the case of using DB#2 for detecting Bsyn-fre and Hsyn with embedding rate 25%, it is possible that the AUCs of our method are slightly lower than those of Luo’s, as shown by the results in Fig. 5.

Another steganalysis method for SS steganography is Yu’s method [23]. It defined a suitability function based on the collection frequency of a word and its context to evaluate

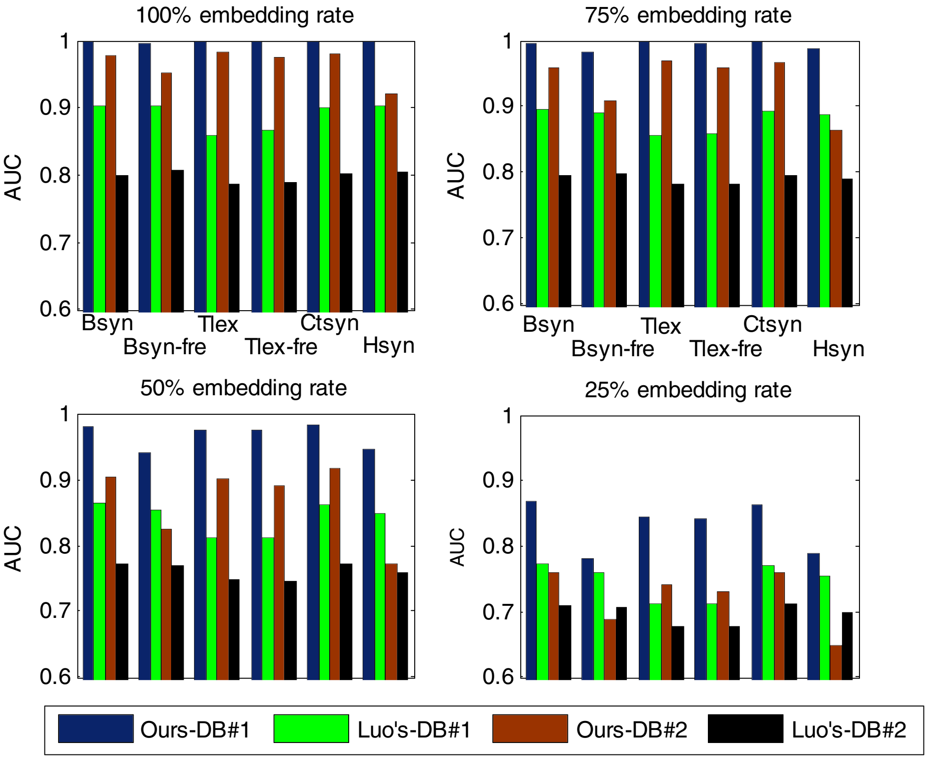

Fig. 5 The bar chart of AUCs for our and Luo’s methods both with DB#1 and DB#2

the suitability of a synonym for its context. A weighted value for each synonym in the text was estimated by the suitability function. As the results of SS steganography, most synonyms with high weighted values would be replaced by the ones with low weighted values. Thus, the expectation and variance of the weighted value sequence in the text composing a feature vector for SVM can be used for distinguishing stego texts from cover texts. However, this feature vector was not discriminative enough. The weighted values of synonyms from not only the same synsets but also the different synsets are so different that the feature vectors of cover texts may have significant difference. On the other hand, compared with the original cover text, only parts of synonyms with hidden message in a stego text were substituted. Although most high weighted values were altered to low ones, a few low weighted values were also changed to high ones. The distance between the feature vectors of texts before and after embedding message was small. The embedding rate was lower; less weighted values were modified by SS steganography. Thus, the stego texts and cover texts were not extremely discriminable by using the feature vector in [20], especially for the stego texts with low embedding rate.

