Headstega: e-mail-headers-based steganography methodology

Abdelrahman Desoky

Department of Computer Science and Electrical Engineering,
University of Maryland Baltimore County,
Baltimore, MD 21250, USA
E-mail: abd1@umbc.edu

Abstract: The frequent exchange of e-mails is widely popular and generates a high volume of traffic that allows communicating parties to establish a covert channel without a suspicious pattern rendering e-mails an attractive steganographic carrier to transmit hidden messages. This was the motive of developing e-mail-headers-based steganography methodology (Headstega). Headstega encodes a message then assigns it to steganographic carriers (e-mail-headers), e.g. recipient's e-mail addresses, names, subject fields, etc., in order to camouflage data by following Nostega paradigm. Thus, Headstega neither hides data in a noise (errors) nor produces noise while a message is concealed in the textual e-mail headers and the e-mail contents (the body of e-mails) are completely legitimate and do not conceal data. The presented implementation, validation and steganalysis of Headstega demonstrate: the robust capabilities of achieving the steganographic goal, the adequate room for concealing data and the superior bitrate to all contemporary text steganography approaches, which is roughly 3.38–7.67%.

Keywords: steganography; information hiding; security.

Reference to this paper should be made as follows: Desoky, A. (2010) 'Headstega: e-mail-headers-based steganography methodology', *Int. J. Electronic Security and Digital Forensics*, Vol. 3, No. 4, pp.289–310.

Biographical notes: Abdelrahman Desoky is a Scientist with more than 18 years experience in the computer field. He is an Independent Consultant, Researcher and Instructor. He is an author of a steganography book and numerous papers. His book is entitled 'Noiseless stenography: the key of covert communications'. He received his PhD from the University of Maryland and his MSc from the George Washington University; both degrees are in Computer Engineering. His PhD dissertation is entitled 'Nostega: a novel noiseless steganography paradigm'. His MSc concentrated on computer architecture and networks. His MSc research is focused and was entitled 'Security architecture for computers and networks'.

1 Introduction

Steganography is the scientific art of concealing the presence of covert communications. Practically, the steganographic goal is to prevent an adversary from suspecting the existence of covert communications (Desoky, 2010, in process). Unlike cryptography, the aim of steganography is not to impede an attacker from decipher a hidden message

like ciphertext. To emphasise, when using any steganographic technique if suspicion is raised, the goal of steganography is defeated regardless of whether or not a plaintext is revealed (Desoky, in process; Martin et al., 2005). Contemporary approaches in the literature are often classified based on the steganographic cover type, e.g. image, audio, graph (Desoky, 2008b; Desoky and Younis, 2006), text, etc. When textual is employed for hiding data and generating the steganographic cover, an approach is usually categorised as textual steganography to distinguish it from non-textual techniques, e.g. image, audio, etc. Textual steganography has become more favourable in recent years since the size of non-textual covers is relatively large and is burdening the traffic of covert communications (Bennett, 2004; Martin et al., 2005; Petitcolas, 1999).

Contemporary steganography techniques hid a message as noise in a cover that is assumed to be unnoticeable. For instance, an encoded message can be embedded into an image by altering it without noticeable degradation (Martin et al., 2005; Petitcolas, 1999). Similarly, hiding a message in a text by modifying the format and style of an existing text (Bennett, 2004; Desoky, 2010, in process; Shirali-Shahreza and Shirali-Shahreza, 2006). Obviously, such alteration of authenticated covers can raise suspicion and the message is detectable regardless of whether or not a plaintext is revelled (Bennett, 2004; Petitcolas, 1999). The same applies to hiding the data in unused or reserved space for systems software, e.g. the designated storage area of an operating system, the file headers on a hard drive, etc. (Anderson et al., 1998; ScramDisk), or in the packet headers of communication protocols, e.g. TCP/IP packets transmitted across the internet (Handel and Sandford, 1996). These techniques are vulnerable to distortion attacks (Desoky, 2010, in process; Petitcolas, 1999).

