Syntactic Steganography Approach for Information Security

Ei Nyein Chan Wai, May Aye Khine
University of Computer Studies, Yangon
einyeinchanwai@gmail.com

Abstract

In today digital age, there are more demands to improve techniques for information security. One of the solutions is steganography, hiding sensitive information within innocent-looking cover media. In this paper, we propose a syntax-based steganography approach by using the syntax bank. The input raw sentence is parsed with the Stanford parser that can produce a phrase structure of the sentence. This parse structure is then used to produce the syntax of the sentence. At the same time, Shannon-Fano coding is used to compress the input secret message as minimum total bits length as possible. Then, syntax transformation task searches the syntax set of the given sentence within the syntax bank, and then transforms it into a desired syntax that can represent the key-controlled semi-randomly generated secret bits intended to hide in the sentence. The resulting stego text will still be innocent-looking by applying semantically unchanged syntax transformation on the input text. Thus, the detection of the secret message may be hard for the intruder.

1. Introduction

The word steganography is of Greek origin and means "concealed writing". Its purpose is to establish communication between two parties whose existence is unknown to a possible attacker. It is the practice of hiding private or sensitive information within something that appears to be nothing out of the usual, and the term applied to any number of processes that will hide a message within an object, where the hidden message will not be apparent to an observer. It is not intended to replace the commonly used cryptography but to supplement it. Steganography conceals the secret message whereas cryptography codes the message so as to render them unintelligible to other than authorized recipients.

It has found usages variously in military, diplomatic, personal and intellectual property applications. It has been widely used since historical times until the present day. In ancient Greece, the hidden messages were tattooed on a slave's (the massager’s) shaved head, hidden by the growth of his hair, and exposed by shaving his head again. Another form of steganography is by using secret inks, under other messages or on the blank parts of other messages. During World War II, a spy for the Japanese in New York City sent information to accommodation addresses in neutral South America by the stego text within the ‘doll’ orders.

In digital steganography for today era, modern steganography includes the concealment of information within computer files. The different types of secret message, such as audio, image, and text, can be hidden in the different types of cover media, such as audio, video, image, text, and so on. Among these different cover media, texts are widely used in several processes. However, it is also the most difficult kind of steganography because it is due largely to the relative lack of redundant information in a text file.

Text steganography is broadly classified into the two categories; linguistic approach which is the art of using written natural language to conceal secret messages and format-based approach which used physical formatting of text as a place in which to hide information. The former can be divided into semantic and syntactic method and the latter can also be divided into line-shift, word-shift, open-space and feature encoding [10].

There are three dimensions in a stego system,

1. Payload Capacity: the ratio of hidden information to cover information.
2. Robustness: the ability of the system to resist against changes in the cover object.
3. Imperceptibility: the potential of the generated stego object to remain indistinguishable from other objects in the same category.[4]

These are often contradictory requirements: for example, imperceptibility limits the payload.

In this paper, a steganographic approach is proposed for linguistic steganography by using the Shannon-Fano compressing algorithm, the statistical Stanford parser and a syntactic method based on the syntax bank.

The rest of the paper is organized as follows. In section 2, a brief overview of existing linguistic steganography methods will be presented. Section 3 will explain the syntax of the language. Section 4 presents our proposed method. Finally, the conclusion and future work will be placed in section 5.

2. Linguistic Steganography

Linguistic Steganography is concerned with making changes to a cover text in order to embed information, in such a way that the changes do not result in ungrammatical or unnatural text. Most of the linguistic steganography methods use either lexical (semantic) or syntactic transformations or combination of both. The synonym substitution is the popular lexical steganography method. It substitutes the original word with one of the word that belongs to the same synonym set of the original word. The syntactic methods transform the grammatical style of the original sentences. It also constitutes the swapping of word that cannot affect the meaning of the original sentence.

2.1. Lexical Steganography

In [4], Brecht Wyseur, Karel Wouters, and Bart Prenee proposed a linguistic steganography based on word substitution over an IRC channel. According to this work, the generation of the word substitution table is based on a session key. They used synonyms from a public thesaurus that fit into the context of the cover text. Each word in a certain subset of the thesaurus represents information.

Ching-Yun Chang and Stephen Clark proposed a method for checking the acceptability of paraphrases in context in [5] by using the Google n-gram data and a CCG parser to certify the paraphrasing grammaticality and fluency. They also proposed two improvements again in [6] by means of the WebIT Google n-gram corpus and vertex colour coding to address the problem that arises from words with more than one sense. In this attempt, words are the vertices in a graph, synonyms are linked by edges, and the bits assigned to a word are determined by a vertex colouring algorithm.

