Modified Linguistic Steganography Approach by Using Syntax Bank and Digital Signature

Ei Nyein Chan Wai and May Aye Khine

***Abstract*—In today digital age, there are more demands to improve techniques for information security. Steganography is one of the popular areas in information protection in order to establish communication between two parties whose existence is unknown to a possible attacker. Among the variety of steganographic methods, linguistic approach is concerned with hiding information in natural language text. Here, our proposed system uses syntax transformations to hide the intended secret message into the cover text. The transformation bases on a syntax bank that consists of a number of syntax sets. The syntax set is a set of all available syntax forms of input sentence. Moreover, our system utilizes shannon-fano compression algorithm, semi-random number assignment, and SHA-512 hash algorithm and digital signature algorithm (DSA) based digital signature to support the capacity, robustness, and innocent-looking capabilities.**

***Index Terms*—Compression, digital signature, linguistic steganography, syntax transformation.**

I. Introduction

Nowadays, information exchange or distribution such as email, e-book and so on, plays a vital role in people's daily activity with the help of the internet. Together with this increased growth of information exchange, information security becomes more important in data storage and transmission. Many researchers explode solutions to achieve this, and steganography becomes one of these solutions.

The word steganography is of Greek origin and means "concealed writing". 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 has found use in variously in military, diplomatic, personal and intellectual property applications.

Steganography has been widely used since historical times until the present day. In ancient greece, the hidden messages were tattooed on a slave's (the messagers') 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. Moreover, Julius Caesar used cryptography to encode political directives. During World War II, a spy for the Japanese in New York City sent information to accommodation addresses in neutral South America by the stegotext within the 'doll' orders.

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 [1].

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

Modern steganography includes the concealment of information within computer files. Electronic communications may include steganographic coding inside of a transport layer, such as a document file, stay image file, audio files, video files, program or protocol. Among them all, 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.

The structure of text documents is identical with what we observe, while in other types of documents such as in picture, the structure of document is different from what we observe. Therefore, in such documents, we can hide information by introducing changes in the structure of the document without making a notable change in the concerned output [2].

Text-steganogrphy proceeds according to the following scheme:

- A secret message (embedded, hidden data) is concealed in cover-text using an embedding algorithm to produce a stego-text.
- The stego-text is then transmitted over a communication channel (Internet).
- Upon its delivery, the secret message is recovered using an extracting algorithm.
- The embedding and the extracting algorithms are augmented by the so called a stego-key to encrypt and decrypt the hidden data respectively [3].

Text steganography is broadly classified into the two categories; Linguistic steganography which is further divided into semantic and syntactic method and format based steganography which is further divided into line-shift, word-shift, open-space and feature encoding [4] as described in the fig. 1.1.

Manuscript received November 22, 2011; revised December 8, 2011.

Authors are with University of Computer Studies, Yangon, Myanmar (e-mail: einyeinchanwai@ gmail.com).

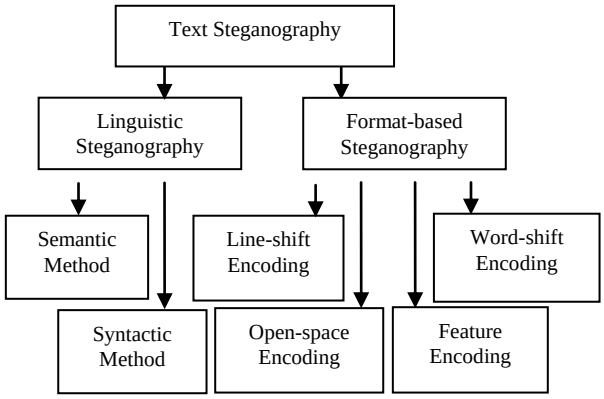

Fig. 1.1. Types of text steganography

In addition, the different text steganography methods have their own drawbacks. For instance, line-shifted and word-shifted methods can be detected and modified by using the ordinary processing tools. Again, semantic method relies on synonyms with same meaning, but there are only a few words that have the same meaning whenever and wherever they are used because the usage of a word can be changed according to its time and place in the sentence.

