Al-Qadisiyah Journal of Pure Science

Volume 28 | Number 1 Article 18

2023

# Steganography using Syntactic Method

Saad Nasser Al.Azzam
*Cyber Security Researcher University of Bisha Faculty of Computing and Information Technology,*
snazzam.199@gmail.com

Fahad Ali Al-Qarni
*Dean of the College of Computing and Information Technology Associate Professor, University of Bisha*

ARTICLE

# Steganography Using Syntactic Method

Saad N. AlAzzam*, Fahad A. Al-Qarni

College of Computing and Information Technology, University of Bisha, Saudi Arabia

**Abstract**

**Information security, which is becoming more crucial as information sharing plays an increasingly significant part in the day-to-day activities of today's people, is one of the methods that may be used for information security. For the sake of information security, we present a method to linguistic steganography that is based on syntactic banks.**

**While the unprocessed input a phrase structure of the sentence is produced by parsing the sentence using the Stanford parser, which can create a phrase structure of the sentence.**

**They create the grammatical structure of the sentence. The input secret message is encoded using Shannon-Fano coding in order to achieve compression. With the absolute minimum, the maximum feasible total bit length after that, the job of syntax transformation looks through the syntax set of the provided.**

**The key-controlled phrase already exists in the syntactic bank and modifies it so that it corresponds to the required syntax that may express the key-controlled This is hidden information that is created semi-randomly and inserted into the text at discrete points. The stego text that is produced as a consequence will still remain the input text is given the appearance of being innocent after a semantically unmodified syntactic transformation is applied to it.**

**Again, we aim to use the SHA-512 hash method to generate a keyed-hash message authentication code (HMAC) to secure the communication. This increases the consistency of the stage text that was produced.**

*Keywords:* Linguistic steganography, Text information hiding, Syntax, Data compression, Random number generation

## 1. Introduction

The term "steganography" refers to the practice of "concealed writing," which is used to create communication between two parties whose existence is hidden from a potential assailant. In addition, it is a phrase that may be given to a variety of procedures that will conceal a message inside an item, so that the observer will not be able to recognize the message that has been concealed.

From ancient times up to the present day, it has been put to use in a variety of ways, including those pertaining to the military, diplomacy, personal life, and intellectual property.

There are three dimensions in a stage system, Payload capacity is the amount of secret data relative to the amount of data used to cover it up.

2) Robustness, or the system's resistance to changes in the cover object.

Thirdly, a created stego object's prospective imperceptibility, or indistinguishability from other items of the same category, is evaluated. [1].

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

Information may be hidden inside computer files using digital steganography, which is also known as contemporary steganography. Steganography is used in today's period. It is possible to conceal many forms of covert messages, such as audio, images, and texts, inside various types of cover media, such as audio, video, images, texts, and so on.

Some examples of cover media include: Texts are among the many diverse cover media that are extensively utilized in a variety of activities. However, it is also the most difficult sort of steganography since, in large part, there is not a lot of redundant information in a text file. This makes it a challenge to conceal information in this way.

Received 15 April 2022; revised 13 May 2022; accepted 22 May 2022.
Available online 26 May 2023

* Corresponding author.
E-mail address: Snazzam.199@gmail.com (S.N. AlAzzam).

https://doi.org/10.29350/2411-3514.1017

There are two main ways to categorise steganography; the first is the linguistic approach, which refers to the use of written natural language to conceal secret messages; the second is the format-based approach, which refers to the use of the physical formatting of text as a place to hide information.

Both the former and the latter may be further subdivided into line-shift, word-shift, open-space, and feature encoding, respectively. The semantic method and the syntactic approach are examples of the former [2].

This research proposes a steganographic approach for linguistic steganography that makes use of the Shannon-Fano compression algorithm, the statistical Stanford parser, and a syntactic strategy based on the syntax bank.

Additionally, a keyed-hash message authentication code (HMAC) is generated using the SHA-512 hash algorithm, which uniquely identifies the crafted script.

In the following section (section 2), a concise summary of the various existing linguistic steganography methods will be provided. The grammar of the language is going to be broken down in Section 3. In Section 4, we will discuss our suggested procedure. In the fifth and last part, the conclusion as well as recommendations for further research will be presented.