On the other hand, a similar argument is made in the literature about textual steganography approaches, such as null-cipher (Kahn, 1996), mimic functions (Wayner, 1992, 2002), NICETEXT and SCRAMBLE (Chapman and Davida, 1997, 2002, 2007; Chapman et al., 2001), translation-based (Grothoff et al., 2005a,b; Stutsman et al., 2006), confusing approach (Topkara et al., 2007) and abbreviation-based (Shirali-Shahreza et al., 2007). The vulnerability and concerns of these textual approaches, as explained in Section 2, can be summarised as follows. Firstly, the textual-cover either introduces detectable flaws (noise), such as incorrect syntax, lexicon, rhetoric, grammar, etc. when generating a text-cover. Obviously, such flaws can raise suspicion about the presence of covert communications. Secondly, the content of the cover may be meaningless and semantically incoherent, and thus may draw suspicion. Thirdly, the bitrate is very small. Since there is a limit on how many flaws a document may typically have, very large documents will be needed to hide few bytes of data. In fact, this applies to non-textual approaches as well. Fourthly, the bulk of the efforts have been focused on how to conceal a message and not on how to conceal the transmittal of the hidden message. In other words, the establishment of a covert communication channel has not been an integral part of most approaches found in the literature. Finally, while these approaches may fool a computer examination, they often fail to pass human inspections. A successful textual steganography approach must be capable of passing both computer and human examinations. These concerns have motivated the development of the e-mail-*head*ers-based *stega*nography methodology (Headstega), introduced in this paper.

Headstega overcomes the issues just mentioned above by only manipulating the textual e-mail headers to camouflage both a message and its transmittal. Basically, Headstega exploits textual elements of e-mail headers data, such as recipient e-mail addresses, names, subject, etc. to conceal messages. Such elements can be fabricated in a

legitimate way in order to embed data without generating any type of suspicious pattern. Simply, it encodes a message then assigns it to such legitimate elements in order to generate a text-cover in a form of e-mail headers. The main advantages of Headstega are as follows. Firstly, the high demand for using e-mails by a wide variety of people creates a high volume of traffic and averts suspicion in the presence of covert communication channels. Secondly, Headstega does not imply a particular pattern (noise) that an adversary may look for. Thirdly, the concealment process of Headstega has no effect on the linguistics of the generated cover (head-cover) because there no linguistic structures are required in e-mail headers to be obeyed. Therefore, a head-cover is linguistically legible and is thus capable of passing both computer and human examinations. Fourthly, Headstega can be applied to all languages. Fifthly, the textual of e-mail headers has plenty of room for concealing data, as demonstrated later in this paper. The observed average bitrate of the current implementation experiments is superior to all contemporary textual steganography approaches found in the literature which roughly 3.38–7.67%. Finally, Headstega is resilient to popular attacks and the hidden message is antidistortion. The implementation and steganalysis validation demonstrate that Headstega methodology is capable of achieving the steganographical goal.

The remainder of this paper is organised as follows. Section 2 describes and discusses the related work found in the literature. Section 3 explains the Headstega methodology in detail. Section 4 demonstrates the Headstega implementation. Section 5 presents the steganalysis validation of Headstega. Finally, Section 6 concludes this paper.

2 Related work

Steganography is the science and art of camouflaging the presence of covert communications. The origin of steganography is traced back to early civilisations (Desoky, 2010, in process). The ancient Egyptians communicated covertly using the Hieroglyphic language, a series of symbols representing a message (Desoky, 2010, in process, 2010). The message looks as if it is a drawing of a picture although it may contain a hidden message that only a specific person who knew what to look for can detect. The Greeks also used steganography, 'hidden writing,' where the name was derived. The aim of this section is to present and discuss the contemporary textual and non-textual steganography approaches as follows.

2.1 Text steganography

Textual steganography approaches conceal data in a text-cover. These approaches can be categorised as follows.

- *Textual format manipulation (TFM)*: this is a non-linguistic steganography technique that hides data by exploiting the format of text (Petitcolas, 1999). TFM modifies an original text by employing spaces, misspellings, fonts, font size, font style, colours and non-colour (as invisible ink) to embed an encoded message. However, comparing the original text vs. the modified text triggers suspicion and enables an adversary to detect where a message is hidden. In addition, TFM can be distorted and may be discerned by human eyes or detected by a computer (Bennett, 2004; Petitcolas, 1999).

