Syntax Bank and HMAC based Syntactic Steganography Approach for Information Security

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

Abstract

Steganography plays an important role for information security in today digital age. Because of the widespread usages of natural language, we propose a linguistic steganography approach based on syntax bank and HMAC in this paper. The proposed system first uses the combination of most popular two lossless data compression methods to compress the input secret message. At the same time, the system extracts the syntax of the incoming cover text sentence. Then, syntax set creation task constructs the syntax set of the given sentence with the help of the syntax bank. The syntax transformation step then transforms the input sentence into a desired syntax that can represent the key-controlled semi-randomly generated secret bits intended to hide in the sentence. In addition, we apply SHA 512 hash algorithm to generate keyed-hash message authentication code (HMAC) in order to present the identity of the resulting stego text. 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

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out of the usual to establish communication between two parties whose existence is unknown to a possible attacker. It is 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. There are three dimensions in a stego system,

- Payload Capacity: the ratio of hidden information to cover information.
- Robustness: the ability of the system to resist against changes in the cover object.
- 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.

It has found uses 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. 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 [8].

In this paper, a steganographic approach is proposed for linguistic steganography by using the combination of Huffman and Shannon-Fano compressing algorithms, syntax extraction by the statistical Stanford parser, a syntactic method based on the syntax bank, and SHA512 based HMAC.

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 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 use 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 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 use synonym replacement, which converts a message into semantically innocuous text. It also uses 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.

The steganography method that based on Chinese language can be seen in [11]. A Chinese linguistic steganography algorithm is presented by utilizing the existing Chinese information processing techniques. The algorithm is based on the substitution of synonyms and variant forms of the same word. Furthermore, in order to decrease the interaction between the surrounding words and the substituted word, the contextual window of sentence is taken into account by using the disambiguation function of Chinese lexical analysis.

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 [7]. 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.

The syntactic method based on Korean language can be described in [13]. This work is useful for agglutinative languages – such as Korean, Turkish, etc. – of which syntactic constituent order is relatively free. The proposed natural language watermarking method consists of several steps. First, it constructs a syntactic dependency tree of input text. Next, they choose target syntactic constituents to move. Then, they embed watermark bits. If the watermark bit does not coincide with the movement bit of the target constituent, they move the syntactic constituent in the syntactic tree. Finally, from the modified syntactic tree, they obtain a marked text. From the experimental results, they show that the coverage of our method is 75%.

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 [12], 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) AP

> "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." [19]

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 [6], 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. For example:

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

3.1. Transformation-of-Sentences

Transformation-of-Sentences is done in various ways. The nature of the sentences can be changed without changing the meaning of the sentences [18]. The most possible transformation of English is active-passive transformation. This can be used for all sentences and clauses that contain subject, verb, and object. For instance, the clause of "we have received the goods" can be changed into "the goods have been received" without changing the meaning of original clause.

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. Proposed Approach

The proposed system uses the combination of two popular lossless data compression algorithms, Huffman and Shannon-Fano compression, to compress the secret message efficiently. It also utilizes Stanford parser to extract the phrase structure of the input text to get the syntax. Moreover, we propose the syntax bank based steganography approach by doing set creation, syntax transformation steps at the sender's side, and syntax checking steps at the receiver's side. The system also applies SHA 512 based keyed-hash message authentication code (HMAC) to improve the robustness of the system.

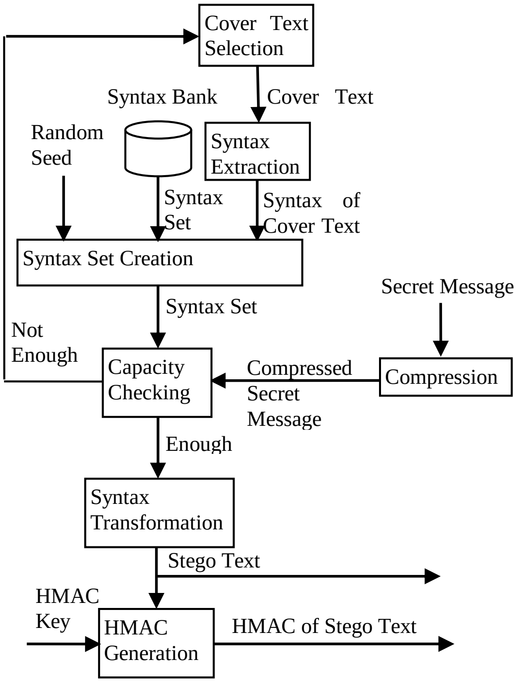

Figure 4.1: Proposed System (Sender Side)