## 2. Linguistic steganography

Making changes to a cover text in order to embed information is the focus of linguistic steganography, which aims to avoid grammatical errors and unnatural-sounding writing without sacrificing either.

The majority of linguistic steganography techniques either rely on lexical (or semantic) changes, syntactic transformations, or some mix of the two. The most common kind of lexical steganography is known as the synonym replacement approach.

It does so by replacing the original term with another word that has a meaning that is, to a large extent, equivalent to that of the original word. The grammatical style of the original phrases is altered by the use of syntactic procedures. Aside from that, it involves changing words in a way that doesn't change the meaning of the original sentence.

### *2.1. Lexical steganography*

The authors of [3] used the technique of synonym substitution by using a word dictionary in order to get synonyms. Also, the secret text that needs to be hidden is first put into a smaller size using the Huffman Compression Algorithm so that it can be used to choose synonyms.

A linguistic steganography that is based on word replacement was suggested by Brecht Wiser, Karel Wolters, and Bart Preen in [1] This steganography would be used via an IRC channel. The word replacement table is made by taking synonyms from a public thesaurus and putting them together with the help of a session key.

### *2.2. Syntactic steganography*

Our most recent research indicates that B. Murphy and C. Vogel are primarily responsible for proposing syntactic strategies for steganography. In [4], they investigated two grammatical occurrences in English that are very predictable and somewhat widespread and that might be employed in data concealment. This is the switching of complementizers and relativizes, which is caused by a well-known method called syntactic parsing.

Others looked at morphosyntactic text watermarking tools and came up with a syntax-based natural language watermarking technique.

Which may be found in [5]. A syntactic tree diagram is generated from the unmarked text using a transformation. The syntactic hierarchy and functional connections are both encoded in this figure.

The watermarking application operates on the sentences while they are in syntax tree format, making binary alterations while Wordnet is in charge. The goal is to avoid any “semantic drops” that may occur otherwise.

The authors of [6] devised a morphosyntax-based natural language watermarking system. In this method, a text is first turned into a syntactic tree diagram, which is then used to make the hierarchies and the functional relationships clear. In order to prevent semantic drops, the watermarking programme acts on the sentences while they are still in the syntax tree structure. It then makes binary changes while still being told what to do by Wordnet and Dictionary.

### *2.3. Combining lexical and syntactic steganography*

Certain works of STEGANOGRAPHY combine syntactic and lexical analysis. These methods, which operate on a phrase-by-phrase basis, may be used to hide the necessary information.

The strategy that is presented in [7] operates at the sentence level while simultaneously using a watermarking method that operates at the word level. Real Pro is used to make natural language, while the XTAG parser is used to parse, make dependency trees, and pull out language features.

## 3. Syntax of language

Syntax is the set of conventions that govern how words are combined in a given language. Word order is another name for syntax. Phrases are generated when words of several parts of speech are put together, and this includes not just propositional phrases but also adjectival and adverbial ones. Rules for phrase structure may be thought of as a graphical representation of how a sentence is put together. S- > NP VP is an example of this. Each sentence consists of a noun phrase and a verb phrase.

Most current parsers will produce the aforementioned phrase structure. The aforementioned noun phrases that make up the structure are treated as either the subject or the object in a form known as subject-verb-object. Multiple research has been done to figure out how to detect the subject, verb, and object of a sentence based on its phrase structure.

In [8], Several of the most well-known syntactical parsers for the English language are used to extract subject predicate-object (subject-verb-object) triplets from English phrases. The Stanford Parser, Open NLP Parser, Link Parser, and Minipar are all examples of such tools.

Furthermore, a clause is the fundamental unit of any given phrase. A verb and a noun serving as the clause's subject are required components. But more than one clause may exist in a single sentence. The main clause, often called the independent clause, may be followed by one or more subordinate clauses [9].

For example, “Once we have confirmed receipt of the items, we will proceed to pay the bill.” [9] A sentence may also be composed of two or more primary (independent) clauses that are connected to one another by coordinating conjunctions. As an example, one can say, “Either I go or he goes."