In [1], the writers used synonym replacement, which converts a message into semantically innocuous text. It also used a word dictionary to get synonym. The input text to be hidden is compressed using Huffman Compression Algorithm and a string of bits is generated. The input bits are consumed in selection of synonyms.

2.2. Syntactic Steganography

According to our recent study, B. Murphy and C. Vogel mainly proposed syntactic methods for steganography. In [2], they assumed a perfect parser and evaluated a set of automated and reversible syntactic transforms that can hide information in plain text without changing the meaning or style of a document. They examined two highly predictable and reasonably common grammatical phenomena in English that can be used in data hiding, the swapping of complementisers and relativisers, which rely on a well-established technology: syntactic parsing.

In [3], they also presented three natural language marking strategies based on fast and reliable shallow parsing techniques, and on widely available lexical resources: lexical substitution, adjective conjunction swaps, and relativiser switching. The first method is representative of function-word near-synonymy relations by searching for the pattern “COMMON-NOUN who” or “COMMON-NOUN which”, and replace the relativiser with *that*. The pattern “ADJECTIVE CONJUNCTION ADJECTIVE COMMON-NOUN” is the pivot for the second method, swapping adjective positions. The third method considered individual content words (adjectives, verbs, nouns and adverbs) to identify likely lexical substitutions using WordNet.

The other people explored the morphosyntactic tools for text watermarking and developed a syntax-based natural language watermarking scheme in [8]. The unmarked text is first transformed into a syntactic tree diagram in which the syntactic hierarchies and the functional dependencies are coded. The watermarking software then operates on the sentences in syntax tree format and executes binary changes under control of Wordnet to avoid semantic drops.

In [9], the authors developed a morphosyntax-based natural language watermarking scheme. In this scheme, a text is first transformed into a syntactic tree diagram where the hierarchies and the functional dependencies are made explicit. The watermarking software then operates on the sentences in syntax tree format and executes binary changes under control of Wordnet and Dictionary to avoid semantic drops.

2.3. Combining Lexical and Syntactic Steganography

Some work in the steganography combine lexical and syntactic methods. The methods work at the sentence level to hide the intended secret information.

In [11], the proposed scheme works at the sentence level while also using a word-level watermarking technique. The two types of modifications that can be used for watermarking text: the robust synonym substitution and syntactic sentence-paraphrasing. Again, it uses XTAG parser for parsing, dependency tree generation (which is called a derivation tree in the XTAG jargon) and linguistic feature extraction and RealPro for natural language generation.

3. Syntax of Language

The syntax of a language is the set of rules that language uses to combine words to create sentences. The parts of speech of words combine into phrases: noun phrase, verb phrase, propositional phrase, adjectival phrase, and adverbial phrase. One way of diagramming the structure of a sentence is called phrase structure rules. For example:

S -> NP VP

"A sentence is made up of a noun phrase and a verb phrase."

NP -> (Det) (AP) N (PP)

"A noun phrase is composed of a noun plus optional determinantes, adjective phrases, and prepositional phrases."

VP -> (Aux) V (NP) (PP) (AdvP)

"A verb phrase is composed of a verb plus optional auxiliary verbs, object noun phrases, prepositional phrases, and adverbial phrases."

AP -> (AdvP) A

"An adjective phrase is composed of an adjective and optional adverbial phrases."

PP -> Prep NP

"A prepositional phrase is composed of a preposition and a noun phrase."

AdvP -> (Adv) Adv

"An adverbial phrase is composed of an adverb and optional modifying adverbs." [12]

Most of today parsers produce the above phrase structure. In subject-verb-object representation, the noun phrases in the above structure become either subject or object of the sentence. Some works have done on extraction of subject(s), verb and object(s) from a sentence's phrase structure.

In [7], extraction of subject-predicate-object (subject-verb-object) triplets from English sentences is done by using well known syntactical parsers for English; namely Stanford Parser, OpenNLP, Link Parser and Minipar.

Moreover, a sentence is actually a clause, a set of words that includes at least a verb and probably a subject noun. But a sentence can have more than one clause: There may be a main clause (or independent clause) and one or more subordinate clauses. [12]

These clauses are connected by a joiner, a conjunction that can join main clause and subordinate clause. The subordinate clause come before or after of the main clause. The results can have the same meaning with the different syntaxes. For example, the following two sentences with different syntax but with the same meaning:

- While she spoke to Mary, Maisie was looking at her watch.
- Maisie was looking at her watch while she spoke to Mary.