Thus, we intend to propose linguistic based approach in this paper. Furthermore, our proposed system emphasizes on English language because it is the most widely used language around the world and over the Internet.

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. In addition, we apply SHA 512 hash algorithm and Digital Signature Algorithm (DSA) to generate digital signature in order to represent the identity of the resulting stego text. 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.

II. 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 another word that possesses mostly the same meaning as 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.

A. Lexical Steganography

In [5], the writers used synonym replacement by using a word dictionary to get synonym. Furthermore, the secret text to be hidden is first compressed by huffman compression algorithm to be consumed in selection of synonyms.

In [1], Brecht wyseur, karel wouters, and bart preneel proposed a linguistic steganography based on word substitution over an IRC channel. The generation of the word substitution table is based on a session key and used synonyms from a public thesaurus.

Ching-yun chang and stephen clark proposed a method for checking the acceptability of paraphrases in context in [6] 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 [7] 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.

B. Syntactic Steganography

According to our recent study, B. Murphy and C. Vogel mainly proposed syntactic methods for steganography. In [8], 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 [9], they also presented three natural language marking strategies: lexical substitution, adjective conjunction swaps, and relativiser switching.

The other people explored the morphosyntactic tools for text watermarking and developed a syntax-based natural language watermarking scheme in [10]. 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 [11], the authors developed a morphosyntax-based natural language watermarking scheme in which 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.

C. Combining Lexical and Syntactic Steganography

Some work in the steganography combine lexical and syntactic methods. These 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. It uses XTAG parser for parsing, dependency tree generation and linguistic feature extraction and RealPro for natural language generation.

III. 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."

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 [13], 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 [14]. For instance,

- After we have received the goods, we will settle the account.

Finally, a sentence can also have two or more main (independent) clauses, joined by coordinating conjunctions [14]. For example,

- Either I go or he goes.

A. 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 [15]. 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.

IV. Proposed Approach

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 syntax forms within the syntax set of the original sentence. This transformed syntax 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. Moreover, the digital signature of the stego text is generated by using SHA-512 hash algorithm, DSA algorithm and sender's private key to identify the integrity of this stego text.

When the stego text reaches to the receiver side, it is firstly checked whether the signature produced by sender's public key is the same as the original signature went together with the stego text. If so, the stego text is 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. If the signatures are different, the receiver can suspect the integrity of the stego text. So, the current stego text is drooped and asked the sender to resend again the message.

The sender's side and the receiver's side of the proposed system are shown in the fig. 4.1 and 4.2 respectively.

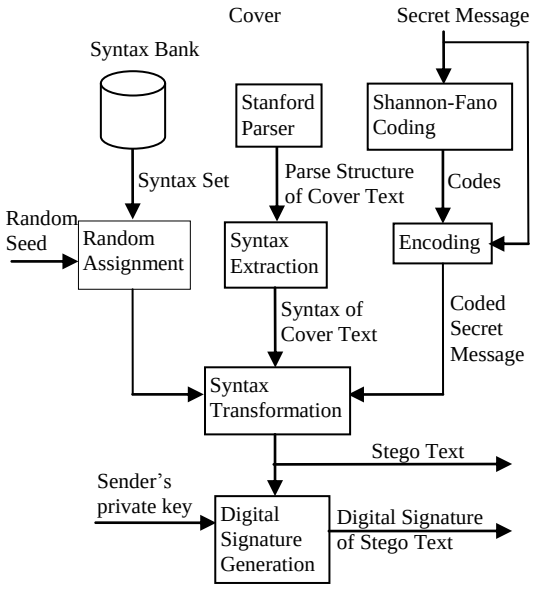

Fig. 4.1. Proposed system (sender side)

