SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2011; **4**:697–718
Published online 25 May 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.208

RESEARCH ARTICLE

Matlist: Mature Linguistic Steganography Methodology

Abdelrahman Desoky

Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, U.S.A.

ABSTRACT

The generated text by Natural Language Generation (NLG) and template systems is meaningful and looks legitimate. Therefore, the Mature Linguistic Steganography Methodology (Matlist) employs NLG and template techniques along with Random Series values (RS), e.g. binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, etc., of Domain-Specific Subject (DSS) to generate noiseless text-cover. This type of DSS, e.g. financial, medical, mathematical, scientific, economical, etc., has plenty of room to conceal data and allows communicating parties to establish a covert channel such as a relationship based on the profession of the communication parties to transmit a text-cover. Matlist embeds data in a form of RS values, function of RS, related semantics of RS, a combination of these, etc. Unlike synonym-based approach, Matlist does not preserve the meaning of text-cover every time it is used. Instead, Matlist Cover retains different legitimate meaning for each message while it remains semantically coherent and rhetorically sound. The presented implementation, validation, and experimental results demonstrate that Matlist is capable of accomplishing the steganographical goal with higher bitrate than all other linguistic steganography approaches. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS
steganography; linguistic steganography; security

***Correspondence**
Abdelrahman Desoky, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, U.S.A.
E-mail: abd1@umbc.edu

1. INTRODUCTION

Linguistic steganography is the scientific art of avoiding the conception of suspicion in covert communications by concealing data in a linguistic-based textual cover. The goal is not to hinder an adversary from decoding the hidden message but to prevent the arousal of suspicion in covert communications. If suspicion is raised when using any steganographic technique, the goal of steganography is defeated regardless of whether or not a plaintext is revealed [1–4]. The concerns of contemporary linguistic steganography approaches may be concluded as follows. Text-cover may contain unusual patterns or numerous detectable flaws (noise) such as incorrect syntax, lexicon, rhetoric, and grammar. In addition, the content may be meaningless and semantically incoherent. Obviously, such flaws can raise suspicion during the covert communications. The bitrates are very minimal. Contemporary efforts are focused on how to hide a message and not on how to hide the transmittal of a hidden message. If contemporary approaches can fool a computer examination, fooling a human examination may appear to be extremely difficult. Not enough attention is given to these faulty issues. A successful linguistic steganography approach must be capable of passing both computer and human examinations. These concerns along with the quality of DSS that is based on RS, and the advances of NLG and template systems have motivated the development of Matlist methodology.

The output of both linguistic steganographical schemes and NLG systems is text. However, their goals are totally different. The goal of linguistic steganographical schemes is to conceal information in non-legitimate text to communicate covertly. On the other hand, the goal of NLG systems is to represent legitimate text either by an on-line-display or audio speech [5]. In the following subsections, a brief review of prior work on linguistic steganography, NLG systems, and Matlist *versus* previous work is presented.

1.1. Linguistic steganography

Since the 20th century, the development of linguistic steganography has been minimal. These approaches can be categorized as follows.

- Series of characters and words: During World War I, the Germans communicated covertly using a series of characters and words known as null-cipher [6–8]. A null-cipher is a predetermined protocol of character and word sequence that is read according to a set of rules such as read every seventh word or read every ninth character in a message.
- Statistical-based: Wayner introduced the mimic functions approach [9,10], which employs the inverse of the Huffman Code by inputting a data stream of randomly distributed bits to produce text that obeys the statistical property of a particular normal text. Therefore, the generated text by mimic functions is resilient against statistical attacks. Mimic functions can employ the concept of both Context Free Grammars (CFG) and van Wijnaarden grammars to enhance the output.
- Synonym-based: Chapman and Davida introduced a steganographic scheme consisting of two functions called NICETEXT and SCRAMBLE that uses a large dictionary, which was later enhanced [11–15]. This approach uses a piece of text to manipulate the process of embedding a message in a form of synonym substitutions. This process preserves the meaning of text-cover (the original piece of text) every time it is used. The synonyms-based approach attracted the attention of numerous researchers within the last decade: Winstein [16,17], Bolshakov *et al.* [18,19], Calvo *et al.* [20], Chand *et al.* [21], Nakagawa [22], Niimi *et al.* [23], Bergmair *et al.* [24–26], Topkara *et al.* [27], Murphy *et al.* [28], and Atallah *et al.* [29,30].
- Noise-based: Grothoff *et al.* introduced the translation-based steganographic Scheme [31–33] to hide a message in the errors (noise) that are generated by a Machine Translation (MT). This approach embeds a message by performing a translation substitution procedure on the targeted translation using translation variations of various MT systems. In addition, it inserts popular errors of MT systems and also uses synonym substitutions in order to increase the bitrate. In the same course of noise-based approach, Topkara *et al.* developed a steganographic scheme to hide information by employing the errors such as typos and ungrammatical abbreviations in a noisy text (e.g. emails, blogs, forums, etc.) [34]. Shirali-Shahreza *et al.* introduced abbreviation-based Scheme [35] to conceal data in the short message service (SMS) of the services used in mobile phones, which is widely used all over the world. Due to the size constrains of SMS and the use of phone keypad instead of the keyboard, a new language called SMS-Texting was invented based on the new abbreviations used to hide a message. Thus, this approach hides data by mainly using a set of abbreviations in SMS-Texting. These approaches are a textual steganography approach that can be categorized as a linguistic approach.
- Nostega-based: Recently, in the new paradigm in steganography research, namely Noiseless Steganography Paradigm (Nostega) [36,37], a message is hidden in the cover as data rather than noise. A number of methodologies have been developed based on the Nostega paradigm. One of these methodologies is the Summarization-Based Steganography Methodology (Sumstega) [38]. Sumstega exploits automatic summarization techniques to camouflage data in the auto-generated summary-cover (text-cover) that looks like an ordinary and legitimate summary. Another linguistic steganographic scheme that is also based on Nostega paradigm is the List-based Steganography Methodology (Listega) [39]. Listega manipulates itemized data to conceal messages in a form of textual list. Yet another is Notes-based Steganography Methodology (Notestega) [40] takes advantage of the recent advances in automatic notetaking techniques to generate a text-cover. Notestega embeds data in the natural variations among both human-notes and the outputs of automatic-notetaking techniques.It is worth noting that the presented Matlist methodology in this paper follows this new paradigm by exploiting NLG and template techniques along with Random Series values (RS) to camouflage data without generating any suspicious pattern.

1.2. Natural language generation and template

NLG is the process of employing a non-linguistic data input to produce an understandable text for both humans and machines. NLG employs knowledge base, artificial intelligence, computational linguistics, and other related techniques to achieve its goal [6,41]. Contemporary NLG techniques employ the knowledge of a domain-specific subject (DSS) [5] and its linguistics to generate texts in a form of reports, assistance messages, documents, and other desirable text. Note that contemporary NLG and template systems generate mature linguistic text [5,41]. Yet, the field of NLG systems has enjoyed significant progress in recent years and is still promising more in the future [5].

Some examples of NLG systems are WeatherReporter [5,42], FoG [5,42], and StockReporter [41,43]. WeatherReporter and FoG generate a textual weather description. The data input to these schemes is a numerical random series [5,41] and the DSS is the weather. This numerical random series represents the numerical weather data and the generated text by these systems describes the changes in weather. However, FoG is more advanced than WeatherReporter and it can generate a textual weather description in two different languages, English and French. Another example of a NLG system is the StockReporter, which was formerly known as the Ana scheme. The data input to the StockReporter scheme is a numerical random series and the DSS is the stock market prices. The numerical random series represents the values of key stocks and the generated text describes the fluctuations in stock market prices.

The template techniques were formerly known as mail-merge technology [5]. Mail-merge techniques have been employed in software packages such as Microsoft Word and others. The core idea of mail-merge is as simple as ‘fill in the blank’ by employing a predetermined template. Generic mail-merge can generate various text based on its input. Theoretically, NLG and mail-merge are equivalent in terms of functionality. To emphasize, any task that can be done by NLG systems can also be achieved by mail-merge systems and *vice versa*. It is argued that mail-merge techniques are NLG techniques [5]. However, from a complexity point of view, the NLG systems are a step ahead of mail-merge. Thus, in this paper NLG system is also referred to template system.

1.3. Matlist *versus* previous work

The text-cover of contemporary linguistic steganography approaches may contain numerous flaws such as incorrect syntax, lexicon, rhetoric, and grammar. In addition, the content of text-cover may be meaningless and semantically incoherent. These unusual patterns can easily raise suspicion in covert communications, which obviously defeats the steganographical goal. For example, in synonyms-based approach suspicion can be easily raised because some synonyms are not semantically compatible. Linguistically, this is due to the fact that there is not a large number of synonyms that can be generally used in various pieces of text. A synonym may be perfect in one piece of text but can be fully erroneous in another piece of text because of its different context. Even if the text-cover of synonym-based approach may look legitimate from a linguistics point of view, given the adequate accuracy of the chosen synonyms, reusing the same piece of text to hide a message is a steganographical concern. If an adversary intercepts the communications and oversees the same piece of text that has the same meaning over and over again with just different group of synonyms between communicating parties, he will question such use. The solution of to this problem is to avoid the reuse of same piece of text. Yet, the source of the original text that is used to hide data has to be kept secret.

Matlist avoids these issues by taking advantage of NLG and template techniques to generate a text-cover that naturally has a different legitimate meaning for concealing different messages while it remains semantically coherent and rhetorically sound. In addition, Matlist neither depends on the secrecy of a particular source of text, as a steganographic cover, nor its NLG system. Obviously, what is not made public is only the encoding system, including a cryptosystem and other related security procedures if used. Matlist does not depend on synonym substitutions for concealing data. Instead, Matlist employs NLG and template techniques to generate a text-cover, in which a message is embedded in a form of RS values (e.g., random series of binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, etc.), function of RS, related semantics of RS, a combination of these, etc.

Unlike synonyms-based steganography, linguistic flaws in noise-based approach are not a concern unless they appear excessively. For instance, all machine translations (MT) produce a translation that usually contains numerous errors (noise). For an adversary to suspect a covert communications, he has to detect unusual frequency of flaws, or odd patterns other than the use of MT. However, Grothoff *et al*. states that one of the concerns is that the continual improvement of machine translation may narrow the margin of hiding data [31–33]. On the contrary, an improvement in NLG is in fact beneficial to Matlist as demonstrated later in Sections 2, 3, and 4.

Similar to the translation-based approach, both the confusing approach [34] and the SMS-based approach [35] hide data in the noise (errors) and as long as the noise looks ordinary, they can fool an adversary. Unlike the translation-based approach, these approaches have no concern about the margin of hiding data, which is the amount of noise (errors) that occur by human in a noisy text e.g. emails, forums, blogs, SMS, etc., because it is not expected that the natural errors to be decreased. Conversely, Matlist neither employs errors nor uses noisy text to conceal data. Instead, it generates flawless text-cover, as demonstrated later in Sections 2, 3, and 4.

The remainder of this paper is organized as follows Section 2 introduces the Matlist methodology, Section 3 demonstrates the implementation of Matlist, Section 4 demonstrates the steganalysis validation of Matlist and experimental results, and Section 5 concludes the paper and highlights directions for future research.

2. MATLIST METHODOLOGY

Bob and Alice are on a spy mission. Bob is a medical practitioner and Alice is a market analyst consultant. Before they went on their mission, which requires them to reside in two different countries, they plot a strategic plan and set the rules for communicating covertly using their professions as a steganographic umbrella. They basically agree on concealing messages through the numerical data and their semantic that is often used in their profession. They make sure that the text-cover every time is generated has different meaning and it remains legitimate to avoid suspicion of using steganographical tool. To make this work, they establish a business relationship as follows. Bob is Alice’s medical doctor and Alice is Bob’s market analyst consultant. When Bob wants to send a covert message to Alice, Bob either posts medical related documents online for authorized patients to access or he can send medical related documents *via* email to the intended patients. These medical related documents conceal messages. Covert messages transmitted in this manner will not look suspicious because of Bob’s profession. Furthermore, Alice is not the sole recipient of Bob’s messages; other non-spy patients also receive their medical documents further warding off suspicion.