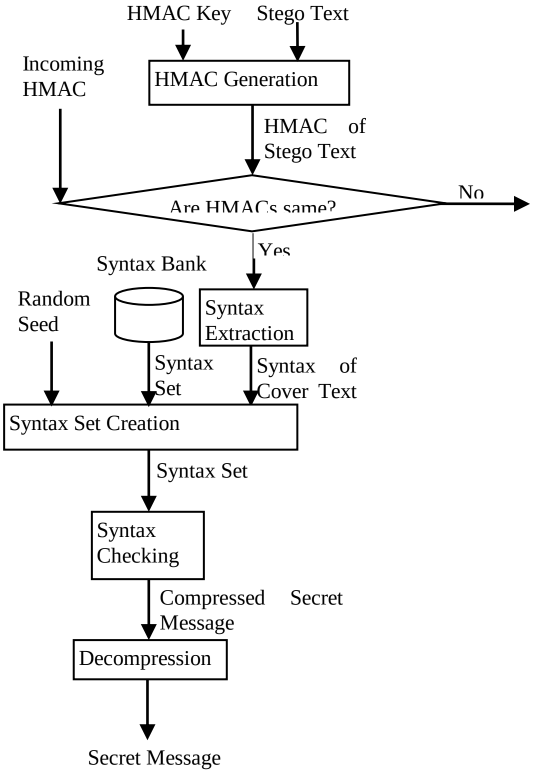

Figure 4.2: Proposed System (Receiver side)

4.1 Compression

This step is the combination of two lossless data compression methods. The first is Huffman compression algorithm. This is an entropy encoding algorithm used for lossless data compression. The process essentially begins with the leaf nodes containing the probabilities of the symbol they represent, and then a new node whose children are the 2 nodes with smallest probability is created, such that the new node's probability is equal to the sum of the children's probability. With the previous 2 nodes merged into one node (thus not considering them anymore), and with the new node being now considered, the procedure is repeated until only one node remains, the Huffman tree. It results a prefix-free codes, that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol that expresses the most common source symbols using shorter strings of bits than are used for less common source symbols. Huffman was able to design the most efficient compression method of this type. But, for a set of symbols with a uniform probability distribution and a number of members which is a power of two, Huffman coding is equivalent to simple binary block encoding, e.g., ASCII coding. [15] To be efficient, our proposed system applies the second compression method with in such case.

The second compression that used in our proposed system is Shannon-Fano method. 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. [16]

4.2 Syntax Extraction

This step uses Stanford parser to extract the phrase structure of the input sentence. This parser 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. [17]

The syntax extraction step modifies the output of this parser, the phrase structure grammar representation of the sentence, as necessary to get the syntax structure of the sentence.

4.3. Syntax Transformation based Text Steganography

This phase is the core of our proposed technique. This step can be divided into two sub-steps: syntax set creation, and syntax transformation at the sender side or syntax checking at the receiver side.

The syntax set creation takes the syntax phrase structure of an input sentence produced by the syntax extraction step as input, constructs and provides a syntax set for this sentence as output.

At the sender side, the capacity checking step checks whether or not the selected cover text have enough hidden capacity for the intended compressed secret message. If so, the syntax transformation step decides which syntax alternative to transform according to the assigned binary sequence, and transforms the input cover text sentence into this chosen syntax. If not, the cover text must be re-chosen.

For the receiver side, the syntax checking step uses the syntax phrase structure of the stego text sentence that is produced by the syntax extraction step and the syntax set that is the output of the syntax set creation step to finds out the corresponding binary sequence.