- *Series of characters and words*: during World War I, the Germans communicated covertly using a series of characters and words known as null-cipher (Kahn, 1996). 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. Apparently, suspicion is raised because the user is forced to fabricate a text-cover according to a predetermined protocol, which may introduce some peculiarity in the text that draws suspicion and defeats the steganographical goal. In addition, applying a brute force attack may reveal the entire message.
- *Statistical based*: Wayner has introduced the mimic functions approach (Wayner, 1992, 2002) which employs the inverse of the Huffman Code by inputting a data stream of randomly distributed bits to produce text that obeys the statistical profile 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. The output of regular mimic functions is gibberish rendering it extremely suspicious (Bennett, 2004; Petitcolas, 1999). However, the combination of mimic functions and CFG slightly improved the readability of the text (Wayner, 1992, 2002). Yet, the text-cover still contains numerous flaws, such as incorrect syntax, lexicon, rhetoric and grammar. In addition, the content of the text-cover is often meaningless and semantically incoherent. These shortcomings may raise suspicion in covert communications.
- *Synonym based*: Chapman and Davida (2007) have introduced a steganographic scheme consisting of two functions called NICETEXT and SCRAMBLE that use a large dictionary (Chapman and Davida, 1997, 2002, 2007; Chapman et al., 2001). NICETEXT 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 in the last decade: Winstein (2008a,b), Bolshakov (2004), Bloshakov and Gelbukh (2004), Calvo and Bolshakov (2004), Chand and Orgun (2006), Nakagawa et al. (2001), Niimi et al. (2003), Bergmair and Katzenbeisser (2004), Bergmair (2004), Bergmair and Katzenbeisser (2004), Topkara et al. (2006), Murphy and Vogel (2007) and Atallah et al. (2001, 2002). Although 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.
- *Noise based*: Grothoff et al. have introduced the translation-based steganographic scheme (Grothoff et al., 2005a,b; Stutsman, 2006) to hide a message in the errors (noise) that are naturally encountered in a machine translation (MT). This approach embeds a message by performing a substitution procedure on the translated text using translation variations of multiple MT systems. In addition, it inserts popular errors of MT systems and also uses synonym substitutions in order to increase the bitrate. Unlike synonyms-based steganography, linguistic flaws in noise-based approach are not a concern unless they appear excessively. However, Grothoff et al.

state that one of the concerns is that the continual improvement of MT may narrow the margin of hiding data. In addition, translation-based approach, as pointed out by Grothoff et al., cannot be applied to all languages because of the fundamental structures are radically different. This generates severely incoherent and unreadable text (Grothoff et al., 2005a,b; Stutsman et al., 2006). On the contrary, Headstega can be applied to all known languages without any exceptions while the generated (text-cover) head-cover is linguistically legitimate. This because nowadays e-mails can be in any language as such allowing the frequent of exchange e-mails in any language while data are concealed in the header of e-mail instead of concealing it in the body of e-mails.

Another noise-based approach has been proposed by Topkara et al. that employs typos and ungrammatical abbreviations in a text, e.g. e-mails, blogs, forums, etc. for hiding data (Topkara et al., 2007). Moreover, Shirali-Shahreza et al. have introduced an abbreviation-based scheme (Shirali-Shahreza et al., 2007) to conceal data using the short message service (SMS) of mobile phones. Due to size constraints of SMS and the use of phone keypad instead of the keyboard, a new language called SMS-Texting was defined to make the approach more practical. However, these approaches are sensitive to the amount of noise (errors) that occurs in a human writing. Such shortcoming not only increases the vulnerability of the approach but also narrows the margin of hiding data. Conversely, Headstega neither employs errors nor uses noisy text to conceal data.

- *Nostega-based*: recently, the new paradigm in steganography research, namely noiseless steganography paradigm (Nostega), has been introduced (Desoky, 2008a, 2009), in which the 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 *sum*marisation-based *stega*nography methodology (Sumstega) (Desoky et al., 2008). Sumstega exploits automatic summarisation techniques to camouflage data in the auto-generated summary-cover (text-cover) that looks an ordinary and legitimate summary. The second linguistic steganographic scheme that is also based on Nostega paradigm is the list-based steganography methodology (Listega) (Desoky, 2009a). Listega manipulates itemised data to conceal messages in a form of textual list. The third linguistic steganography methodology, notes-based steganography methodology (Notestega) (Desoky, 2009c) that takes advantage of the recent advances in automatic notetaking techniques to generate a text-cover. Notestega pursues the variations among both human notes and the outputs of automatic notetaking techniques to conceal data. The fourth linguistic steganography methodology, mature linguistic steganography methodology (Matlist) (Desoky, in press), employs random series of a domain specific subject along with NLG and template techniques to generate a text-cover that is naturally has a different legitimate meaning for concealing different messages while it remains semantically coherent and rhetorically sound.