When Alice decides to send Bob a message, she does it in the same manner as Bob, except she uses her profession to do so. She posts market analysis reports that Bob or anyone

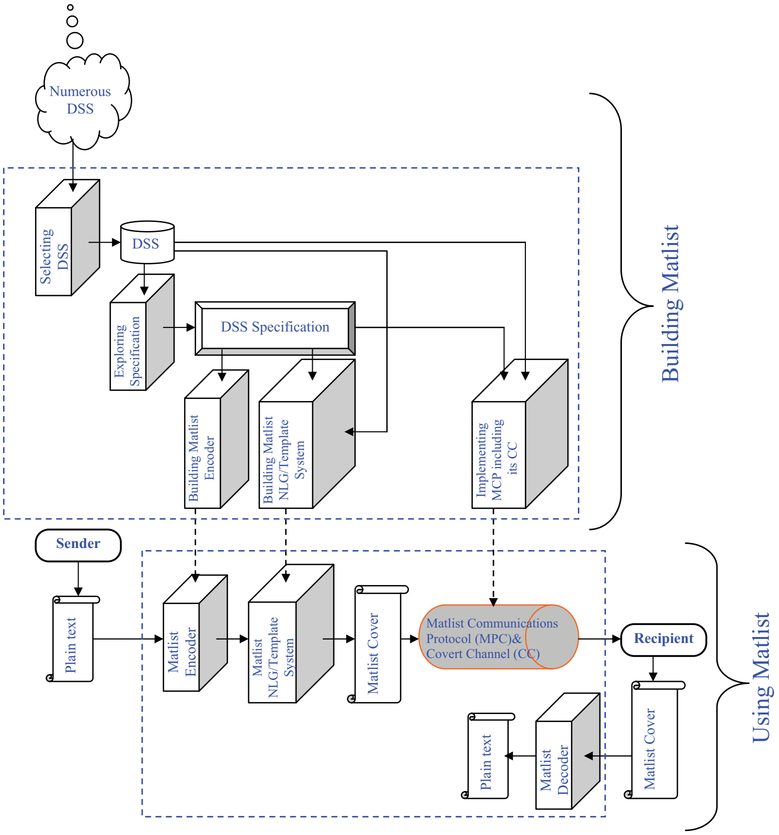

Figure 1. The architecture and the use of Matlist. It shows the interaction of various Matlist Modules to build Matlist. Then, it shows the use of Matlist scheme by the communicating parties.

else can access or she sends market analysis related documents *via* email to a set of clients that includes Bob. These market analysis reports conceal a hidden message. However, only Bob will be able to unravel the hidden message because he knows the rules of the game. Alice's communications looks legitimate and nothing is suspicious because she is a market analyst and she has a business relationship with both Bob and other non-spy clients. Alice or Bob can use real data from their professions and their established business relationship to make their covert communications legitimate. If real data is not used, then untraceable data can be fabricated to avoid comparison attack if an adversary attempts to trace data and compare it to its original. In addition, Bob and Alice are using their professions as linguistic domain-specific subject (DSS) for concealing a message.

The above scenario demonstrates how Matlist methodology can be used. Matlist methodology is demonstrated in the remainder of this section.

2.1. Matlist architecture

Matlist achieves legitimacy by basing the camouflage of both a message and its transmittal on a particular DSS. As stated earlier, in the above example of Bob and Alice, using the same DSS of the intended users gives legitimacy for camouflaging both the message and its transmittal. The following is an overview of the Matlist architecture, which consisted of five modules as shown in Figure 1:

(1) *DSS Determination* (Module 1): Determines a DSS, such as financial, medical, mathematical, scientific, economical, etc., that is appropriate for achieving the steganographical goal. The major factor is the use of random series values in the DSS. Examples include the use of binary, hexadecimal, octal, alphabetic, alphanumeric in computer science subjects as Discrete math, digital circuit, data structure, etc., and decimal values in financial documents.

(2) *DSS Specifications* (Module 2): Explores the properties and criteria of the DSS, selected by Module 1, and its RS to define DSS Specifications, which includes but not limited to, the appropriate linguistics for concealing data. Next, Modules 3 and 4 will construct Matlist Encoder and NLG system based on the defined DSS Specifications.

(3) *Building Matlist Encoder* (Module 3): Implements an encoder by employing the DSS Specifications, from Module 2, to encode messages. For example, based

on the properties of the RS, the encoder may represent the message as numerical values of RS (e.g., 43, 93, 109, 83, 4, etc.), a function of RS values, the linguistics used (e.g. increased, decrease, subset, not subset, etc.), or combination of these.

(4) *Building Matlist NLG* (Module 4): Implements an NLG or Template system by employing the process of building NLG system [5] (outlined in Section 2.5) along with the outputs of all previous modules. The constructed NLG or template system must be capable of accepting the encoded message, by Matlist Encoder, as an input to generate a text-cover.

(5) *Implementing Matlist Communications Protocol* (Module 5): As mentioned earlier, Matlist averts the suspicion during the transmittal of a hidden message by basing the camouflage of both a message and its transmittal on the same DSS. This module defines how the sender would deliver the text-cover covertly to the recipient.

The following subsections explain these modules in detail.

2.2. DSS determination (module 1)

The communicating parties must first agree on a DSS that they will use to conceal a message. For example, legitimate users may employ a financial, medical, mathematical, scientific, or economical report as the DSS. To enable the usage of the Matlist methodology, the selected DSS must involve a random sequence of recognizable tokens (e.g. binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, or any other form). To illustrate, changes in prices can be one form of a random series. Other selection criteria include the suitability of the chosen DSS for concealing a message without raising suspicion. Moreover, Matlist narrows the scope of the DSS in order to limit the linguistics used in a text-cover. For instance, Matlist favors the stock market as a DSS over the general economy. Such limitation will ease the text generation process and enhance its maturity [5].

The following elaborates on key criteria for selecting the DSS.

Random Series Based: Randomness in this context means that the members of the series do not exhibit patterns like the time series (e.g. 2, 4, 6, 8, etc.) where the increase and decrease are predicted. A DSS based on a random-series allows Matlist to conceal a message in text without violating any pattern. For example, if the message is encoded using numbers, it should be possible to blend these numbers in the cover text without exhibiting inconsistency, e.g. decrease in a value when an increase is expected. As will be demonstrated in Section 3, a DSS that is based on a random series has plenty of room for concealing messages. Nonetheless, such process requires careful manipulation. For instance, while prices in the stock market, foreign exchange rates, or temperatures are considered random, they are still somewhat controlled. In other words, there are limitations placed on a random series by its DSS and thus it will be imperative to pick a DSS that suits a particular steganographic encoder scheme. One would argue that the selection of the DSS and the encoder scheme are inter-related and thus it is hard to decide which one should be established first. However, determining the DSS is influenced by other factors and criteria and would need to be selected first. In addition, the specifications of the DSS can be exploited in order to ensure the feasibility of the camouflaging process as elaborated later.

Appropriateness of DSS: The chosen DSS has to fit the communicating parties and provide some ground for justifying the communications. It is also recommended that the chosen DSS suits the desired frequency of communications. With some domain-specific subjects, it may be possible to send messages every hour or so. For example, it is customary for a stockbroker to receive a market update every half-hour. On the other hand, some domain-specific subjects may not justify more than one message per month, per season or even per year. For example, someone would not very often receive an e-mail message from a utility company about the rate of energy consumption or payment history. It is also worth noting that multiple domain-specific subjects may also be employed to enable periodic and sporadic communications. In this case, the sender may need to have multiple scenarios in order to avoid suspicion about the sender–receiver association.

2.3. DSS specifications (module 2)

This module studies the properties and criteria of the selected DSS in order to generate its specifications. These specifications will be a base of constructing Matlist Encoder and NLG system to conceal messages. The specifications include two main aspects: the linguistics and the random series.

Linguistics Properties and Criteria: In order to generate an appropriate text-cover, the text has to be derived from the DSS Specifications. In other words, the linguistics of the text-cover has to be compatible with its DSS. The linguistics properties and criteria include, but not limited to, the following:

- The factors and inference of speech, text, or report. In other words, reasons that motivate a topic or event to be reported.
- The linguistic structure of a DSS. For example, the linguistic structure of a DSS such as the stock market is mainly a description of the fluctuation in stock market prices.
- The vocabulary, phrases, and technical terms that are popularly used in the chosen subject.
- The style of the presentation. This usually depends on the target reader or audience, e.g., general public, academics, professionals, students, etc.
- The text structure, e.g., documentary, report, e-mail, etc.

Properties and Criteria of Random Series: The properties and criteria of random series are critical for concealing a message because they directly affect the way a message is encoded and how a linguistic-cover is generated. These specifications of RS may include, but not limited to, the following:

- Type of the RS members, e.g., binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, etc.
- How are the RS used? For example, numbers are used in the medical field such as measuring blood pressure to represent measurements.
- What causes the differences among RS values? For example, the numbers reported about the stock market may represent the fluctuations in stock market prices.
- Constraints imposed on the RS values, e.g. numbers, may have to be greater than a particular value. For instance, the numbers that usually appear in a medical report for blood pressure will have to be within the normal range of living human beings and thus the message encoding scheme will have to cope with such a constraint.
- Is there any relationship among a subset of the members of an RS? For example, one member may be the average of few other members or a set is a subset of other and so on.

However, the specifications of RS and its related linguistics are fully integrated together and can be used to generate a text-cover, as elaborated below from a linguistics point of view.

Attributes: Investigating and studying each member of the RS as if it is alone and not in a series. After investigating each member of the RS and its nature, the result of the study will be defined as the criteria for the next stage of generating the Matlist Cover (linguistic-cover). For example, a member of an RS can be an integer, a real, an even, an odd, a prime, a float, or any other form other than numbers. This criterion can ease the task of forming the Matlist Code which will be used for generating the Matlist Cover (linguistic-cover) as will be demonstrated in the implementation section. For example, in a DSS such as the stock market, the price of a particular stock dropped or rose to a particular value that is an integer type not a real number or *vice versa*. This value as an integer or a real type can drive a sentence such as 'rose up to twenty dollars even' or 'dropped by twenty one dollars and five cent'.

One-To-One Relationship: In a one-to-one relationship, each member of an RS (set) has two neighbors excluding the first and last members of the RS. Investigation based on both the DSS and general relationships is conducted by studying all aspects of the RS and its relationships in the series (set). For example, the RS are studied in conjunction with its neighbors. In more detail, the first member (N_1) of the RS is studied in conjunction with the immediate member succeeding it (N_2) and the last member (N_k) of the RS is studied in conjunction with the immediate member preceding it (N_{k-1}). The study of the first and last member of the RS, it is only a one-to-one relationship. A one-to-one relationship can play a role for generating the Matlist Cover (the mature linguistic-cover) as will be demonstrated in the implementation section. For instance, in a DSS such as the stock market, the price of a particular stock dropped or rose to a particular value on the first day or the last day of a month. This event can drive a sentence such as 'the stock rose twenty monetary units' or 'the stock dropped by twenty one monetary units'.

One-To-Two Relationship: In a one-to-two relationship, all members of the RS excluding the first and last members of the RS are studied with the members of the RS succeeding and preceding it. This type of relationship is one-to-two. A one-to-two relationship can play a role for generating the Matlist Cover, as will be demonstrated in the implementation section. For example, in a DSS such as the stock market, the price of a particular stock dropped or rose to a particular value during a month. This event can drive a sentence such as 'the stock rose 20 monetary units after dropping by 21 monetary units last week'.