4.3.1 Syntax Set Creation

The proposed method uses syntax bank that consists of a number of the syntax groups and has already shared between the sender and the receiver. This set creation task takes the syntax phrase structure produced by the above extraction step as input. It then uses this syntax to search for its transformable syntax alternatives group in syntax bank. If there is more than one clause in the input sentence, the syntax set forms by the combination of syntax groups of all clauses in the sentence.

4.3.1.1 Syntax Set

A syntax set is a combination of all available syntax alternatives for all clauses of a sentence. All members of the set are semi-randomly assigned with a unique binary sequence for each. This syntax set's size of a sentence can be calculated by the following equation.

$$L = \prod_{i=1}^{N} M_i \qquad (1)$$

Where

M = the number of syntactic forms for each clause

N = the number of clauses in a sentence

L = the size of syntax set

The number of secret bits which can be hidden in a sentence is $\log_2$ of the size of syntax set of the sentence.

4.3.1.2 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 4.3: The algorithm for generating unique random number

This algorithm can generate the random sequence without repeat. This means that there is exactly one occurrence of a number within the sequence. For example, in the case of random number sequence from 0 to 3, there is no two 2s in the sequence. The sequence will be 0123, 0213, 0312, and so on.

Only the sender and receiver who shared the seed can generate the random sequence of correct order. Even the intruder obtains the syntax set; it cannot be possible to assign the correct binary numbers sequence because of lack of knowledge about the seed to produce the sequence.

4.3.2 Syntax Transformation

This step transforms the input sentence into the desired syntax form. As for a prototype, our system now implemented and tested with only active-passive transformation. This can be done by the following procedure.

- The phrase structure of the sentence produced by the parser is used to define subject (noun phrase that come before verb phrase), verb (verb phrase), object (noun phrase that come after verb phrase), and other complement phrases (such as adverb phrase).
- The main action verb in the verb phrase is then transformed into its past participle form with the help of the verb table. The verb phrase for the passive form of the sentence is constructed by adding the appropriate singular/plural form of helping verb to the past participle form of the main verb.
- The passive sentence is constructed by making direct object into the subject, adding the passive formed verb phrase, and placing the original subject into a propositional phrase beginning with "by".

There are some limitations in interchanging the active sentence into passive form. These are because of the performance of the parser used. For our system, we assume that the parser used, the Stanford parser, is a perfect parser.

4.4 Example case

4.4.1 Compression Step

As an example, the secret word "go to Bago" is compressed as follows:

Table 4.1: Huffman code of "go to Bago"

Character	Frequency	Code
o	3	10
g	2	01
	2	00
t	1	110
B	1	1110
a	1	1111

By using the above codes, the coded secret message is 01 10 00 110 10 00 1111 1110 01 10.

4.4.2 Syntax Extraction Step

Input sentence:
King Bruce watched its movements although he faced his troubles.

Parser output:

```
( ROOT ( S ( NP ( NNP King ) ( NNP Bruce ) ) ( VP ( VBD watched ) ( NP ( NP ( DT the ) ( NN spider ) ( POS 's ) ) ( NNS movements ) ) ( SBAR ( IN although ) ( S ( NP ( PRP he ) ) ( VP ( VBD faced ) ( NP ( PRP$ his ) ( NNS troubles ) ) ) ) ) ) ) ( . . ) ) )
```

Extracted Syntax:

NP	NNP NNP	King Bruce
VP	VBD	watched
NP	DT NN POS NNS	the spider 's movements
SBAR	IN	although
NP	PRP	he
VP	VBD	faced
NP	PRP$ NNS	his troubles

Figure 4.4. Syntax Extraction of the input sentence

4.4.3 Syntax Transformation based Method

Binary intended to hide: ………00……….