## 4. Proposed approach

The procedure begins with the parser analysing the cover text and the secret message being compressed using the Shannon Fano method. The parsed sentence from the cover text is then transformed into one of the syntactic forms included in the lexical inventory of the original phrase.

The longest binary sequence in the compressed form of the secret message's binary representation has been attached to this specific changed syntax. Each word in the cover text goes through the procedures outlined above as long as there is a compressed secret message remaining to hide.

The cover text becomes the stage text after there is no longer any binary sequence to hide, and both the cover text and the codes that compressed the secret are prepared for transmission through the communication channel. The secret is hidden in the stage text until there is no more binary sequence to conceal it.

The HMAC of the stage text is created using the SHA-512 hash algorithm and the secret key that has been previously communicated between the sender and the receiver to ensure the validity of the stage text. The purpose of this is to protect the integrity of the script for the theatre.

The first thing that happens when the stage text arrives at its destination is that it is checked to see whether the HMAC it was sent with is the same as the HMAC it was sent with when it was produced using the shared key.

When this occurs, the stego text is sent through a parser to figure out its grammar. The next step is a syntactic check, which identifies, phrase by phrase, the stego text's corresponding syntax set. In this stage, we are also tasked with determining the corresponding binary sequence.

If you follow these steps for each phrase of the stego text, the binary form of the compressed secret message will be maintained. This data is then decompressed using the accompanying codes.

If the HMACs don't match, the recipient has grounds to doubt the legitimacy of the stego message. This has screwed up the existing stego text, thus the sender has been prompted to try sending it again.

### *4.1. Shannon-Fano algorithm*

A prefix code may be generated using this technique, which takes into account both a set of symbols and the probabilities attached to those symbols. After sorting the symbols from most likely to least likely, they are divided in half such that their combined probability is as close to 50/50 as is realistically possible.

Then, the initial digits of the codes for each symbol are assigned. Any sets with more than one remaining member will have the same procedure applied to them again to determine the next digits of their codes.

### *4.2. The Stanford Parser*

Here we see a Java implementation of a probabilistic NLP parser. A programme called a natural language parser analyses text in an effort to determine its grammar. It achieves this by using the linguistic information it has acquired from human

parsing to the problem of figuring out how to best understand novel sentences. These statistical parsers continue to make errors, but for the most part they are rather accurate.

This parser's output is a phrase-structure grammar representation of the sentence, which is fed into the sender's syntactic transformation phase and the receiver's syntactic verification step.

### *4.3. Syntactic steganography using syntax bank*

The proposed approach takes use of a syntactic bank, which is a collection of syntaxes sets previously exchanged between the sender and the receiver. Each conceivable syntactic form of a sentence is assigned a binary number in a semi-random fashion, and the resulting collection is called a syntax set.

It turns out that the number of hidden bits in a sentence is directly related to the size of the syntax set in which the sentence's initial syntax occurs. Thus, as the number of possible syntaxes grows, so does the potential number of concealed bits. If the input sentence includes many clauses, the syntax set will contain syntax for the whole phrase as well as syntax for each component separately.

#### *4.3.1. Key-controlled semi-random number assignment*

In order to generate the unique random sequence that is assigned to the set's syntactic rules, the sender and the receiver must first exchange a key. With this method, it is possible to produce the random sequence without any recurring patterns.

The right order of the random sequence can only be generated by the sender and the recipient, provided that they share the seed. Even if the intruder is able to access the syntax set, since they lack information on the seed, it is impossible for them to give the right binary number sequence. Here is a description of the way to make random integers that are all different from each other.

```
function generateUniqueRandom (Long seed, int max) return 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;
```

#### *4.3.2. Syntax transformation*

During this stage, the phrase that was provided as input is modified into the required syntactic form. The active–passive transition is the one with the greatest number of possibilities. This is applicable to any and all sentences and clauses that include a subject, verb, and object in their construction.

In addition to that, there is also a possibility to switch the order of the clauses in the sentence. In addition to this, there may be a great number of different methods to change the phrase while preserving its meaning. Some examples include topicalization, adverb displacement, and other similar techniques.