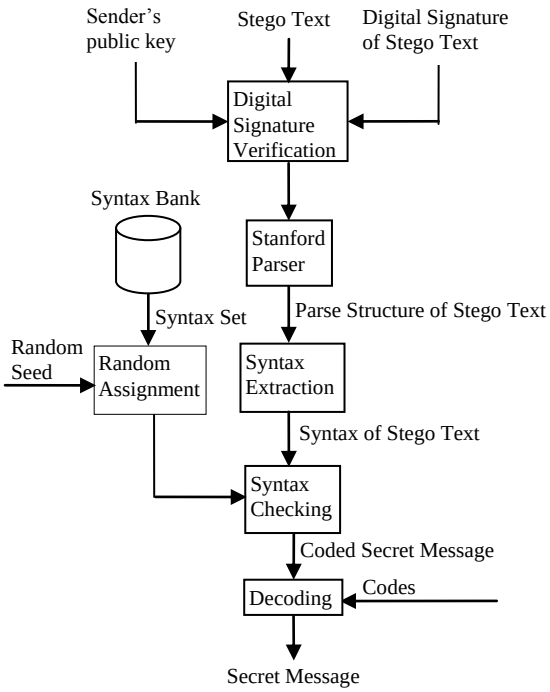

Fig. 4.2. Proposed system (receiver side)

A. 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 [16].

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

TABLE 4.1: SHANNON-FANO CODE OF “MESSAGE”

Character	Frequency	Code
e	2	0
s	2	10
a	1	110
g	1	1110
m	1	1111

By using the above codes, the coded secret message is `1111 0 10 10 110 1110 0`.

B. 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 [17].

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.

C. Syntactic Steganography using Syntax Bank

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.

For example, the cover text of the secret message “message” is as follows:

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

Fig. 4.3. Example cover text

Then, the secret message after hiding the first two bits of the coded secret message, 11, is shown below.

After the goods have been received, the accounts will be settled ……

Fig. 4.4. Example stego text

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.

a. 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.

In our system, we have implemented a table that contains pairs of English clause syntax and its set number. When input sentence’s syntax is available after passing through the Stanford parser, our system divides this syntax clause by clause. Then, it searches the syntax of first clause in the syntax bank. If found, the system extracts all syntax forms with same set number as the input first clause. By doing so, we can get the syntax set for a clause, and this is done for all clauses in the input sentence. The syntax forms of all clauses in the sentence are then combined to produce the syntax set for the whole sentence.