One-To-Many Relationship (Classes): In a one-to-many relationship, every aspect and the nature of each member of the RS (set) and its relationship to the entire series (set) or subset are investigated and studied based on a DSS and general relationships. Defining the results of this investigation can play a role for generating the Matlist Cover (the mature linguistic-cover) as will be demonstrated in the implementation section. For instance, in a DSS such as the stock market, the price of a particular stock dropped or rose to a particular value during a month. This event can drive a sentence such as 'the price rose up to its highest value during the month'.

Note that the presented properties and criteria in this section are just an example and other optimized specifications and criteria can be integrated with Matlist.

2.4. Building Matlist encoder (module 3)

Literately, coding is a very well researched technical area and there are numerous published techniques that can be employed to generate steganographic code [9,10,44,45]. Therefore, this subsection only focuses on key issues that affect the implementation of Matlist Encoder. Matlist Encoder generates a steganographic code (Matlist Code) in the form of a random series (RS), linguistics of RS, or a combination these based on the specifications of the chosen DSS. Then, Matlist Code (encoded message) will serve as an input to Matlist NLG system to generate the text-cover.

Mathematically, defining relationships in a random series is relatively more difficult than defining it in a time series. This is not the case in Matlist. For example, prices in the stock market, foreign exchange rates, temperatures, etc., are not controlled but random. Furthermore, there are limitations placed on the random series by its DSS. However, generating a steganographic code that preserves similar specifications of the DSS is feasible. If a message is in a form of RS then the relationships to itself (in terms of its attributes), its neighbor(s), a subset of the series, or the entire set (series). These relationships can share in generating artificial properties that looks legitimate. The process in this module is similar to the process of Module 2 (DSS Specifications).

Examples: As mentioned earlier, properties driven from relationships such as one-to-one, one-to-two, and one-to-many can be employed by Matlist Encoder to generate a steganographic code. For instance, a set of values in the form of a random series as similar as possible to the random series of the chosen DSS can form Matlist Code (an encoded message). If Matlist Code is fully represented by the random series, then it may take a form such as an integer, a real, an even, an odd, a prime, a float, or any other form than numbers. The fluctuations in random series values can be described using the linguistics, properties, of a DSS through Matlist NLG system. Matlist Code can be represented in the Matlist Cover using other forms than a random series, such as the linguistics that is related to the RS. To emphasize, in the stock market, the price of a particular stock dropped or rose to a particular value on the first day or the last day of a month. This can be described as one-to-one or one-to-two relationships in a random series. The specifications of such DSS are the base for describing the fluctuation in stock market prices. For instance, encoding words like events of dropped, rose, etc. can conceal a message. Matlist Code will be demonstrated in Section 3. Note that Matlist Decoder will simply be the inverse mode of the Matlist Encoder.

2.5. Building Matlist NLG or template system (module 4)

In this module, an NLG system or a template will be implemented to conceal messages. However, employing or modifying an existing NLG system or template is feasible as long as the generated text-cover will not raise suspicion. Matlist NLG/Template employs the encoded message (Matlist Code), generated by Matlist Encoder, to generate a text-cover. Obviously, the NLG systems or templates must be examined linguistically, steganographically, and technically by both humans and computers before using them. This gives more advantages and robustness to Matlist for passing both human and machine examinations because it has already passed these examinations by legitimate examiners. Furthermore, Matlist can employ either unaltered authenticated data (it is not authenticated text) or fabricated untraceable data to generate Matlist Cover avoiding the comparison attacks. It is worth noting that from the Matlist point of view, both NLG systems and templates are the same, as stated by Reiter and Dale [5], since both are capable of generating the Matlist Cover, as will be demonstrated in Section 3. Thus, in this paper NLG system is also referred to template system. In general, the implementation of templates based on a NLG is relatively easy and inexpensive. NLG system or template can be implemented in any language. However, in this paper, the implementation is only in English. Building Matlist NLG/Template system is mainly based on: the DSS Specifications generated by Module 2, Matlist Encoder and its Code constructed by Module 3, and the process of building NLG systems.

The process of building NLG systems, including templates, may be summarized as Reiter and Dale states in seven procedures [5] as follows:

(1) *Content Determination:* Determines the required information to be presented in the generated text. In a DSS such as weather reporting, the needed information may include temperatures, rainfall, rainy days, rain quantity, mist, fog, etc.
(2) *Document Structuring*: Determines the required informational classes, such as classification, grouping, ordering, of content and relates each class to its rhetorical terms. This informational classification can be in a form of informational tree (categorization) as shown in Figure 2.

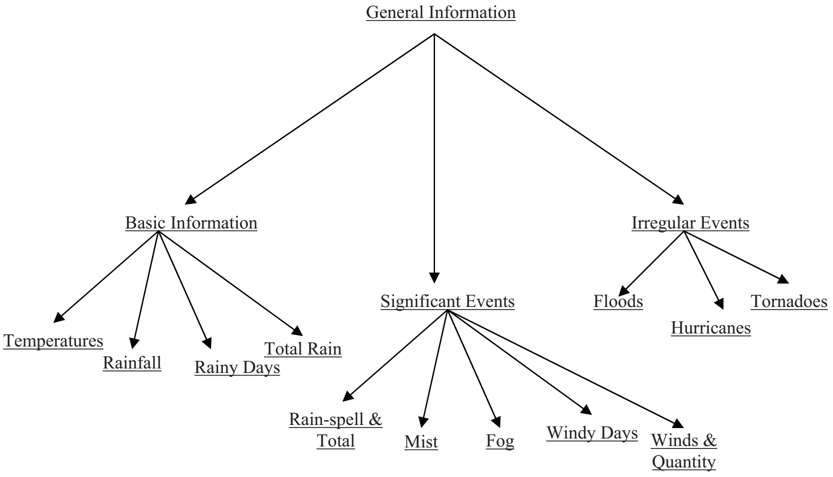

Figure 2. A possible structure of an informational tree of a weather report.

(3) *Lexicalization*: Determines the required linguistics, such as specific words and syntaxes, to be used in the output.
(4) *Referring Expression Generation*: Involves the determination of the required expressions that correlate entities. The relationship among these entities e.g. fluctuation of market prices, temperatures, etc., can be used to generate informative text.
(5) *Aggregation*: Determines the details of mapping the informational tree, from document structuring procedure, into linguistics structures such as sentences, paragraphs, etc.
(6) *Linguistics Realization*: Transforms the abstract representations of a sentence level that is collected from previous procedure(s) (the output of the previous procedure(s) is unordered text) into the required readable text.
(7) *Structure Realization*: Transforms the linguistics structures, e.g. paragraphs, sections, etc., into the required encoding sequence that generate the actual text. This step is analogous to converting a pseudo code, algorithm, or flowchart, into a particular programming language.

The above was just a brief overview of the process of building NLG and template systems. Since the focus of this paper is the linguistic steganography, for more details refer to Reference [5].

Once the Matlist scheme is implemented, camouflaging a message will be done in two steps. First, generating the required Matlist Code. Second, Matlist Code will serve as input to the Matlist NLG system or the Matlist Template to generate the Matlist Cover, which will be demonstrated in the implementation section.

2.6. Implementing Matlist communications protocol (module 5)

Covert communications is done through two steps, concealing a message, then transmitting the hidden message. Contemporary steganography approaches are focused on how to hide a message and not on how to hide the transmittal of a hidden message. Concealing the transmittal of a hidden message is as important as concealing a message. Consider the following scenario, a sender when communicating covertly always uses the same steganographic technique and the same steganographic cover type (e.g. translation-based, image-based, or audio-based). Furthermore, the sender always uses email to deliver a hidden message. Covert communications using the same steganographic technique, cover type, and email transmission all the time, will raise suspicion. An adversary overseeing this type of communications will be flagged and suspicion will be raised. Suspicion is raised because the adversary will wonder why the emails always contain one of the following: a translated document, an image, or an audio file. It is unusual for someone to send such content by email all the time. If the sender has no legitimate reason for sending an email containing one of the mentioned items, suspicion can be raised even if the content does not look suspicious, and nothing is detected. Suspicion is raised because of the way of delivering the hidden message not because of a vulnerable hiding technique used. However, it is more convincing when a sender has a website and posts a hidden message on it for a recipient to retrieve rather than sending the message through an email all the time. Another example, a sender in the financial industry has a legitimate reason for distributing a price analysis graph. Suspicion will not be raised if a message is concealed in the graph because of the legitimacy of distributing financial graphs. On the other hand, if the graph is a medical report, suspicion will be raised because the sender has no legitimate reason for sending a medical report. To emphasize, the way of delivering the hidden message can raise suspicion even if using a secure hiding technique.

Matlist averts the suspicion that may arise during the transmittal of a hidden message by basing the camouflage of both a message and its transmittal on a DSS. In addition, it should be imposed on the intended users to employ the appropriate: arrangements, techniques, policy, rules, and any other related requirements for achieving the steganographical goal.

Matlist Communications Protocol (MCP) works in the following way, as shown in Figure 1. A sender and a recipient communicate covertly using Matlist, and they agree to the following:

A. The particular specifications and configurations of Matlist scheme and its Decoder.
B. The particular specifications, configurations, policy, arrangements, and techniques of establishing a covert channel for the legitimate users to communicate covertly.

Once MCP is agreed upon, the intended users are ready to communicate covertly with each other using Matlist.

3. MATLIST IMPLEMENTATION

This section demonstrates possible implementation examples for five different DSS of consumer prices index (CPI), elementary math, selling books, chemistry, and discrete math. It discusses some important aspects of the implementation, and highlights possible directions for future implementation. Note that these are just few examples and it is expected to be implemented differently and achieving better results. The purpose of the presented implementation is to show the Matlist's capability of achieving the steganographical goal rather than making the adversary's task difficult to decode a message. Employing a hard encoding system or cryptosystem to protect a message is obviously recommended, feasible, and simple using any contemporary encoder or cryptosystem. Similarly, employing compression techniques to increase the bitrate can easily be accomplished by using the appropriate contemporary compression techniques. However, this is not the focus of this paper. Thus, for the simplicity neither cryptosystem nor compression technique is used in this paper. Given the availability of numerous encoding, encryption, and compression techniques in the literature [9,10,44,45] that can be employed, the discussion in the balance of this section will focus on the generation of Matlist Cover rather than the message encoding.

3.1. DSS of consumer prices index

A text example of the DSS of consumer prices index (CPI) is presented in Sample 1 in the Appendix Section. This sample is authenticated and was written by human and the source that was picked from is the U.S. Bureau of Labor Statistics [46]. Obviously, the sample of CPI is not written for concealing a message, but it was only written for CPI purposes. Sample 1, in the Appendix Section, is provided to show how the DSS of CPI looks. Evidently, collecting the numerical values in this sample will form a random series (RS) that is constrained by its domain, as indicated before in Section 2. However, it is still in the form of RS. Apparently, the text is just a linguistic description of the fluctuations among the values and shows that CPI is an appropriate DSS to be employed by Matlist methodology as demonstrated next.

3.1.1. First implementation example of CPI.

In this implementation example, Matlist predetermines the Matlist Encoder to encode a message based on the DSS Specifications of the CPI. Matlist Encoder employs a PSM Encoder [47,48] without encryption to assist in generating the Matlist Code. Note that PSM Encoder is not a part of the contribution and it is just used as an example. Matlist Code is generated as follows. Matlist employs a PSM Encoder to convert the plaintext message to a binary message then grouping its binary in lengths of seven digits. The grouping in lengths of seven digits will result in a value of 0 up to 127 in decimal. In other words, changing the value from 0000000 up to 1111111 in binary. Matlist Encoder employs an index that starts from 1 referring to 0 in integer (in binary 0000000) to 128 referring to 127 in integer (in binary 1111111). Matlist Encoder uses this index technique to avoid the occurrence of the value of zero in the encoded messages (the Matlist Code, which is in an RS form). This index plays a role as if Matlist Encoder adds 1 to each value after PSM encodes the message. To illustrate, the Matlist Encoder encodes the message as follows:

- The plaintext of the message is '*Use my secret key*'.
- The concatenated binary string of the ASCII representation of this message is: '0101010101110011011001010010011011010111100100100000011100110110010101100011011100100110010101110100001001101011011001010111 1001'
- Slicing this string (from the previous step) into 7 bits each will result in: 0101010 1011100 1101100 1010010 0000011 0110101 1110010 0100000 0111001 1011001 0101100 0110111 0010011 0010101 1101000 0100000 0110101 1011001 0101111 001
- Converting the individual slices (from the previous step) into decimals results in: '42 92 108 82 3 53 114 32 57 89 44 55 19 21 104 32 53 89 47 1 '.