Moreover, according to the detecting algorithm of this method, if the word has n synonyms (including itself), it needs to query the results in Google at least n times, which makes the detection process very time-consuming. What is worse, the program would be forced to terminate whenever Google detects the automated traffic, since automated queries are against their Terms of Service [7]. This method is not suitable for real-time detection of secret message in a text.

Because of the difficulty mentioned above, only parts of the texts in our sample set were selected to test the detection performance of Yu's method. 1,000 cover texts with relatively small file sizes, and the corresponding stego texts generated by Tlex with four embedding rates were used. The train set was composed of 500 cover texts and 350 stego texts for each embedding rate. Table 5 lists the AUCs for our and Yu's methods with DB #1 and DB #2. And the corresponding ROC curves are shown in Fig. 6. Experimental results show that our method outperforms Yu's. The cover texts and stego texts in these experiments having small file sizes result in small embedding capacities, and when the embedding rate is low, the number of modifications caused by embedding operations is minuscule. Thus the values of the AUCs of our method in Table 5 are slightly less than those of Tables 1 and 2.

6 Conclusion

In this paper, the word frequency is taken into consideration to represent the statistical attribute of a synonym. After successfully converting the synonyms within a text into their

Table 5 The AUCs of our and Yu's methods with both DB#1 and DB#2

Steganographic tool	Embedding rate	AUC			
		Our method with DB #1	Yu's method with DB #1	Our method with DB #2	Yu's method with DB #2
Tlex	100%	0.9992	0.9431	0.9695	0.9061
	75%	0.9964	0.9146	0.9395	0.8868
	50%	0.9681	0.8340	0.8582	0.8058
	25%	0.8069	0.6757	0.6807	0.6727
Average	–	0.94265	0.84185	0.861975	0.81785