b. 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 generate unique random (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;
```

Fig. 4.5. 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.

c. 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. For example: "play" is transformed into "played". 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.

d. SHA-512 and DSA based Digital Signature

A digital signature is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides the capability to generate and verify signatures. Signature generation makes use of a private key to generate a digital signature. Signature verification makes use of a public key which corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. Public keys are assumed to be known to the public in general. Private keys are never shared. Anyone can verify the signature of a user by employing that user's public key. Signature generation can be performed only by the possessor of the user's private key.

A hash function is used in the signature generation process to obtain a condensed version of data, called a message digest. The message digest is then input to the digital signature (ds) algorithm to generate the digital signature. The digital signature is sent to the intended verifier along with the signed data (often called the message). The verifier of the message and signature verifies the signature by using the sender's public key. The same hash function must also be used in the verification process [18].

In this system, we intend to use SHA-512 hash algorithm to produce message digest for generting the digital signature. The maximum message size of this algorithm is 2128-1bits 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 [19].

D. Experimental Result

We tested our system with 11 text files as cover text. The following figure shows the hidden capacity of these files.

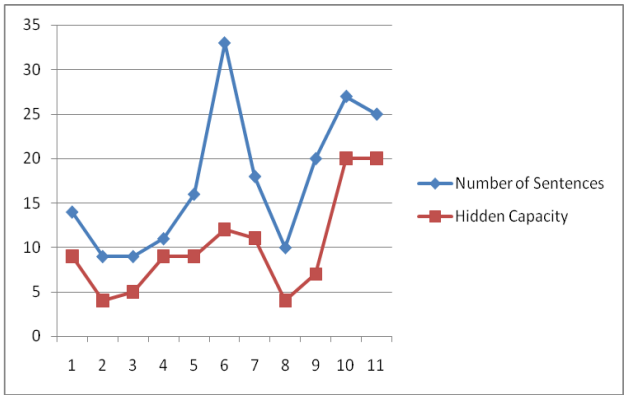

Fig. 4.6. Hidden capacity of tested files

The average payload capacity is about 0.6 per sentence. As the hidden capacity of syntax based 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 the proposed system is measured by the judgments of 20 people. 95% of the judgments said that the input cover text and the output stego text have the same meaning. Therefore, the stego text maintains the innocent-looking property of the cover text.

The robustness of the system can be achieved by applying SHA-512 based digital signature to the output stego text. An adversary, who does not know the private key of the sender, cannot generate the correct signature. So, the receiver can determine the integrity of the incoming stego text by verifying the signature with the sender's public key.

V. Conclusion and Future Work

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 forms 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.

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

References

[1] B. Wyseur, K. Wouters, and B. Preneel, "Lexical natural language steganography system with human interaction," in *Proc. 6th European Conf. on Information Warfare and Security*, pages 303-312, July 2007.

[2] M. Shirali-Shahreza and S. Shirali-Shahreza, "High capacity persian/arabic text steganography, " *Journal of Applied Sciences*, vol. 8, no. 24, pp. 4173-4179, 2008.

[3] R. Jabri, B. Ibrahim, and H. Al-Zoubi, "Information hiding: a generic approach," *Journal of Computer Science*, vol. 5, no. 12, pp. 933-939, ISSN 1549-3636, 2009.

[4] H. Singh, P.K. Singh, and K. Saroha, "A survey on text based steganography," in *Proc. 3rd National Conf.; INDIACom-2009 Computing For Nation Development*, February, 2009.

[5] A. M. Nanhe, M.P. Kunjir, and S. V. Sakdeo, "Improved synonym approach to linguistic steganography," Available: http://dsl.serc.iisc.ernet.in/~mayuresh/ImprovedSynonymApproachToLinguisticSteganography.pdf, (see at 15.3.2011).

[6] C. Y. Chang and S. Clark, "Linguistic steganography using automatically generated paraphrases," presented at the 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.

[7] C.Y. Chang and S. Clark, "Practical linguistic steganography using contextual synonym substitution and vertex colour coding," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP-10)*, pages 1194-1203, Cambridge, MA, October 2010.

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

[9] B. Murphy and C. Vogel, "Statistically-constrained shallow text marking: techniques, evaluation paradigm and results," in *Proc. SPIE Conf. on Security and Steganography and Watermarking of Multimedia Contents IX*, San José, January 2007.

[10] H. M. Meral, E. Sevinç, E. Ünkar, B. Sankur, A. S. Özsoy, and T. Güngör, "Syntactic tools for text watermarking," in *Proc. SPIE Conf. on Security and Steganography and Watermarking of Multimedia Contents IX*, San José, January 2007.

[11] H. M. Meral, B. Sankur, A. S. Özsoy, T. Güngör, and E. Sevinc, "Natural language watermarking via morphosyntactic alterations," *Computer Speech and Language*, vol. 23, pp. 107–125, 2009.

[12] M. Topkara, U. Topkara, and M. J. Atallah, "Words are not enough: sentence level natural language watermarking," presented at the MCPS'06, Santa Barbara, California, USA, October 2006.

[13] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenić, "Triplet extraction from sentences," presented at the 10th International Multi-conference on Information Society (IS-2007), Ljubljana, Slovenia, October, 2007.

[14] http://webspace.ship.edu/cgboer/syntax.html (see at 14.7.2011)

[15] http://www.english-for-students.com/Transformation-of-Sentences.html (see at 10.9.11)

[16] http://www.wikipedia.org (see at 10.7.2011)

[17] http://www-nlp.stanford.edu/software/lex-parser.shtml (see at 14.7.2011)

[18] *Digital Signature Standard (DSS), FIPS PUB 186-1*, 1998.

[19] Q. Dang, "Recommendation for applications using approved hash algorithms," NIST Special Publication 800-107, February, 2009.