Matlist then explores the feasible criteria that is compatible with the criteria of the selected DSS.

If Matlist will use numerical values in the Matlist Cover then the value of zero should not be used or should be rarely used. Therefore, in this example Matlist manipulates its code (Matlist Code) which is the RS by adding 1 to each value in order to avoid the occurrence of zero value, as stated before. As a result, the Matlist Code of the message in the form of integer values is as follows:

> '43 93 109 83 4 54 115 33 58 90 45 56 20 22 105 33 54 90 48 2 '.

In addition, the distribution of RS that is generated in this manner would not raise suspicion because using the index of RS values which is adding 1 to each of RS values will play a partial role of randomizing the message. For instance, a real value of '101', which is the ASCII of lowercase 'e', would be mapped to a different value by adding 1. Then, the new value will become '102', which is the ASCII of lowercase 'f'. Obviously, the letter 'e' and 'f' have totally different frequency. Also, a value of '90', which is uppercase of letter 'Z', would be mapped to a different value by adding 1. Hence, the new value will become '91', which is the ASCII of character '['. The letter 'Z' and character '[' have also totally different frequency. Similarly, the entire message is randomized. In reality, both the steganography and cryptography are complementing each other. Therefore, it is not only strongly recommended but it is essential to conceal a ciphertext instead of plaintext. Note that Matlist Code is referred also to as the 'encoded message' or the 'steganographic code'.

Attributes:

- A member of an RS can be an in a form of integer, a real, etc., and it is constrained by the DSS of CPI.*Inter-Member Relationship*:
- One-To-One Relationship between a member and its neighbor. The first number of the Matlist Code is ' 43' and the next number is ' 93'. The following properties can be tabulated:
  (a) The first number is less than the second number (its neighbor) or the second number is greater than the first number.
  (b) The difference between the two numbers is 50.
  (c) The noticeable trend is 'Rose'.
  - One-To-Two Relationship between a member and its neighbors of the Matlist Code. The

Table I. Example of the relationship of one-to-many (classes).

Highest (H)		Medium (M)		Lowest (L)	
Index	Value	Index	Value	Index	Value
7	115	9	58	20	2
3	109	12	56	5	4
15	105	17	54	13	20
2	93	6	54	14	22
10	90	19	48	8	33
18	909	11	45	16	33
4	83	1	43		

following properties can be tabulated for the second number and its neighbors (43 **93** 109):

(a) The 2nd number is greater than the 1st number by 50

(b) The 2nd number is less than the 3rd number by 16

(c) The noticeable trend is ‘ Rose’ and ‘Rose again’.

(d) These would be repeated for all values of the Matlist Code.

- One-To- Many Relationship (Classes) between each member and the entire or subset of the Matlist Code. For example, the range of the numbers in the RS can be sliced into three sub-ranges and the numbers then get grouped according to their values. In other words, the Matlist Code, such as ‘*43* **93** **109** **83** 4 54 **115** 33 58 **90** *45* *56* 20 22 **105** 33 *54* **90** *48* 2 ’, can be categorized in classes as the following procedure.

(a) The highest members of the above Matlist Code (the RS) are indexed as follows 7, 3, 15, 2, 10, 18, and 4 with their values as 115, 109, 105, 93, 90, 90, and 83 respectively (as marked in bold above and also shown in Table I).

(b) The medium members, marked in italics, are indexed as follows 9, 12, 17, 6, 19, 11, and 1 with their values as 58, 56, 54, 54, 48, 45, and 43 respectively, and also shown in Table I.

(c) The lowest members are underlined and indexed as follows 20, 5, 13, 14, 8, and 16 with their values as 2, 4, 20, 22, 33, and 33, respectively, and also shown in Table I.

The relationship of one-to- many can drive a sentence such as ‘The price rose up to its highest value during the month’.

Matlist Encoder encodes Matlist Code in the form of real numbers:

‘0.43 0.93 0.109 0.83 0.4 0.54 0.115 0.33 0.58 0.90 0.45 0.56 0.20 0.22 0.105 0.33 0.54 0.90 0.48 0.2 ’

Table II. An example message encoding scheme by using value range.

Range		Code-Words
125	127	Mounting
120	124	Leap
115	119	Augment
110	114	Escalated
105	109	Elevated
100	104	Jump
95	99	Boost
90	94	Increase
85	89	Climbed
80	84	Ascend
75	79	Inflate
70	74	Equate low/high
65	69	Cling low/high
60	64	Move up/down
55	59	Hold low/high
50	54	Budge up/down
45	49	Retain up/down
40	44	Lose
35	39	Dip
30	34	Depress
25	29	Flop
20	24	Fall
15	19	Decreased
10	14	Deflate
5	9	Decline
0	4	Sink

The above Matlist Code is embedded directly in the Matlist Template, as shown in Sample 2 of the Appendix Section.

3.1.2. Second implementation example of CPI.

The technique in this implementation example, as shown in Table II, is to define distinct code-words for the various ranges of values in the RS (Matlist Code). In order to allow decoding the message, a qualifier is used with the individual code-words to identify the particular number in the designated range. For example, the Matlist Code-word ‘*lost*’ is equal to a value range from 40 to 44. When using the text ‘*lost 0.4 per cent*’ in the linguistic-cover to determine the exact value, simply look-up the index from 40 to 44 so that the index value will be from 1 to 5. The index value 4 is equal to 43 and so on. Note that 0.4 refers to the index 4. Table III shows the code-words of the Matlist Code for the message ‘*Use my secret key*’. Unlike synonym-based approach, the code-words in this example are not synonyms. To emphasize, code-word ‘*lost*’ in a particular position in the Matlist Template may be substituted by another code-word ‘*increase*’ which is the antonym. As a result, the entire Matlist Cover will retain different legitimate meaning while it is semantically coherent, every time it is used for concealing different messages. The code-words are used naturally

Table III. Details of the encoding of the message 'Use my secret key' by using the value-range scheme of Table II. The qualifier field indicates the order of the coded number in the corresponding range so that the receiver can decode the message.

Initial Matlist Code	Corresponding Code-Words	Qualifier
43	Lose	3
93	Increase	3
109	Elevated	4
83	Ascend	3
4	Sink	4
54	Budge	4
115	Escalated	5
33	Depress	3
58	Hold	3
90	Climbed	5
45	Lose	5
56	Hold	1
20	Decreased	5
22	Fell	2
105	Jump	5
33	Depress	3
54	Budge	4
90	Climbed	5
48	Retain	3
2	Sank	2

because these code-words are a subset of the linguistics of the DSS used which are defined by specifications of the DSS used. Furthermore, the use of NLG or a template provides and maintains a correct text generation. Therefore, code-words in Matlist methodology do not cause any noise in its text-cover, as shown in Sample 3 of the Appendix Section.

3.1.3. Text substitution.

The previous examples are given in the form of a generic report. Alternatively, words or a combination of words can be substituted with other words or combinations of words. This technique is simple such as the format of a 'wizard form and fill in the blank' [5]. Unlike synonym-based, in Matlist text substitution procedure may not preserve the same meaning of text and can give other meaning. For example, '*increased*' can be substituted by antonym such as *decreased*; '*As recently*' can be substituted by a specific date or month such as *February*; '*our department*' can be substituted with a corporate name such as the *Department of Labor*; **'***the first product*' can be substituted with an actual item like *fuel*; '*lost 0.2 per cent*' can be substituted with other numerical values; '*in the second period*' can be substituted with a date, month or any other time including *in the second quarter, in the second month or in February;* and so on. These examples of text substitution represent only a few samples of what can be substituted. Matlist does not cause any errors when it employs any text substitution techniques. Obviously, Matlist methodology is based on a natural language generation or template techniques where these techniques ensure the production of legitimate text. Matlist Text Substitution (MTS) is a feature in Matlist that gives Matlist the advantage of being flexible in generating the Matlist Cover and it can be used to increase the bitrate. This technique is elaborated next section.

3.1.4. Third implementation example of CPI.

In this implementation example, the technique used directly maps code-words to an exact value. First of all, the entire data that are represented in the Matlist Cover, in Sample 4 of the Appendix Section, are true and authenticated data (information not the text). The source of the authenticated data (not text), used in this example, is the U.S. Bureau of Labor Statistics [49]. These data are collected and embedded along with Matlist Code to generate Matlist Cover. There is a tremendous amount of authenticated data available, especially by employing the World Wide Web (Internet) which can be employed to generate Matlist Cover. Note that the use of the authenticated data is totally different from using an existing text. The use of an existing text is vulnerable to comparison attack. Therefore, Matlist uses the authenticated data and it does not use existing text. In other words, the use of authenticated data is referred to only the use of the informational facts and not their existing text. Furthermore, this technique does not embed a message in a form of numerical values of Matlist Code. Instead, it embeds a message in the linguistics of the generated text. This technique is similar to the previous one (Section 3.1.2 and 3.1.3) except that the code-words are assigned to exact values. The idea is simply to define an implicit mapping of code-words that are used naturally in the DSS to represent the fluctuations in its RS. As stated earlier, it is unlike synonym-based approach because the code-words are not synonyms. For instance, a code-word '*lost*' in a particular position in the Matlist Template may be substituted by another code-word '*increase*' which is not only different meaning but it is also antonym. Yet, Matlist Cover remains semantically coherent and rhetorically sound. The receiver will collect the words in the text-cover (Matlist Cover), convert them to the corresponding numbers, and decode the numbers to form the hidden message. Table IV shows the code-words that can be used to conceal a message, as shown in Sample 4 of the Appendix Section. The Matlist Encoder encodes the message '*Use my secret key*' as follows. Matlist Encoder employs a PSM Encoder to convert the plaintext message to a binary string and then slice the string into groups of five digits. Grouping in lengths of five digits yields numbers in the range of 0 to 31 (in binary from 00000 to 11111). Table V shows the code-words that need to be used and the order of their appearance in the Matlist Cover. Since no values of the RS will conceal data, there is no need for using an index such as adding 1 to each value of the PSM Code. To emphasize, the value of zero was avoided in all previous techniques by using the index. This technique can produce text-cover like the one that is shown in Sample 4 of the Appendix Section.

Table IV. Directly mapped code-word to exact values.

Value	Code Word
31	Mounting
30	Leap
29	Augment
28	Escalated
27	Elevated
26	Jump
25	Boost
24	Gain
23	Increase
22	Climbed
21	Ascend
20	Inflate
19	Check
18	Equate Low/High
17	Cling Low/High
16	Move Up/Down
15	Hold Low/High
14	Budge Up/Down
13	Retain Up/Down
12	Devalue
11	Reduce
10	Lose
9	Dip
8	Depress
7	Flop
6	Fall
5	Shrink
4	Decreased
3	Deflate
2	Decline
1	Droop
0	Sink

3.2. Other DSS

3.2.1. DSS of elementary math.

A text example of the DSS of Elementary Math is presented in Sample 5 of the Appendix Section. This sample, which picked from [50,51], is authenticated and was written by human only for teaching purposes and obviously it is not written for concealing data. It is provided to show how the DSS of Elementary Math looks. Apparently, the text is just a normal question in Elementary Math that contains a real RS and its value almost has no constrain. Thus, DSS of Elementary Math is an appropriate DSS to be employed by Matlist methodology as demonstrated implementation example. Sample 5 of the Appendix Section also shows that a simple NLG or template system can be employed by Matlist to generate text-cover. This fact is confirmed by Sample 6 of the Appendix Section. Sample 6 of the Appendix Section is generated to conceal the message '*Use my secret key*' and its Matlist Code is generated in the same way as shown in Section 3.1.1. The message is directly embedded in the form of RS, as shown in Sample 6 of the Appendix Section. Obviously, a bitrate for such domain will be very high.