Finally, a sentence can also have two or more main (independent) clauses, joined by coordinating conjunctions. [12]

For this compound sentence, the clauses can exchange without affecting the meaning of the sentence. For instance:

- Either I go or he goes.
- Either he goes or I go.

4. Proposed Approach

The proposed scheme combines the Stanford parser to parse the input cover text, the Shannon-Fano compression algorithm to compress the secret message so that the more secrete message can be embedded, and rules bank based syntactic steganography method to conceal the compressed secret message bits into the cover text to produce the semantic preserving stego text.

Firstly, the cover text is parsed by the parser while the secret message is compressed by the Shannon-Fano algorithm. Then, the parsed cover text sentence is transformed into one of the rules within the syntax set of the original sentence. This transformed rule is the one that has been marked with the longest binary sequence in the compressed binary form of the secret message. As long as the compressed secret message remains to hide in the cover text, the above processes are done for each of the cover text sentences. When there is no more binary sequence to hide, the cover text becomes the stego text that contains the secret in it, and ready to send over the communication channel together with the codes that compressed the secret.

When the stego text reaches to the receiver side, it is firstly parsed by the parser to get the grammar structure of it. Then the syntactic checking step finds the syntax set of the stego text sentence by sentence. Moreover, this step finds out the corresponding binary sequence of it. By carrying out these steps for each sentence of the stego text, the binary representation of the compressed secret message will be retained. This is then decompressed by the codes came together with it.

The sender's side and the receiver's side of the proposed system are shown in the figure 1 and 2 respectively.

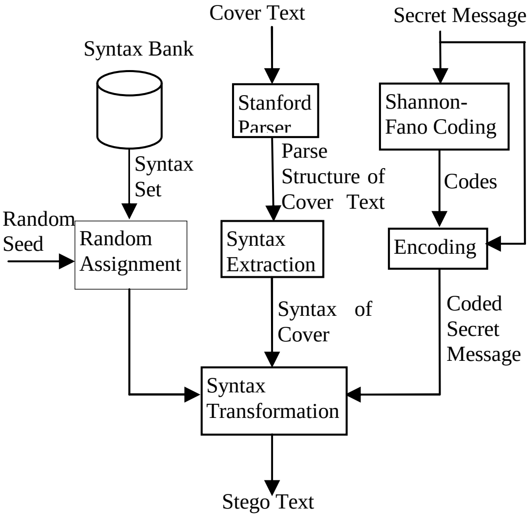

Figure 1. Proposed System (Sender Side)

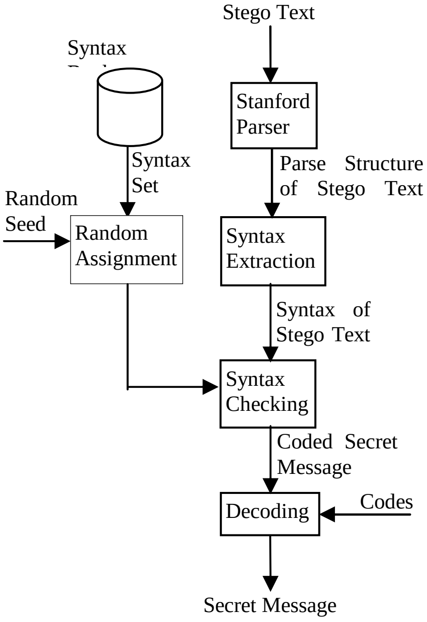

Figure 2. Proposed System (Receiver side)

4.1 Shannon-Fano Algorithm

This is a technique for constructing a prefix code based on a set of symbols and their probabilities. The symbols are arranged in order from most probable to least probable, and then divided into two sets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned; symbols in the first set receive "0" and symbols in the second set receive "1". As long as any sets with more than one member remain, the same process is repeated on those sets, to determine successive digits of their codes. When a set has been reduced to one symbol, of course, this means the symbol's code is complete and will not form the prefix of any other symbol's code. [13]

For example, the secret word “bamboo” is compressed as follows:

Table 1. Shannon-Fano code of “bamboo”

Character	Frequency	Code
b	2	0
o	2	10
a	1	110
m	1	111

By using the above codes, the coded secret message is 0 110 111 0 10 10.