### *4.4. SHA-512 based keyed-hash message authentication code (HMAC)*

Message Authentication Code based on SHA-512 for Message Encryption (HMAC) The HMAC protocol is a method for authenticating messages that makes use of cryptographic hash functions. HMAC may be used in conjunction with any iteratively validated cryptographic hash function, as long as a shared secret key is also present. The ciphertext-cracking prowess of HMAC is directly proportional to the characteristics of the underlying hash function.

The SHA-512 hash algorithm is going to be the one that we employ to generate HMAC for this system. The largest size of a message that can be processed by this method is 2128 bits, while the maximum size of a block is 1024 bits. The end result is a message digest with 512 bits of data.

It has a collision resistance strength of 256 bits and an estimated preimage resistance strength of 512 bits. Both of these values are in bits.

### *4.5. Experimental result*

During the time this article was being written, we put our system through its paces, using six text files containing a total of two hundred phrases as cover text. The typical capacity of the payload, expressed in bits per phrase, is around 0.6. The capacity of our technique is within the permitted range, since the hidden capacity of syntax-based algorithms is typically between 0.5 and 1.0 bits per phrase.

The payload capacity of the system that we have presented may be increased by including more ways of transformation. The number of different syntactic forms our system can handle will directly affect how well it works.

The opinions of twenty different people were collected to determine how imperceptible the

suggested system is. 95% of the decisions say that the meaning of the cover text and the meaning of the stego text are the same.

Applying an HMAC algorithm that is based on SHA-512 to the output stego text is a method that may be used to make the system more resistant. Because of this HMAC, it is possible to tell if the steganographic text that has been received has kept its integrity. This, in turn, makes it possible to make the robustness stronger.

## 5. Conclusion and future work

As our approach is syntax-based rather than format-based, it will not affect the visual presentation of the cover text in any way. The resulting stego text sentences retain the same meaning as their corresponding cover text phrases.

This is because the syntax set of the proposed system is a trove of several syntaxes that might all result in the same meaning. Since the proposed method retains both the text's look and its content, it may produce language that feels natural enough to use as the cover text.

In addition, the proposed technique takes use of semi-random assignment based on key values for the syntactic forms included in the syntax set. Assuming the intruder has the syntax set up, without knowing the key, they will not be able to get the correct binary value. Because of this, the proposed system that we have is enhanced.

## References

[1] Wyseur B, Wouters K, Preneel B. Lexical natural language steganography systems with human interaction. In: Proceedings of the 6th European conference on information warfare and security; 2008. p. 313–20.
[2] Singh H, Singh PK, Saroha K. A survey on text based steganography. In: Proceedings of the 3rd National Conference, Vol. 3. Bharati Vidyapeeth's Institute of Computer Applications and Management; 2009, February. p. 332–5. 3.
[3] Nanhe AM, Kunjir MP, Sakdeo SV. Improved synonym approach to linguistic steganography design and proof-of-concept implementation 2008.
[4] Murphy B, Vogel C. The syntax of concealment: reliable methods for plain text information hiding. In: Security, steganography, and watermarking of multimedia contents IX, 6505. SPIE; 2007, February. p. 351–62.
[5] Meral HM, Sevinc E, Sankur B, Özsoy AS, Güngör T. Syntactic tools for text watermarking. In: Security, Steganography, and Watermarking of Multimedia Contents IX, 6505. SPIE; 2007, March. p. 339–50.
[6] Meral HM, Sankur B, Özsoy AS, Güngör T, Sevinç E. Natural language watermarking via morphosyntactic alterations. Comput Speech Lang 2009;23(1):107–25.
[7] Topkara M, Topkara U, Atallah MJ. Words are not enough: sentence level natural language watermarking. In: Proceedings of the 4th ACM international workshop on Contents protection and security; 2006, October. p. 37–46.
[8] Rusu D, Dali L, Fortuna B, Grobelnik M, Mladenic D. Triplet extraction from sentences. In: Proceedings of the 10th International Multiconference" Information Society-IS; 2007, October. p. 8–12.
[9] see at 14.7.2011, http://webspace.ship.edu/cgboer/syntax.html.