Table V. Message encoding using a directly mapped code words to exact value (of the sliced binary representation of message) from Table IV.

Order	PSM Sliced Binary String	Group value in Integer	Matlist Code Word
1	01010	10	Lose
2	10101	21	Ascend
3	11001	25	Boost
4	10110	22	Climbed
5	01010	10	Lose
6	01000	8	Depress
7	00011	3	Deflate
8	01101	13	Retain Up/Down
9	01111	15	Hold Low/High
10	00100	4	Decreased
11	10000	16	Move Up/Down
12	00111	7	Flop
13	00110	6	Fall
14	11001	25	Boost
15	01011	11	Reduce
16	00011	3	Deflate
17	01110	14	Budge Up/Down
18	01001	9	Dip
19	10010	18	Equate Low/High
20	10111	23	Increase
21	01000	8	Depress
22	01000	8	Depress
23	00011	3	Deflate
24	01011	11	Reduce
25	01100	12	Devalue
26	10101	21	Ascend
27	11100	28	Escalated
28	1	1	Droop

3.2.2. DSS of selling books.

Given the growing online businesses nowadays, there are so many people selling new and used books. Such DSS of Selling Books allows communicating parties to send and receive legitimate book prices averting suspicion in covert communications. To see a legitimate sample of book prices, which do not conceal messages, refer to any online books seller (e.g. amazon.com, ebay.com, yahoo stores, etc.). Apparently, any individual that is selling books online can post book prices of unrelated subjects. This is because such seller is also a buyer for both new and used books. The text in this DSS is just a linguistic description of book titles, author names, prices, etc., and either the fluctuations among book prices or their alphabetic string (e.g. different book titles, different author names, etc.) can be the steganographic carrier to conceal data. Matlist Code will be represented by different prices, as shown in Sample 7 of the Appendix Section. Yet, different book titles or author names can form alphabetic strings of RS (Matlist Code). Sample 7 of the Appendix Section conceals the message '*Use my secret key*' and its Matlist Code is generated in the same way as shown in Section 3.1.1. It employs true and authenticated data (not text) to form the text-cover. The data, not the text, were collected on 06 May 2008 from www.amazon.com using Internet search engines. If an adversary looks at the data of Matlist Cover, he will conclude that the book titles, author names, and prices are valid data. Thus, this technique justifies the use of Matlist.

3.2.3. DSS of chemistry.

A text example of the DSS of Chemistry is presented in Sample 8 in the Appendix Section. Sample 8 of the Appendix Section [52] is authenticated and was written by human (Department of Chemistry at Ohio State University) only for teaching purposes and obviously it was not written for concealing data [52]. It is provided to show how the DSS of Chemistry looks. Apparently, the text is a normal question in Chemistry that contains numerical values that can form an RS. Since the question is multiple choices, most likely only one is correct and others must be wrong. Thus, concealing data in the incorrect answers is easy and the text is still legitimate. Such topics in Chemistry are appropriate DSS to be employed by Matlist methodology as demonstrated next. Sample 8 of the Appendix Section also shows that a simple NLG or template can easily be employed by Matlist to generate text-cover, as confirmed by example in Sample 9 of the Appendix Section. Sample 9 of the Appendix Section conceals the message '*stop*' and its Matlist Code (the encoded message) is generated in the same way as shown in Section 3.1.1. The question in Sample 9 of the Appendix Section is similar to other questions in the DSS of Chemistry, for more authenticated samples, which do not conceal messages, refer to Reference [52].

3.2.4. DSS of discrete math.

The DSS of Discrete Math is a well-known topic in the filed of computer science. The use of binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, etc., in such DSS is an ordinary practice. For more detail about the DSS of Discrete Math, refer to Reference [53]. Sample 10 of the Appendix Section conceals the message '*stop*'. The Matlist Code (steganographic code) of a message is generated in a similar way of Section 3.1.1 and 3.1.3. Tables VI and VII detail the Matlist Code that is used in Sample 10 of the Appendix Section. Evidently, a bitrate for such a domain will be superior, without raising suspicion to achieve the steganographical goal. This is because of the excessive use of random values and strings of binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, etc. in such domain.

Table VI. Shows the Matlist Code using the alphabetic letters. Apparently, other symbols can be included such as numerical values. However, these are just for showing the feasibility of such domain.

Steganographic Values	Coded Letters	Non-Coded Letters
0000	A	Q
0001	B	R
0010	C	S
0011	D	T
0100	E	U
0101	F	V
0110	G	W
0111	H	X
1000	I	Y
1001	J	Z
1010	K	
1011	L	
1100	M	
1101	N	
1110	O	
1111	P	

Table VII. Shows the Matlist Code using some notations of Discrete Math. Apparently, other notations can be also included. Nonetheless, these are just for showing the feasibility of such domain. 'These notations' values are circulated. For example, the notation 'subset' is equal to '000' 1st time used, 2nd time used it will be equal '001', 3rd time used it will be equal '010', and so on. In other words, after the 1st time a notation is used it adds 1 every time it is used. Note, the code-words are not synonyms.

Steganographic Values	Symbols	Code Word
000	$\subseteq\supseteq$	Subsets
001	$\subset\supset$	Proper subset
010	$\not\subset$	Not subset
011	$\not\subset$	Not proper subset
100	$\cap$	Intersection of sets
101	$\cup$	Union of sets
110	$\in$	Element a member of a set
111	$\notin$	Element is not a member of a set

3.3. Matlist bitrate

The aim of this section is to show the bitrate of contemporary linguistic steganography approaches *versus* Matlist bitrate. Therefore, the linguistics experiment of Matlist investigated the bitrate of the contemporary approaches and the results are as follows:

(1) The statistical-based approach, namely mimic functions approach [9,10], observed an average of 0.90% bitrate based on Spam Mimic Scheme [54].

(2) The synonym-based approach is:
- The revised NICETEXT Scheme [15] may be achieves proximally bitrate of 0.29%.
- Winstein's Scheme [16,17,28], as clamed, achieves approximately 0.5% (roughly is about 6 bits per Penn Treebank sentence [16,17]).
- The scheme of Murphy *et al.* [28] may achieve 0.30% bitrate per sentence and not every sentence in the text-cover conceals data. In addition, the size of sentience will effect the bitrate because there is a short and long sentence. This

Table VIII. shows the bitrate of the presented Matlist Scheme up to date.

DSS	Bitrate of Matlist
CPI	0.58-1.02%
Elementary Math	19.09-21.51%
Books Seller	1.35-2.16%
Chemistry	2.424%
Discrete Math	18.4%

implies that the overall bitrate will be lower than 0.30% and it is based on the size of the text cover, how many sentences were used to conceal data, and the average number of words per sentience.
- The scheme of Nakagawa *et al*. [22] may achieve bitrate of 0.06%, 0.12% and for real application 0.034%.

(3) The noise-based approach is:
- The translation-based Scheme [31–33] roughly achieves bitrate of 0.33%.
- The confusing Scheme [34] approximately achieves bitrate of 0.35%.
- The SMS-based scheme (abbreviations-based) [35] can hide few bits (the paper did not indicate how many bits) in a file of several kilobytes (the paper did not indicate how many kilobytes). This means that it is extremely a low bitrate. Other techniques of SMS-based, according to Reference [35], are not linguistics and the hidden message is subject of distortion attack.

Matlist, based on experimental observation, achieves superior bitrate than all contemporary approaches, as shown in Table VIII. The bitrate of Matlist may differ from one DSS to another and from one implementation to another as observed. In regards of message size, generally, the size of messages is a concern of most if not all steganography approaches. However, in particular to the presented Matlist scheme, Matlist is capable of camouflaging long messages. For example, in regards to the presented implementation example, if a message does not fit within a single report of a CPI, elementary math quiz, book prices, chemistry question, or discrete math problem then Matlist will distribute it on multiple piece of text without decreasing the bitrate.

4. STEGANALYSIS VALIDATION

The aim of this section is to show the resilience of Matlist to possible attacks. Matlist is a public methodology; however, the word ‘public’ does not imply in this paper that an adversary has the same or entire Matlist scheme. It is assumed that an adversary is well knowledgeable about the methodology of Matlist, but he does not have the specifications of the actual implementations of all Matlist components used.

4.1. Traffic attack

Traffic attack is the procedure of investigating and cracking steganographic communications by investigating only the communications’ traffic without investigating a particular steganographic cover. If the steganographical users are communicating with each other in a visible manner by sending, receiving, accessing, or obtaining materials without a legitimate reason for doing so, then suspicion can be raised without any further investigation. For example, a medical doctor would not exchange weather analysis reports with one of his patients. Such communications can easily raise suspicion because a medical doctor should send medical documents not weather documents. Traffic attack can be applied to any contemporary steganographic techniques regardless of the steganographic cover type, e.g., image-cover, audio-cover, text-cover, etc., and can achieve successful results with relatively low costs. Further investigations can be applied once suspicion is raised during a traffic attack.

Matlist ensures that the communicating parties establish a secure covert channel for transmitting the hidden message covertly. In other words, Matlist naturally camouflages the delivery of a hidden message in a way that makes it appear legitimate and innocent. The scenario discussed in Section 2 demonstrates how the communications between Bob and Alice would not be unusual because their professions play the role of DSS that legitimizes the visible communications. As long as there is a legitimate reason for sending, receiving, accessing, or obtaining a particular material, suspicion can be averted. Hence, the Matlist steganographic communications will remain unseen to the adversary because, by establishing a covert channel, the delivery of a hidden message is also hidden to achieve unseen delivery for the unseen. In addition, investigating all traffics between communicating parties, like the examples above (in Section 3), are impossible given the astronomical volume of traffics to suspect, rendering Matlist a resilient methodology.

4.2. Contrast attack

One of the intuitive sources of noise that may alert an adversary is the presence of contradictions in the text such as finding, e.g. CPI in Section 3, the value of a product edging up while saying that it has decreased. It is worth noting that the traffic analysis, discussed earlier, can also be pursued as a base for launching contrasts attacks in case the data is not publicly accessible. In the later case, an adversary can contrast the use and reference to data values in the same cover or track them over some period of time in order to identify any inconsistency. Contradictions would surely raise suspicion about the existence of a hidden message. Countering such an attack is always a challenge. The Matlist scheme, as demonstrated through the examples in Section 3, is simply made contrast-aware in order to avert such attacks by generating text-cove that is free of contradictions.

4.3. Comparison attack

Noise detection, in the context of comparison attacks, reflects an alteration of authenticated data or text. The goal is to find any incorrect or altered data that may imply the presence of a hidden message. Domain-specific subjects (DSS), like the CPI, are inherently public domain information and an adversary can access older CPI reports. The same applies when authenticated data, such as weather forecast or historical temperature measures, are used. Avoiding such attacks is a challenge because it requires consistency with publicly accessible data. However, Matlist is very resilient against such attacks. The enormous amount of data (not text) available, both publicly and privately, provides sufficient sources of correct and consistent data for embedding all sorts of messages. The use of available data does not imply the use of an existing text but it means only the use informational facts. The 472K-size report from the U.S. Bureau of Labor Statistics [49], used in Section 3.1.4, is just a sample of available data (informational facts not text). As long as an attack is known, it is feasible to be avoided simply by constructing the steganographic scheme to be aware of contemporary attacks. For example, if the communicating parties are concerned about comparison attacks then Matlist should be made comparison-aware in order to avoid such an attack, as demonstrated in Section 3.