4.2 The Stanford Parser

This is a Java implementation of probabilistic natural language parsers, a program that works out the grammatical structure of sentences. For instance, which groups of words go together (as "phrases") and which group of words is the subject or object of a verb. It uses knowledge of language gained from hand-parsed sentences to try to produce the *most likely* analysis of new sentences. Although these statistical parsers still make some mistakes, but commonly work rather well. [14]

The output of this parser, the phrase structure grammar representation of the sentence, is used as the input of the syntactic transformation stage of the sender and the syntactic checking stage of the receiver.

4.3. Syntactic Steganography using Syntax Bank

The proposed method uses syntax bank that consists of a number of the syntax sets and has already shared between the sender and the receiver. A syntax set is a set of all available syntax forms of a sentence which are semi-randomly assigned a binary number for each. The number of secret bits which can be hidden in a sentence depends on the number of available syntaxes in the syntax set in which the sentence’s original syntax exists.

If there is more than one clause in the input sentence, the syntax set includes not only the syntax for the whole sentence, but also that for each clause.

At the sender side, the syntax transformation step takes the grammar structure produced by the parser as input, and transforms it into its syntax form. Then, this rule is checked to see which of the syntax set it belongs to. When such a set is found, the syntax with longest length for the desired binary sequence of the compressed secret message is applied on the sentence.

The following figure describes the example case of the proposed method.

Secret Message: bamboo

Coded Secret Message: 0 110 111 0 10 10

Cover text:
After we have received the goods, we will settle the accounts. ………..

Stego text:
The accounts will be settled after the goods have been received…………

Figure 3. Example case

For the receiver side, the syntax checking step uses the grammar structure to produce the syntax of the stego text sentence. When getting this rule, it checks which syntax set possesses this and what the corresponding binary sequence of this in it is.

4.3.1 Key-Controlled Semi-Random number assignment

The sender and the receiver have already shared a key that is used as a seed to produce the same random sequence assigned to the syntactic rules of the set. The algorithm that can produce the unique random numbers is described as follows:

```
function generateUniqueRandom (Long seed, int max) returns random
    temp = generate new-random within 0 to max interval;
    if ( ! previous-random) add temp to previous-random;
    else {
        while ( temp € previous-random)
            temp = generate new-random;
    }
    return temp;
```

Figure 4. The algorithm for generating unique random number

4.3.2 Syntax Transformation

This step transforms the input sentence into the desired syntax form. The most possible transformation is active-passive transformation. This can be used for all sentences and clauses that contain subject, verb, and object. In addition, there is also possible to interchange the clauses back and front. Apart from this, there may be many other ways to transform the sentence retaining its meaning such as topicalization, adverb displacement, and so on.

4.4 Experimental Result

As our system is still implementing, we can only show a partial experimental result about this. At the time of writing, our test case includes 50 sentences with about 10 words pre sentence. Among these sentences, 20 sentences are simple sentences while the rest of the sentences are compound and complex sentences that consists of two clauses.

Also, we have implemented active-passive and clauses interchanging methods for syntax transformation.

For simple sentences, there is one syntax set with two syntax forms, active and passive. The following table shows the hidden capacity of this set. The 10 out of 20 sentences can be applied to the active-passive transformation.

Table 2. Syntax set of simple sentences

Set	Hidden bits per sentence	No. of Sentences
Active and passive	1	10

In the case of compound and complex sentences, there are four syntax sets. These sets are described in the following table.

Table 3. Syntax sets of compound and complex sentences

Set	Hidden bits per sentence	No. of Sentences
Clauses interchanging	1	7
Clauses interchanging, main clause's active and passive	2	11
Clauses interchanging, subordinate clause's active and passive	2	0
Clauses interchanging, both clauses' active and passive	3	4

This payload capacity of our proposed system can be improved by adding other transformation methods. The more syntax forms we can apply to, the better the capacity of our system will be.

Moreover, our proposed system will not change the appearance of the cover text because it is based upon the syntax instead of the format-based method. In addition, the syntax set of the proposed system is a collection of different syntax forms that can produce the same meaning. The syntax transformation task only transforms from one form to another within the same syntax set. Thus, the meaning of the result stego text sentences is the same as their original cover text sentences. Due to this retaining appearance and meaning, the proposed method can produce natural looking text as the cover text.