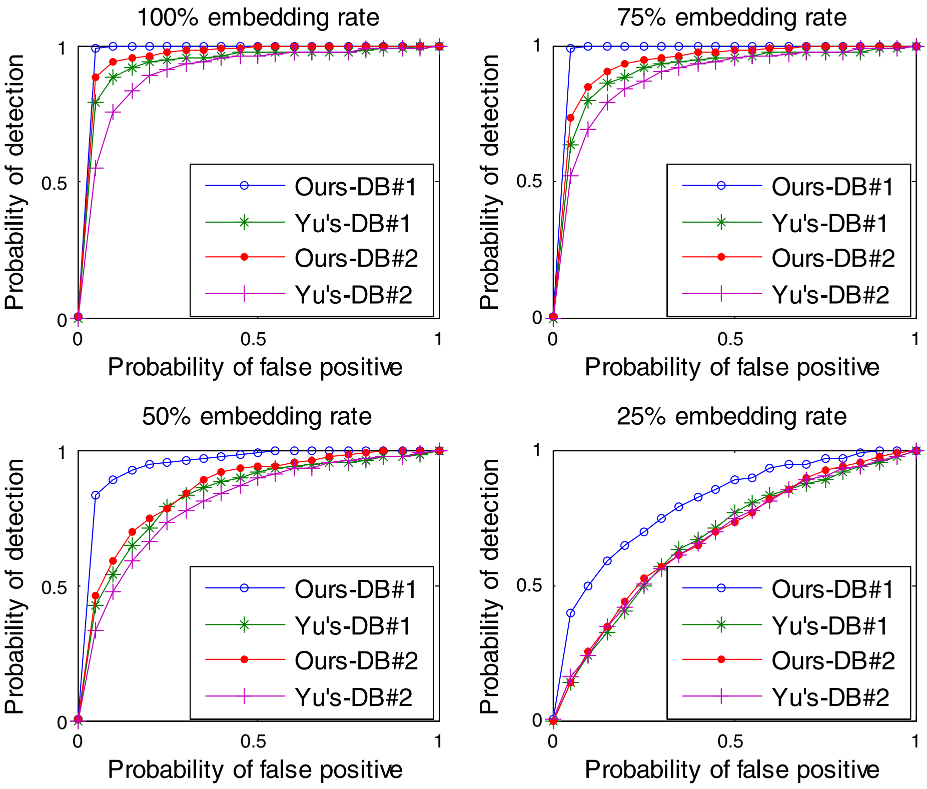

Fig. 6 ROC curves of our and Yu's methods with DB #1 and DB #2 against Tlex

attribute pairs, differences between the statistical characteristics of attribute pairs in cover and stego texts are analyzed. An effective feature vector is obtained in order to detect the SS steganography. Not only the existing SS steganographic algorithms but also the future possible algorithms using different synonym coding strategies to encode synonyms in frequency order have been investigated. The detection performances of our steganalysis are discussed, and the experimental results show that the proposed method achieves good performance in all the above cases.

When the proposed steganalysis without the knowledge of the synonym database used in SS steganography, it may lead the detection probability of the steganalysis drop. We built an absolute synonym database from WordNet for experiments. Although the results demonstrate that the detection performance of our steganalysis is still good, it is worse than that in the case of using the same synonym database with the one in SS steganography. In the future, more endeavors should be made to improve the detection performance of the steganalysis with an arbitrary synonym database.

Finally, the experimental results have shown that the proposed method has significant advantages over other methods. It is noteworthy that the proposed method is only supported by a preprocessed synonym database. When extracting the feature vector from the text, neither huge corpus nor Google is needed. The speed of the proposed steganalysis detecting a text is fast.

Acknowledgments This work was supported in part by National Natural Science Foundation of China (Nos. 60973128, 61073191, 61070196, 61070195, 61103215, 61173141, 61173142, and 61232016), National Basic Research Program 973 of China (Nos. 2010CB334706, and 2011CB311808), 2011GK2009, GYHY201206033, 201301030, 0S2013GR0445 and PAPD fund.

References

1. Atallah MJ, Raskin V, Crogan M, Hempelmann C, Kerschbaum F, Mohamed D, Naik S (2001) Natural language watermarking: design, analysis, and a proof-of-concept implementation. In: Proceedings of 4th International Workshop Information Hiding, Lecture Notes in Computer Science, Springer, Berlin, vol 2137, pp 185–199
2. Bolshakov A (2004) A method of linguistic steganography based on collocationally-verified synonymy. In: Proceedings of 6th International Workshop Information Hiding, Lecture Notes in Computer Sciences, Springer, Berlin, vol 3200, pp 180–191
3. Chang CC, Lin CJ (2010) LIBSVM: a library for support vector machines. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Chapman MT, Davida GI (1997) Hiding the hidden: a software system for concealing ciphertext as innocuous text. In: Proceedings of the International Conference on Information and Communications Security, Lecture Notes in Computer Sciences, Springer, Berlin, vol 1334, pp 333–345
5. Chen ZL, Huang LS, Yu ZS, Yang W et al (2008) Linguistic steganography detection using statistical characteristics of correlations between Words. In: Proceedings of 10th International Workshop on Information Hiding, Lecture Notes in Computer Sciences, Springer, Berlin, vol 5284, pp 224–235
6. Chiang YL, Chang LP, Hsieh WT, Chen WC (2003) Natural language watermarking using semantic substitution for Chinese text. In: Proceedings of 2nd International Workshop Digital Watermarking, Lecture Notes in Computer Sciences, Springer, Berlin, vol 2939, pp 129–140
7. Google Terms of Service (2010) [Online]. Available: http://www.google.com/accounts/TOS?hl=en
8. Leech G, Rayson P, Wilson A (2001) Word frequencies in written and spoken english: based on the British National Corpus. Longman, London
9. Leech G, Rayson P, Wilson A (2010) Word frequencies in written and spoken english: based on the British National Corpus. [Online]. Available: http://ucrel.lancs.ac.uk/bncfreq/
10. Liu YL, Sun XM, Gan C, Wang H (2007) An efficient linguistic steganography for Chinese text. In: Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, pp 2094–2097
11. Liu YL, Sun XM, Liu YP, Li CT (2008) MIMIC-PPT: mimicking-based steganography for microsoft PowerPoint document. Inf Tech J 7(4):654–660
12. Luo G, Sun XM, Xiang LY, Liu YL, Gan C (2008) Steganalysis on synonym substitution steganography. J Comput Res Dev (Chinese) 45(10):1696–1703
13. Meral HM, Sankur B, Özsoy AS, Güngör T, Sevinç E (2009) Natural language watermarking via morphosyntactic alterations. Comput Speech Lang 23(1):107–125
14. Muhammad HZ, Rahman SMSAA, Shakil A (2009) Synonym based Malay linguistic text steganography. In: 2009 Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, pp 423–427
15. Shirali-Shahreza MH, Shirali-Shahreza M (2008) A new synonym text steganography. In: Proceedings of the 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp 1524–1526
16. Taskiran CM, Topkara U, Topkara M, Delp EJ (2006) Attacks on lexical natural language steganography systems. In: Proceedings of the SPIE, Security, Steganography and Watermarking of Multimedia Contents VIII, vol 6072, pp 97–105
17. Topkara U, Topkara M, Atallah MJ (2006) The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In: Proceedings of the 8th Workshop on Multimedia and security. ACM Press, pp 164–174
18. Wang Y, Moulin P (2007) Optimized feature extraction for learning-based image steganalysis. IEEE Trans Inf Forensics Secur 2(1):31–45
19. Winstein K (2010) Lexical steganography through adaptive modulation of the word choice hash. [Online]. Available: http://alumni.imsa.edu/~keithw/tlex/lsteg.ps
20. Winstein K (2010) Tyrannosaurus lex. [Online]. Available: http://alumni.imsa.edu/~keithw/tlex/
21. WordNet (2010) [Online]. Available: http://wordnet.princeton.edu/