4.4. Linguistic attack

Linguistic examination is to distinguish the text that is under attack from normal human language. Distinguishing the text from normal human language can be done through the examination of meaning, syntax, lexicon, rhetoric, semantic, coherence, and any other issues that can help to detect or suspect the existence of a hidden message. These examinations are used to determine whether or not the text is under attack is abnormal. The generated text by the contemporary NLG and template systems is meaningful, syntactically correct, lexically valid, rhetorically sound, semantically coherent, grammatically correct, and legitimate [5]. Since Matlist is based on NLG and template techniques, the generated text (Matlist Cover) is most likely free of linguistic errors. Generally, the linguistic limitation that is imposed by employing the DSS makes it possible for any NLG and template systems to be linguistically error free [5,41]. Furthermore, if there is any engine error it should not be a concern for two reasons; first, it is feasible to resolve and fix any implementation problem; second, nothing is concealed in errors. Therefore, it is obvious that Matlist is capable of passing any linguistic attacks by both human and machine examinations.

4.5. Statistical signature

In this paper, the statistical signature (profile) of a text refers to the frequency of words and characters used. An adversary may use the statistical profile of normal text that contains no hidden message and compare it against a statistical profile of the suspected text to detect any differences. An alteration in the statistical signature of a normal text can be a possible way of detecting a noise that an adversary would watch for. Tracking statistical signatures may be an effective means for attack because it can be easily automated and combined with traffic analysis. However, Matlist is resiliently resistant to statistical attacks as will be demonstrated by the experimental results below and for more experimental information about statistical signature attacks refer to Reference [37].

4.5.1. Word frequency.

Human language in general and the English language in particular, have been statistically investigated [55,56] to discover its statistical properties. The most notable study on the frequency of words was done by George Kingsley Zipf [55,56]. Zipf investigated the statistical occurrences of words in the human language and *in particular the English language*. Based on the statistical experimental research, Zipf concluded his observation, which is known as Zipf's law [55,56]. Zipf's law states that the word frequency is inversely proportional to its rank in an overall words frequency table, which lists all words used in a text sorted in a descending order of their number of appearances. Mathematically, Zipf's law implies that $W_n \sim 1/n^a$, where W_n is the frequency of occurrence of the n^{th} ranked word and 'a' is a constant that is close to 1. Based on such a mathematical relationship, a logarithmic scale plot of the number of words' appearance and their rank will yield a straight line with a slope '-a' that is close to -1. The value of 'a' is found to depend on the sample size and mix. Zipf's law was originally observed on a huge bundle of textual collections containing numerous different DSS by different authors, different writing-styles, different writing-fingerprints, etc. Consequently, this huge bundle of textual collections is fairly blended which causes the occurrence of approaching or reaching Zipfian of -1.

The Matlist experiment applied Zipf's law directly on Matlist Cover considering the worse case scenario that an adversary knows Matlist methodology and knows if there is a hidden message where it is concealed. Unlike Zipf's experiment, the Matlist experiment applied Zipf's law on a short piece of text with a unique DSS. Based on the experimental observation, shown in Table IX, Matlist Cover (that contains a hidden message) holds a Zipfian slope with an average of -0.87016. On the other hand, the unaltered authenticated text of the same domain, without any hidden message, holds a Zipfian slope with an average of -0.75611, as shown in Table IX. Apparently, both Matlist Cover and the unaltered authenticated text of the same domain (that contains no hidden message) do not fully obey Zipf's law. However, Matlist Cover is closer to the Zipfian of -1 than its DSS (the unaltered authenticated text of the same domain that contains no hidden message). The difference between both slopes of Matlist Cover and its DSS (the authenticated text that contains no hidden message) is very

Table IX. The Zipfian distribution (logarithmic scale) of Matlist Cover with a hidden message *versus* the text of the chosen DSS without a hidden message. The equation is a linear curve fitting of the results. R^2 is the squared error.

	Matlist Cover			Text without hidden message		
Text #	Equation	R^2	Slope($-a$)	Equation	R^2	Slope($-a$)
1	$-0.8922x + 1.6735$	0.9141	−0.8922	$-0.8245x + 1.4915$	0.9329	−0.8245
2	$-0.8923x + 1.6595$	0.8952	−0.8923	$-0.8741x + 1.698$	0.9467	−0.8741
3	$-1.0243x + 1.7418$	0.9145	−1.0243	$-0.7412x + 1.266$	0.9251	−0.7412
4	$-1.0683x + 1.8115$	0.9145	−1.0683	$-0.8542x + 1.6855$	0.9512	−0.8542
5	$-1.1287x + 1.9761$	0.893	−1.1287	$-0.9557x + 1.8569$	0.9559	−0.9557
6	$-1.1287x + 1.9761$	0.893	−1.1287	$-0.737x + 1.4103$	0.9201	−0.737
7	$-1.107x + 2.0269$	0.9051	−1.107	$-0.737x + 1.4103$	0.9201	−0.737
8	$-0.8459x + 1.4629$	0.9165	−0.8459	$-0.758x + 1.2825$	0.9091	−0.758
9	$-0.8068x + 1.4024$	0.9107	−0.8068	$-0.7493x + 1.428$	0.9109	−0.7493
10	$-0.8022x + 1.3283$	0.892	−0.8022	$-0.6697x + 1.4098$	0.9173	−0.6697
11	$-0.7883x + 1.3009$	0.885	−0.7883	$-0.705x + 1.4186$	0.9257	−0.705
12	$-0.7521x + 1.1838$	0.8818	−0.7521	$-0.6559x + 1.2942$	0.8882	−0.6559
13	$-0.6779x + 1.0286$	0.8827	−0.6779	$-0.7171x + 1.1889$	0.9159	−0.7171
14	$-0.7613x + 1.2248$	0.9069	−0.7613	$-0.6052x + 0.9868$	0.8342	−0.6052
15	$-0.7607x + 1.1986$	0.8939	−0.7607	$-0.9121x + 1.5605$	0.9461	−0.9121
16	$-0.7804x + 1.2725$	0.8795	−0.7804	$-0.8504x + 1.3719$	0.9015	−0.8504
17	$-0.7881x + 1.2988$	0.8734	−0.7881	$-0.7116x + 1.3634$	0.8902	−0.7116
18	$-0.7745x + 1.2665$	0.8774	−0.7745	$-0.7093x + 1.363$	0.9035	−0.7093
19	$-0.8885x + 1.5789$	0.9288	−0.8885	$-0.7352x + 1.329$	0.9185	−0.7352
20	$-0.8003x + 1.3395$	0.8722	−0.8003	$-0.7085x + 1.3469$	0.9021	−0.7085
21	$-0.859x + 1.497$	0.9162	−0.859	$-0.6697x + 1.4098$	0.9173	−0.6697
22	$-0.8617x + 1.5046$	0.9271	−0.8617	$-0.6603x + 1.2676$	0.8973	−0.6603
23	$-0.8617x + 1.5046$	0.9271	−0.8617	$-0.671x + 1.3073$	0.9037	−0.671
24	$-1.4287x + 3.2792$	0.8331	−1.4287	$-1.4148x + 3.3485$	0.9348	−1.4148
Average			−0.89498			−0.75021

minimal and it is just −0.14477 because as shown the text that does not contain a hidden message it fluctuates by similar values. To emphasize, Matlist Cover is also very close to its DSS (the authenticated text that contains no hidden message). This makes Matlist Cover on the safe side of both a Zipfian of −1 and the Zipfian of its DSS (the authenticated text that contains no hidden message).

Moreover, when applying Zipf's law for a list of 16 Matlist Cover and 16 pieces of text from the same DSS of Matlist Cover (the authenticated data that contains no hidden message) [57] the Zipfian slope for Matlist Cover holds at −1.4287 and its DSS holds at −1.4148. Again, both slopes do not fully obey Zipf's law and are far from the Zipfian of −1. However, Matlist is still in the same region of its DSS (the authenticated data that contains no hidden message). The difference between both slopes of Matlist Cover and its DSS (the authenticated data that contains no hidden message) is very minimal and it is just −0.0139. To emphasize, Matlist Cover is also very close to its DSS (the authenticated data that contains no hidden message).

The Matlist's experiment of word frequency concludes the following. Since the Matlist methodology is based on a DSS, then when applying Zipf's law, Matlist Cover should be similar to a Zipfian of its DSS (the unaltered authenticated text of the same domain that contains no hidden message) and it is not required to fully obey Zipf's law (Zipfian of −1). To emphasize, if the Zipfian slope of the Matlist DSS (the unaltered authenticated text of the same domain that contains no hidden message) is equal to *N* value, then Matlist Cover should be either equal to or close to the *N* value. Generally, it is feasible to fool any attacks as long as the attack models are known, simply by constructing the steganographic scheme as attacks-aware. Furthermore, it is feasible to alter the natural language in a way that can fool Zipf's law if it is required. Simply, Matlist can be designed as Zipf-aware [31–33] since the statistical model is already known.

4.5.2. Letter frequency.

Generally, in any language some letters appear at higher frequencies than others [37,58–60]. For example, in the English language the letters 'E', 'T', and 'A' are the most-frequently-occurring letters and 'J', 'Q', and 'Z' appear the least. However, in some DSS this general observation does not hold. For example, the words 'judgment', 'jurisdiction', 'injured', 'injuries', 'judicial', 'jury', and 'subject' are used frequently in court related documents which gives the letter 'J' an uncommonly high frequency. Similarly, the letter 'Q' in the DSS of Queuing System (in a telecommunications field) boosts the frequency of the

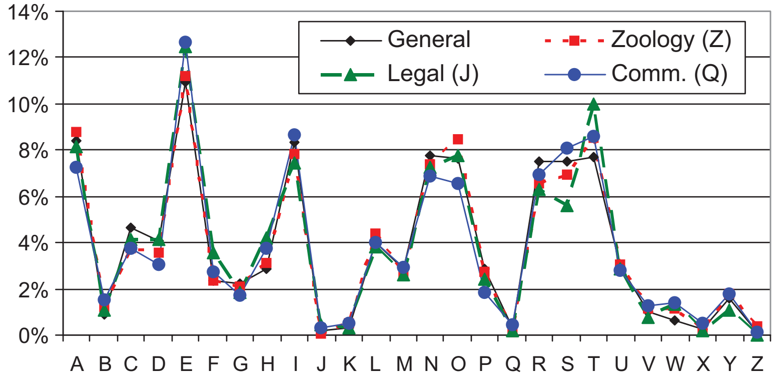

Figure 3. Distribution of letter usage in general and domain-specific literature.

letter ‘Q’, and the letter Z in some DSS such as Zoology have uncommonly high frequencies.

However, it was observed that the overall impact on the frequencies of the various letters is not that dominant since the words that increase the use of a certain letter also boost the appearance of others [37]. Figure 3 confirms this observation by comparing the plot of the Letter Frequency Distribution (LFD) in documents from four different DSS. The multiple DSS set is based on the 2005–2006 Graduate Catalog of the University of Florida Gainesville [61], which contains over 1.4 million characters. The other sets are based on text from Queuing System of telecommunications field [62], Zoology [63], and court documents [64]. The LFD of these four different sets of text are different but roughly obey the characteristics of the letter frequency-distribution-plot of each other, as shown in Figure 3. In other words, the peaks and valleys of each plot of the LFD closely match each other. These four different sets of text are authenticated text and not used for concealing a message.

Similarly, comparing the plot of both the average LFD of Matlist Cover (contains a hidden message) and the average of LFD of the unaltered authenticated text of the same domain (without a hidden message) are almost the same, as shown in Figure 4. In other words, the peaks and valleys of both LFD plots almost match each other.