Syntax of the incoming cover text sentence:
NP VBD NP although NP VBD NP

Syntax Set:
10 NP VBD NP although NP VBD NP
01 NP VBD NP although NP VBD VBN by NP
00 NP VBD VBN by NP although NP VBD NP
11 NP VBD VBN by NP although NP VBD VBN by NP

Syntax to transform
NP VBD VBN by NP although NP VBD NP

Stego text:
The spider's movements are watched by King Bruce although he faced his troubles…………

Figure 4.5. Sender side

Stego text:
The spider's movements are watched by King Bruce although he faced his troubles…………

Syntax of the incoming stego text sentence:
NP VBD VBN by NP although NP VBD NP

Syntax Set:
10 NP VBD NP although NP VBD NP
01 NP VBD NP although NP VBD VBN by NP
00 NP VBD VBN by NP although NP VBD NP
11 NP VBD VBN by NP although NP VBD VBN by NP

Extracted compressed Secret message:
………00………

Figure 4.6. Receiver side

4.5. SHA-512 based Keyed-Hash Message Authentication Code (HMAC)

HMAC is a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative approved cryptographic hash function, in combination with a shared secret key. The cryptographic strength of HMAC depends on the properties of the underlying hash function [12].

To compute a MAC over the data 'text' using the HMAC function, the following operation is performed:

$$HMAC(K,text)_t = H((K_0 \oplus opad) \| H((K_0 \oplus ipad) \| text))_t \qquad (2)$$

Where

B = Block size (in bytes) of the input to the Approved hash function.
H = An Approved hash function.
ipad = Inner pad; the byte x'36' repeated B times.
K = Secret key shared between the originator and the intended receiver(s).
K0 = The key K after any necessary pre-processing to form a B byte key.
opad = Outer pad; the byte x'5c' repeated B times.
t = The number of bytes of MAC.
text = The data on which the HMAC is calculated; text does not include the padded key. The length of text is n bits, where 0 £ n < 2B - 8B.
|| = Concatenation
⊕ = Exclusive-Or operation. [9]

In this system, we intend to use SHA-512 hash algorithm to produce HMAC. The maximum message size of this algorithm is 2^{128}-1bits and its block size is 1024 bits. The final result is a 512-bit message digest. As the estimated collision resistance strength of any approved cryptographic hash function is half the length of its hash value, it is believed to have collision resistance strength of 256 bits. Again, the estimated preimage resistance strength is 512 bits. [14]

4.6 Experimental Result

The proposed system has implemented and tested as a prototype constructed only for the active-passive transformable sentence. Thus, this fact affects the hidden capacity of our system. The following chart shows the capacity of our system that is implemented only for the active-passive transformation and tested at the natural language text.

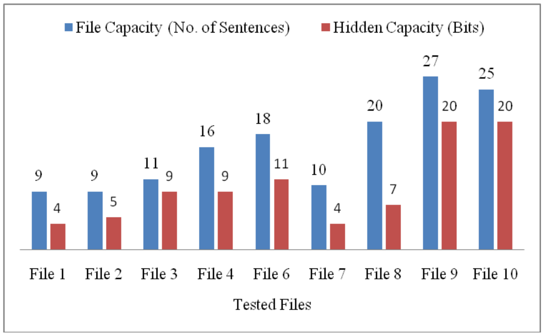

Figure 4.7. Hidden Capacity of proposed system

As an average, the hidden capacity of the proposed system is about 0.6 per sentence. As the hidden capacity of syntax based steganography methods is normally between 0.5 and 1.0 per sentence, the capacity of our method is within the acceptable range. 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.

The imperceptibility of our system can be determined by applying one of the sentence similarity measure techniques because the statistical similarity measures between sentences, based on symbolic characteristics and structural information, could measure the similarity between sentences without any prior knowledge but only on the statistical information of sentences. Here, Word Vector based sentences similarity measure is used to find out the imperceptibility of the system. In this technique, the word vectors of sentences should be constructed first. If the words in $w(s_a)$ and $w(s_b)$ are assigned with weights, s_a and s_b can be represented by the bags of words:

$$\begin{cases} v(s_a) = \{(w_1, w_{a1}), (w_2, w_{a2}), \dots, (w_{i+j}, w_{a(i+j)})\} \\ v(s_b) = \{(w_1, w_{b1}), (w_2, w_{b2}), \dots, (w_{i+j}, w_{b(i+j)})\} \end{cases} \quad (3)$$

Then cosine similarity between sentences can be calculated by

$$Cosine\ (S_a, S_b) = \frac{\sum_{k=1}^{i+j} w_{ak} w_{bk}}{\sqrt{\sum_{k=1}^{i+j} w_{ak}^2}\sqrt{\sum_{k=1}^{i+j} w_{bk}^2}} \quad (4)$$

If a word occurs two or more times in one sentence, the weight of the word is accumulated. The similarity of our system is about 0.85 in average.