22. Yang JL, Wang JM, Wang CK, Li DY (2007) A novel scheme for watermarking natural language text. In: Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, vol. 2, pp. 481–484
23. Yu ZS, Huang LS, Chen ZL, Li LJ, Zhao XX, Zhu YW (2008) Detection of Synonym-Substitution Modified Articles Using Context Information. In: Proceedings of 2nd International Conference on Future Generation Communication and Networking, vol 1, pp 134–139

Lingyun Xiang received her B.E. degree in computer science and technology, in 2005, and the Ph. D. degree in computer application, in 2011, Hunan University, Hunan, China. Currently, she is a Lecturer in College of Computer and Communication Engineering, Changsha University of Science & Technology, Hunan, China. Her current research interests include information security, steganography, steganalysis, machine learning, pattern recognition and computer vision.

Xingming Sun received his B.S. degree in mathematics from Hunan Normal University, Hunan, China, in 1984, the M.S. degree in computing science from Dalian University of Science and Technology, Dalian, Liaoning, China, in 1988, and Ph.D. degree in computing science from Fudan University, Shanghai, China, in 2001. He is currently a Professor in College of Computer and Software, Nanjing University of Information Science and Technology, Jiangsu, China. His research interests include network and information security, digital watermarking, database security, and natural language processing.

Gang Luo received his B.E. degree in microelectronics technology, in 1998, the M.E. degree in software engineering, in 2004, and the Ph. D. degree in computer application, in 2008, Hunan University, Hunan, China. Currently, he is a Lecturer in College of Information Science and Engineering, Hunan University, Hunan, China. His research interests include information security, steganalysis, information hiding, and cryptanalysis.

Bin Xia received his B.E. in Computer Science and Technology from Hunan University, China, in 2008. He is currently pursuing his M.E. in computer science and technology at the College of Information Science and Engineering from Hunan University, Hunan, China. His research interests include steganography and steganalysis, image processing, and pattern recognition.