4.6. Results of human examination experiment

In linguistic steganography, human examination is essential, for inspecting a particular text whether or not it contains a hidden message, because contemporary approaches violate the normality of human text when they conceal messages. Therefore, Matlist’s experiment is conducted by human examination. To avoid any conflict of interest, it is imperative that a qualified person, having no relationship with the author, administrates the experiment and selects the examiners. As such, the Program Director of Ph.D. (PDP) in language, literacy, and culture [65] at the University of Maryland, Baltimore County administrated the experiment of this paper. According to the PDP website, the PDP was also the vice president and the director of the International and Corporate Education, Center for Applied Linguistics, Washington, DC (1978–1992). Furthermore, the PDP received adequate information about linguistic steganography orally and by softcopy. The softcopy consists of both the linguistic steganography background and the experiment. The experiment contained the sample texts of Matlist Cover and authenticated text. The experiment was done on one of the most challenging DSS, which is CPI. The PDP decided to select a group of graduate students to be the examiners.

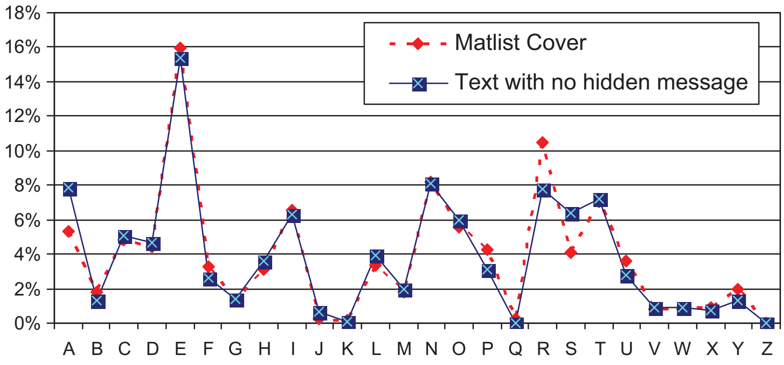

Figure 4. Distribution of average letter usage in a Matlist Cover and the version of the text that contains no hidden message.

The PDP had an open discussion about the experiment with the examiners to ensure that they got the required information for suspecting the right text. The experiment asked the seven examiners the following questions about each text:

- Do you suspect a hidden message?
- If yes, state which paragraph(s) is suspicious and why?
- The PDP, after consulting with the author, requested from the examiners to include any suggestions or comments.

After more than 2 weeks, the results were returned from the examiners and the following was concluded:

- No examiner suspected a hidden message.
- 3 examiners made comments about the authenticated text and the Matlist Cover.
- 3 examiners made comments only about the authenticated text
- One examiner has no comment.

The comments concluded that the CPI text was generally hard to read and the writing style of CPI reports is uncommon. It is obvious that anyone without experience with CPI text will be uncomfortable reading it. It is also a good sign that the three examiners who commented about Matlist Cover also made similar comments about the authenticated text. Moreover, the other three examiners commented only about the authenticated text and they did not comment on the Matlist Cover. Thus, the experiments confirmed that Matlist methodology is capable of fooling human examiners.

5. CONCLUSIONS

Matlist achieves the steganographical goal by basing the text-cover generation and its transmittal on a DSS. The qualified DSS is the domain that is an RS-based of any value type including binary, decimal, hexadecimal, octal, alphabetic, alphanumeric, etc. This type of DSS, e.g. financial, medical, mathematical, scientific, economical, etc., has adequate room for concealing data and allows communicating parties to establish a covert channel, such as a relationship based on the profession of the communication parties, to transmit a text-cover. Since the generated text by NLG and template techniques is meaningful, rhetorically sound, semantically coherent, and legitimate, Matlist takes advantage of these techniques to generate noiseless text-cover. A message can be embedded in a form of an RS values, as a function of RS values, the related semantics of RS, a combination of these, etc. The experimental results confirmed that Matlist Cover is capable of fooling both human and machine examinations. The presented implementation achieves a superior bitrate to all other contemporary linguistic steganography approaches. Matlist is a truly a public methodology that does not rely on the secrecy of its technique. Improving the bitrate is worth investigating in the future.

Appendix

Sample 1: CONSUMER PRICE INDEX: DECEMBER 2006

The Consumer Price Index for All Urban Consumers (CPI-U) increased 0.1 per cent in December, before seasonal adjustment, the Bureau of Labor Statistics of the U.S. Department of Labor reported today. The December level of 201.8 (1982–1984 = 100) was 2.5 per cent higher than in December 2005.

The Consumer Price Index for Urban Wage Earners and Clerical Workers (CPI-W) increased 0.2 per cent in December, prior to seasonal adjustment. The December level of 197.2 (1982–1984 = 100) was 2.4 per cent higher than in December 2005.

The Chained Consumer Price Index for All Urban Consumers (C-CPI-U) increased 0.1 per cent in December on a not seasonally adjusted basis. The December level of 117.1 (December 1999 = 100) was 2.4 per cent higher than in December 2005. Please note that the indexes for the post-2004 period are subject to revision. CPI for All Urban Consumers (CPI-U) on a seasonally adjusted basis, the CPI-U increased 0.5 per cent in December, the first advance since August. Energy prices, which had declined in each of the preceding three months, rose 4.6 (cont.).

Sample 2: CONSUMER PRICE INDEX: Quarterly Report

As recently reported by our department, the consumer price index for the first product lost 0.43 per cent in the second period. However, the second period level increased 0.93 per cent higher than last year's second period.

The consumer price index for the second product elevated 0.109 per cent in the second period. The second period level ascended 0.83 per cent higher than last year's second period.

The consumer price index for the third product sank 0.4 per cent in the second period. However, the second period level barely budged 0.54 per cent higher than last year's second period. The consumer price index for the fourth product augmented 0.115 per cent in the second period, following a 0.33 per cent depressed value in the first period. The fifth product costs held 0.58 per cent in the first period while increasing 0.90 per cent in the second period. The index for the seventh product retained 0.45 per cent and the consumer price index for the eighth product held 0.56 per cent. The ninth product fell 0.20 per cent in the second period after falling 0.22 per cent in the first period. The index for the tenth product elevated 0.105 per cent, after being depressed by 0.33 per cent in five of the six major subgroups. The consumer index for all goods and services not including the eighth product and the ninth product barely budged 0.54 per cent in the second period, following a 0.90 per cent increase in the first period.

DOI: 10.1002/sec

The consumer price index for the eleventh product retained 0.48 per cent in the second period. However, the second period level sank 0.2 per cent lower than last year's second period.

In summation, the CPI rose during this study period. However, in the next quarter, the price trend is forecasted to drop.

Sample 3: CONSUMER PRICE INDEX: Quarterly Report

As recently reported by our department, the consumer price index for the first product lost 0.4 per cent in the second period. However, the second period increased 0.4 per cent higher than last year's second period.

The consumer price index for the second product elevated 0.5 per cent in the second period. However, the second period level ascended 0.4 per cent higher than last year's second period.

The consumer price index for the third product sank 0.5 per cent in the second period. However, the second period level barely budged 0.5 per cent higher than last year's second period. The consumer price index for the fourth product augmented 0.5 per cent in the second period, following a 0.4 per cent depressed value in the first period. The consumer price index for the fifth product held 0.4 per cent in the first period after increasing 0.1 per cent in the second period. The consumer price index for the seventh product retained 0.1 per cent and the index for the eighth product held 0.2 per cent. The consumer price index for the ninth product fell 0.1 per cent in the second period after falling 0.3 per cent in the first period. The consumer price index for the tenth product elevated 0.1 per cent, after being depressed by 0.4 per cent in five of the six major subgroups. The consumer index for all goods and services, not including the eighth product and the ninth product, barely budged 0.5 per cent in the second period, following a 0.1 per cent increase in the first period.

The consumer price index for the eleventh product retained 0.4 per cent in the second period. However, the second period level sank 0.3 per cent lower than last year's second period.

In summation, the CPI rose during this study period. However, in the next quarter, the price trend is forecasted to drop.

Sample 4: CONSUMER PRICE INDEX: October's and November's Report of 2006

As recently reported by our department, the CPI for food and beverages lost 0.1 per cent in the period of October to November. The CPI for food ascended 0.3 per cent in the period of September to October higher than the period of October to November. The CPI for food at home boosted 0.3 per cent in the period of September to October. The CPI for cereals and bakery products climbed 0.4 per cent in the period of October to November higher than the period of September to October.

The CPI for fruits and vegetables lost 2.2 per cent in the period of October to November. The CPI for dairy and related products depressed 0.6 per cent in the period of October to November. The CPI for Other food at home deflated 0.3 per cent in the period of October to November, after a 0.2 per cent retention in the period of September to October. The CPI for other foods held 0.1 per cent the period of September to October while decreasing 0.6 per cent in the period of October to November. The CPI for other miscellaneous foods moved up 0.1 per cent in the period of October to November after flopping 0.4 per cent in the period of September to October. CPI Alcoholic beverages fell 0.1 per cent in the period of October to November after boosting 0.2 per cent in the period of September to October.

The CPI for the apparel reduced 0.3 per cent in the period of October to November, after deflating 0.7 per cent in the period of September to October. The CPI for infants' and toddlers' apparel barely budged 0.1 per cent down in the period of October to November, after a 1.4 per cent rise in the period of September to October. The CPI for men's and boys' apparel dipped 1.0 per cent in the period of September to October. However, the period of October to November equated 0.9 per cent lower than the previous month. The CPI for the footwear increased 0.5 per cent in the period of September to October. However, CPI for the footwear in the period of October to November depressed to 0.0 per cent lower than last month. The CPI for the women's and girls' apparel depressed 1.2 per cent in the period of September to October. However, the period of October to November deflated 0.3 per cent lower than last month.

The CPI for the transportation reduced 3.1 per cent in the period of September to October. However, CPI for the transportation in the period of October to November devalued 0.9 per cent less than the previous month. The CPI for motor vehicle parts and equipment ascended 0.2 per cent in the period of September to October, following a 0.5 per cent escalation in the period of October to November. Finally, the CPI for the public transportation dropped 1.0 per cent in the period of September to October. However, the period of October to November dropped 1.9 per cent more than last month.

The upcoming Quarterly Report will include the period of November to December with a summarization of the entire quarter.

Sample 5:

Example 2:
Look at these numbers:
3, 7, 5, 13, 20, 23, 39, 23, 40, 23, 14, 12, 56, 23, 29
The sum of these numbers is equal to 330
There are fifteen numbers.
The mean is equal to 330 ÷ 15 = 22
The mean of the above numbers is 22

Sample 6:

Find the mean of the following numbers?
43 93 109 83 4 54 115 33 58 90 45 56 20 22 105 33 54 90 48 2

Sample 7:

Book title	Author	Price
Memoirs of Halide Edib	Halide Adivar Edib	$43.00
Mandeville's Used Book Price Guide	Richard L. Collins	$93.00
Conclog: A Methodological Approach to Concurrent Logic Programming	Jean-Marie Jacquet	$109.00
From Paris to Peoria: How European Piano Virtuosos Brought Classical Music to the American Heartland	R. Allen Lott	$83.00
Bank Shots (Kindle Edition)	Jim Strahle	$4.00
Grad Guides Book 3: Biological Sciences 2007	Peterson's	$54.00
Elements of Chemical Reaction Engineering (3rd Edi.)	H. Scott Fogler	$115.00
Good Practice Teacher's Book	Marie McCullagh, Ros Wright	$33.00
DDC Learning Macromedia Flash 5	Suzanne Weixel	$58.00
Frontiers of Combining Systems 2 (Studies in Logic and Computation)	Dov M. Gabbay	$90.00
THE SPIDER - Master of Men - Volume 6, number 1 - June 1935	John Howitt; J. Fleming	$45.00
Murder In The Dark	Kerry Greenwood	$56.00
Hands-On Celebrations	Yvonne Y. Merrill	$20.00
Ramayana Book Two: Ayodhya	Valmiki, Sheldon Pollock	$22.00
Black History	Mary Ellen Snodgrass	$105.00
The Trail of Tears	John P. Bowes	$33.00
The Celebrity Black Book 2008	Jordan McAuley	$54.00
Nonlinear Dynamics and Chaos	J. M. T. Thompson, H. B. Stewart	$90.00
How to Establish a Unique Brand in the Consulting Profession	Alan Weiss	$48.00
Good and Evil Coloring Book #1	Michael Pearl	$2.00

Sample 8:

Assume that the freezing point and melting point of antifreeze, which are - 45°C and 115°C respectively, are defined as 0°A and 100°A respectively. What is the boiling point of water in °A?

a) 83 b) 95 c) 78 d) 105 e) 91

Sample 9:

The ultimate temperature is _________ if the exact heat of Al is 0.21 cal/g °C when 152 g of Al at 75.0 °C in 145 g of H_2O at 23.5 °C.

a) 58 b) 94 c) 14 d) 120.1 e) 32.8

Sample 10:

$q \notin r$ True [ ] or False [ ]
if q={J,L,K} and r={D,H,L,I,A}

REFERENCES

1. Petitcolas FAP, Anderson RJ, Kuhn MG. Information hiding – a survey. *Proceedings of the IEEE* 1999; **87**(7): 1062–1078.
2. Kipper G. Investigator's Guide to Steganography, CRS Press Inc.: Boca Raton, FL, USA, 2004; 15–16.
3. Davern P, Scott M. Steganography its history and its application to computer based data files. Technical Report Internal Report Working Paper: CA-0795, School of Computing, Dublin City University 1995.
4. Johnson NF, Jajodia S. Exploring steganography: seeing the unseen. *IEEE Computer* 1998; **31**(2): 26–34.
5. Reiter E, Dale R. Building Natural Language Generation Systems. Cambridge University Press: UK, 2000.
6. Kahn D. The Codebreakers: The Story of Secret Writing, (revised edn). Scribner: New York, USA, 1996.
7. Bennett K. Linguistic steganography: survey, analysis, and robustness concerns for hiding information in text.