What is more, the syntax sets of our method are the collections of syntax forms that can produce the same meaning. The number of bits that can be hidden by a syntax set is defined as the power of two of the number of forms in this set. For instance, the syntax set that contains two syntax forms can be used to hide one secret bit; the set with four forms can hide two secret bits, and so on. Thus, the more the number of syntax forms in the set, the more secret bits this set can be used to hide. According to our recent experiment, most sentences of less than ten words possess the syntax set of two forms. Thus, the hiding capacity of these sentences is one bit. As the sentence is longer and more complex, the hiding capacity of this will be increase.

Furthermore, the method we have proposed uses the key-controlled semi-random assignment for syntax forms in the syntax set. The intruders who do not have the key cannot generate the same random sequence. Thus, even though they could have the syntax set, they cannot achieve the exact binary value without having the key. This improves the security strength of our proposed system.

5. Conclusion

This work tries to develop a linguistic steganography approach by combining the statistical parser to parse the sentence, the Shannon-Fano compression method to achieve high payload, and the syntactic method that used a syntax bank to produce the innocent-looking text messages for avoiding the suspicion of an observer. As a result of using Shannon-Fano compression algorithm and the syntax set, our method can conceal longer secret message than the earlier methods. Again, syntax transformation only changes the syntax form of the cover text to hide the desired coded secret message's bits. Thus, this transformation does not destroy the meaning of cover text and retain it as before. Furthermore, the security of our method is enhanced by using the key-controlled semi-random assignment to the syntax forms in the syntax set.

For future work, we will add more syntax transformation methods to achieve more and more efficient and effective system.

References

[1] Aniket M.nanhe, Mayuresh P.Kunjir, Sumedh V.Sakdeo, "Improved Synonym Approach to Linguistic Steganography", http://dsl.serc.iisc.ernet.in/~mayuresh/ImprovedSynonymApproachToLinguisticSteganography.pdf, (see at 15.3.2011).

[2] B. Murphy and C. Vogel, "The syntax of concealment: Reliable methods for plain text information hiding," in Proceedings of the SPIE Conference on Security and Steganography and Watermarking of Multimedia Contents IX, San José, January 2007.

[3] B. Murphy and C. Vogel, "Statistically-constrained shallow text marking: techniques, evaluation paradigm and results," in Proceedings of the SPIE Conference on Security and Steganography and andWatermarking of Multimedia Contents IX, San José, January 2007.

[4] Brecht Wyseur, Karel Wouters, Bart Preneel, "Lexical Natural Language Steganography System with Human Interaction", in Proceedings of The 6th European Conference on Information Warfare and Security, pages 303-312, July 2007.

[5] Ching-Yun Chang, Stephen Clark, "Linguistic Steganography Using Automatically Generated Paraphrases", Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 591–599, Los Angeles, California, June 2010.

[6] Ching-Yun Chang, Stephen Clark, "Practical Linguistic Steganography using Contextual Synonym Substitutionand Vertex Colour Coding", in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-10), pp.1194-1203, Cambridge, MA, October 2010.

[7] Delia Rusu, Lorand Dali, Blaž Fortuna, Marko Grobelnik, Dunja Mladenić, "Triplet Extraction from Sentences", 10th International Multi-conference on Information Society(IS-2007), Ljubljana, Slovenia, October, 2007.

[8] Hasan M. Meral, Emre Sevinç, Ersin Ünkar, Bülent Sankur, A. Sumru Özsoy, Tunga Güngör, "Syntactic tools for text watermarking", in Proceedings of the SPIE Conference on Security and Steganography and Watermarking of Multimedia Contents IX, San José, January 2007.

[9] Hasan Mesut Meral, Bülent Sankur, A. Sumru Özsoy, Tunga Güngör, Emre Sevinc, "Natural language watermarking via morphosyntactic alterations", Computer Speech and Language 23, pages 107–125, 2009.

[10] Hitesh Singh, Pradeep Kumar Singh, Kriti Saroha, "A Survey on Text Based Steganography", in Proceedings of the 3rd National Conference; INDIACom-2009 Computing For Nation Development, February, 2009.

[11] Mercan Topkara, Umut Topkara, Mikhail J. Atallah, "Words Are Not Enough: Sentence Level Natural Language Watermarking", MCPS'06, Santa Barbara, California, USA, October 2006.

[12] http://webspace.ship.edu/cgboer/syntax.html (see at 14.3.2011)

[13] http://www.wikipedia.org (see at 20.2.2011)

[14] http://nlp.stanford.edu/pubs/lex-parser.shtml (see at 28.2.2011)