The robustness of the system can be achieved by applying SHA-512 based HMAC to the output stego text. Because of this HMAC, the integrity of the incoming stego text can be determined and the robustness can be improved.

5. Conclusion

The proposed method tries to develop a linguistic steganography approach by combining the statistical parser to parse the sentence, Huffman and Shannon-Fano compression methods to reduce the length of secret message that can affect the total number of characters that can be hidden in the sentence, and the syntactic method that used a syntax bank to produce the innocent-looking text messages for avoiding the suspicion of an observer. To improve robustness, the proposed system use SHA-512 based HMAC to identify the integrity of the stego text.

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 meaning of the result stego text sentences is the same as their original cover text sentences because the syntax set of the proposed system is a collection of different syntax alternatives that can produce the same meaning. Due to this retaining appearance and meaning, the proposed method can produce natural looking text as the cover text.

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 strength of our proposed system.

References

[1] A.M.Nanhe, M.P.Kunjir, and S.V.Sakdeo, “Improved Synonym Approach to Linguistic Steganography”, http://dsl.serc.iisc.ernet.in/~mayuresh/ImprovedSynonymApproachToLinguisticSteganography.pdf, (see at 15.12.2011).

[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] B.Wyseur, K.Wouters, and B.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] C.Y.Chang, and S.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.

[6] D.Rusu, L.Dali, B.Fortuna, M.Grobelnik, and D.Mladenić, “Triplet Extraction from Sentences”, 10th International Multi-conference on Information Society(IS-2007), Ljubljana, Slovenia, October, 2007.

[7] H.M.Meral, B.Sankur, A.S.Özsoy, T.Güngör, and E.Sevinc, “Natural language watermarking via morphosyntactic alterations”, Computer Speech and Language 23, pages 107–125, 2009.

[8] H.Singh, P.K.Singh, and K.Saroha, “A Survey on Text Based Steganography”, in Proceedings of the 3rd National Conference; INDIACom-2009 Computing For Nation Development, February, 2009.

[9] Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, USA, “The Keyed-Hash Message Authentication Code (HMAC)”, FIPS PUB 198, FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Issued March 6, 2002.

[10] J.Zhang, Y.Sun, H.Wang, and Y.He, “Calculating Statistical Similarity between Sentences”, Journal of Convergence Information Technology, Volume 6, Number 2. February 2011.

[11] L.Yuling, S.Xingming, G.Can, and W.Hong, “An Efficient Linguistic Steganography for Chinese Text”, IEEE International Conference on Multimedia and Expo, 2007.

[12] M.Topkara, U.Topkara, and M.J.Atallah, “Words Are Not Enough: Sentence Level Natural Language Watermarking”, MCPS’06, Santa Barbara, California, USA, October 2006.

[13] M.Y.Kim, “Text Watermarking by Syntactic Analysis”, 12th WSEAS International Conference on Computers, Heraklion, Greece, July 2008.

[14] Q.Dang, “Recommendation for Applications Using Approved Hash Algorithms”, NIST Special Publication 800-107, February 2009.

[15] http://en.wikipedia.org/wiki/Huffman_coding (see at 20.12.2011)

[16] http://en.wikipedia.org/wiki/Shannon-Fano_coding (see at 20.12.2011)

[17] http://nlp.stanford.edu/pubs/lex-parser.shtml (see at 28.12.2011)

[18] http://www.english-for-students.com/Transformation-of-Sentences.html (see at 10.12.11)

[19] http://webspace.ship.edu/cgboer/syntax.html (see at 14.12.2011)