Technical Report CERIAS Tech Report 2004-13, Purdue University, 2004.

8. Kessler GC. An overview of steganography for the computer forensics examiner. An edited version, issue of Forensic Science Communications. *Technical Report* 2004; **6**(3).
9. Wayner P. Mimic functions. *Cryptologia* 1992; **XVI**(3): 193–214.
10. Wayner P. Disappearing Cryptography, (2nd edn). Morgan Kaufmann: San Francisco, CA, USA, 2002; 81–128.
11. Chapman M, Davida G. Hiding the hidden: a software system for concealing ciphertext as innocuous text. In *Proceedings of the International Conference on Information and Communications Security.* Vol. 1334 of Lecture Notes in Computer Science, Springer, pp.335–345, Beijing, P.R. China, November 1997.
12. Chapman M. Hiding the hidden: a software system for concealing ciphertext as innocuous text. *Master's Thesis*. University of Wisconsin-Milwaukee, May 1997.
13. Chapman M, Davida GI. Nicetext system official home page. www.nicetext.com
14. Chapman M, Davida GI. Plausible deniability using automated linguistic steganography. In International Conference, InfraSec 2002 Bristol, UK, October 1–3, 2002 Proceedings, G Davida and Y Frankel, (eds). Springer: Berlin, Heidelberg, 2002; 276–287.
15. Chapman M, Davida GI, Rennhard M. A practical and effective approach to large-scale automated linguistic steganography. In Proceedings of the Information Security Conference (ISC ′), volume 2200 of Lecture Notes in Computer Science. Springer: Malaga, Spain, 2001; 156–165.
16. Winstein K. Lexical steganography through adaptive modulation of the word choice hash, January 1999. alumni.imsa.edu/<keithw/tlex/lsteg.ps
17. Winstein K. Lexical steganography. alumni.imsa.edu/<keithw/tlex
18. Bolshakov IA. A method of linguistic steganography based on collocationally-verified synonymy. In Information Hiding 6th International Workshop, IH 2004, Toronto, Canada, May 23–25, 2004, Revised Selected Papers, Fridrich JJ (ed.). Springer: Berlin, Heidelberg, 2004; 180–191.
19. Bolshakov IA, Gelbukh A. Synonymous paraphrasing using wordnet and internet. In Farid Meziane and Elisabeth Elisabeth Metais, (eds). *Natural Language Processing and Information Systems: 9th International Conference on Applications of Natural Language to Information Systems*, NLDB2004, volume 3136 of Lecture Notes in Computer Science, pp. 312-323. Springer, June 2004.
20. Calvo H, Bolshakov IA. Using selectional preferences for extending a synonymous paraphrasing method in steganography. In JH Sossa Azuela (ed). *Avances en Ciencias de la Computacion e Ingenieria de Computo - CIC'2004: XIII Congreso Internacional de Computacion*, pp. 231–242, October 2004.
21. Chand V, Orgun CO. Exploiting linguistic features in lexical steganography design and proof-of-concept implementation. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS' 06)* Volume 6, page126b, IEEE January 2006.
22. Nakagawa H, Sampei K, Matsumoto T, Kawaguchi S, Makino K, Murase I. Text information hiding with preserved meaning - a case for japanese documents. IPSJ Transaction, 42(9):2339-2350, 2001. Translated to English: www.r.dl.itc.u-tokyo.ac.jp/nakagawa/academic-res/finpri02.pdf
23. Niimi M, Minewaki S, Noda H, Kawaguchi E. A framework of text-based steganography using sd-form semantics model. IPSJ Journal, 44(8), August 2003. www.know.comp.kyutech.ac.jp/STEG03/STEG03-PAPERS/papers/12-Niimi.pdf
24. Bergmair R, Katzenbeisser S. Content-aware steganography: About lazy prisoners and narrow-minded wardens. In Proceedings of the 8th Information Hiding Workshop, Lecture Notes in Computer Science. Volume 4437, 109-123, Springer Verlag, Berlin/Heidelberg, September 2007. In print.
25. Bergmair R. Towards linguistic steganography: a systematic investigation of approaches, systems, and issues. *B.Sc. Project*, April 2004, the University of Derby.
26. Bergmair R, Katzenbeisser S. Towards human interactive proofs in the text-domain. In *Proceedings of the 7th Information Security Conference (ISC'04)*, Springer Lecture Notes in Computer Science, September 2004.
27. Topkara U, Topkara M, Atallah MJ. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *MM&Sec '06: Proceeding of the 8th workshop on Multimedia and security*, pp.164–174, New York, USA, 2006. ACM Press.
28. Murphy B, Vogel C. The syntax of concealment: reliable methods for plain text information hiding. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, January 2007.
29. Atallah MJ, Raskin V, Crogan M, Hempelmann C, Kerschbaum F, Mohamed D, Naik S. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In Ira S. Moskowitz, (ed.). *Information Hiding: Fourth International Workshop*, Volume 2137 of Lecture Notes in Computer Science, pp. 185–199. Springer, April 2001.
30. Atallah MJ, Raskin V, Hempelmann C, Topkara M, Sion R, Topkara U, Triezenberg KE. Natural language water-

marking and tamperproofing. In Fabien AP Petitcolas (ed). Information Hiding: Fifth International Workshop, volume 2578 of Lecture Notes in Computer Science, pp. 196–212. Springer: October 2002.

31. Grothoff C, Grothoff K, Alkhutova L, Stutsman R, Atallah M. Translation-based steganography, *Technical Report CSD TR# 05-009*, Purdue University, 2005. (CERIAS Tech Report 2005-39).
32. Grothoff C, Grothoff K, Stutsman R, Alkhutova L, Atallah M. Translation-based steganography. In *Proceedings of Information Hiding Workshop (IH 2005)*, pp. 213-233. Springer-Verlag, Barcelona, Spain, June 2005.
33. Stutsman R, Grothoff C, Atallah M, Grothoff K. Lost in just the translation. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC' 06)*, Dijon, France, April 2006.
34. Topkara M, Topkara U, Atallah MJ. Information hiding through errors: a confusing approach. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, January 2007.
35. Shirali-Shahreza M, Shirali-Shahreza MH. Text steganography in SMS. *International Conference on Convergence Information Technology*, Vol., Issue, 21-23, pp.2260-2265, Nov. 2007.
36. Desoky A. Nostega: a novel noiseless steganography paradigm. *Journal of Digital Forensic Practice* 2008; **2**(3): 132–139
37. Desoky A. Nostega: a novel noiseless steganography paradigm, *Ph.D. Dissertation*, University of Maryland, Baltimore County, May 2009.
38. Desoky A, Younis M, El-Sayed H. Auto-summarization-based steganography. In *Proceedings of the 5th IEEE International Conference on Innovations in Information Technology*, Al-Ain, UAE, December 2008.
39. Desoky A. Listega: list-based steganography methodology. *International Journal of Information Security* 2009; **8**(4): 247–261.
40. Desoky A. Notestega: notes-based steganography methodology. *Information Security Journal: A Global Perspective* 2009; **18**(4): 178–193.
41. Kukich K. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting of the ACL, Massachusetts Institute of Technology*, Cambridge, MA, pp.145-150, June 15–17, 1983.
42. CoGenTex Inc., WeatherReporter www.cogentex.com
43. Ana the Stock Reporter (StockReporter) www.ics.mq.edu.au/<ltgdemo/StockReporter/about.html
44. Koblitz N. A Course in Number Theory and Cryptography, (2nd edn). Springer-Verlag New York, Inc.: New York, NY, USA, 1994; 54–76.
45. Johnson NF, Katzenbeisser S. A Survey of steganographic techniques. In Information Hiding, S, Katzenbeisser F Petitcolas (eds). Artech House: Norwood, MA, 2000; 43–78.
46. Consumer Price Index: December 2006: ftp://ftp.bls.gov/pub/news.release/History/cpi.01182007.news
47. Desoky A, Younis M. PSM: Public Steganography Methodology, Technical Report TR-CS-06-07, Department of Computer Science and Electrical Engineering. University of Maryland: Baltimore County, 2006.
48. Desoky A, Younis M. Graphstega: graph steganography methodology. *Journal of Digital Forensic Practice* 2008; **2**(1): 27–36.
49. U.S. Bureau of Labor Statistics, Full CPI: www.bls.gov/cpi/cpid0612.pdf
50. Elementary Math: www.mathsisfun.com/mean.html
51. Elementary Math: www.rbechtold.com/math4.html
52. Chemistry quizzes by the Department of Chemistry The Ohio State University. lrc-srvr.mps.ohio-state.edu/under/chemed/qbank/quiz/bank1.htm
53. Ralph P. Grimaldi Discrete Combinatorial Mathematics: An Applied Introduction, (3rd edn). Pearson: USA 1994.
54. Spam Mimic: www.spammimic.com
55. Zipf GK. (Introduction by Miller, G.A.) The Psycho-Biology of Language: An Introduction to Dynamic Philology. MIT Press: Cambridge, MA, 1968.
56. Li W. Random texts exhibit Zipf's-law-like word frequency distribution. *IEEE Transactions on Information Theory* 1992; **38**(6): 1842–1845.
57. Consumer Price Index (Archive): ftp://ftp.bls.gov/pub/news.release/History
58. Pfleeger CP. Security In Computing. Prentice-Hall Inc.: NJ, 2000; 21–65.
59. Stallings W. Cryptography and Network Security: Principles And Practices. Prentice-Hall Inc.: NJ, 2003; 21–50.
60. Lewand RE. Cryptological Mathematics. The MAA Inc.: DC, 2000; 1–44.
61. Graduate Catalog 2005-2006 University Of Florida, Gainesville: gradschool.rgp.ufl.edu/current-files/current-catalog.pdf
62. University of Minnesota: Modeling and Analysis of Flexible Queueing Systems: www.ie.umn.edu/faculty/faculty/pdf/nrl.pdf
63. University of Otago, Postgraduate Catalog of Zoology: www.otago.ac.nz/Zoology/pdf/postgraduate_handbook.pdf
64. The Los Angeles Superior Court Civil General Information: http://www.lasuperiorcourt.org/civil/main.htm#3
65. The PDP: userpages.umbc.edu/<crandall/index.htm