It is worth noting that the presented Headstega methodology in this paper follows this new paradigm by exploiting e-mail headers to camouflage data without generating any suspicious pattern.

2.2 Non-textual steganography

Non-textual steganography approaches can be categorised based on its file type, such as image, audio and graph. Image steganography is based on manipulating digital images to conceal a message. Such manipulation often renders the message as noise. In general, image steganography suffers from several issues such as the potential of distortion, the significant size limitation of the messages that can be embedded and the increased vulnerability to detection through digital image processing techniques (Martin et al., 2005). Audio-covers have also been pursued. Example of audio steganography techniques include LSB (Cvejic and Seppanen, 2004a,b), spread spectrum coding (Bender et al., 1996; Kirovski and Malvar, 2001), phase coding (Ansari et al., 2004; Bender et al., 1996) and echo hiding (Ansari et al., 2004; Gruhl et al., 1996). In general, these techniques are too complex, and like their image-based counterpart, are still subject to distortion and are vulnerable to detection (Cvejic and Seppanen, 2004b; Desoky, 2010, in process; Martin et al., 2005; Petitcolas, 1999). The hidden message may become to a great extent a foreign body in the cover and thus makes those schemes vulnerable to detection. In addition, contemporary steganography schemes rely on private or restricted access to the original unaltered cover in order to avoid the potential of comparison attacks, which is considered as a major threat to the covert communication. Basically, an adversary can detect the presence of a hidden message by comparing a particular image- or audio-cover to the original image or audio file and finding out that some alterations have been made.

Hiding information in an unused or reserved space in computer systems (Anderson et al., 1998; ScramDisk). For example, Windows 95 operating system has around 31 kB unused hidden space which can be used to hide data. Another example, unused space in file headers of image, audio, etc. can also be used to hide data. This depends on the size of the hard drive used. TCP/IP packets used to transport information across the internet have unused space in the packet headers (Handel and Sandford, 1996). The TCP packet header has six unused (reserved) bits and the IP packet header has two reserved bits. There are tremendous packets that are transmitted over the internet can convey and transmit a secret data. However, these techniques are vulnerable to distortion attacks (Desoky 2010, in process; Petitcolas, 1999).

Recently, a graph steganography (Graphstega) methodology has been developed (Desoky, 2008b; Desoky and Younis, 2006). Unlike all other schemes, the message is naturally embedded in the cover by simply generating the cover based on the message. Graphstega camouflages a message as data points in a graph and thus the message would not be detectable as noise. The approach is shown to be resilient to a wide range of attacks, including a comparison attack by untraceable or authenticated data. Similarly, Chestega (Deskoy and Younis, in press) exploits popular games, such as chess, checkers, crosswords, domino, etc. for concealing messages in an unaltered authenticated data. Graphstega and Chestega represent a new paradigm in steganography research in which the message is hidden in the cover as data rather than noise. Headstega follows this new paradigm by exploiting textual e-mail headers to camouflage data without generating any suspicious pattern.

3 Headstega methodology

To illustrate Headstega, consider the following scenario. Bob and Alice are on a spy mission. Bob and Alice like any other ordinary people each one owns an e-mail account. 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 friendship as a steganographic umbrella to justify sending and receiving e-mails. Basically, they agree on concealing messages only in e-mail headers by embedding data in a form of e-mail addresses, names, individual titles, abbreviations, etc. in such a way that looks unsuspicious, whereas the context of e-mails (the body of e-mails) is fully legitimate and nothing is concealed in it. To make this work, Bob and Alice, by legitimate reason, have the right to send, receive and forward e-mails, e.g. personals, advertising, business, invitation, etc. for either an individual or a group of people, e.g. group of friends. Covert messages transmitted in this manner will not look suspicious because the relationship between Bob and Alice is legitimate which justifies the discernable communications. Furthermore, Alice is not always the sole recipient of Bob's e-mail and *vice versa*. However, other non-spy people also receive such e-mails further warding off suspicion. As mentioned earlier, these e-mails conceal data only in the headers whereas the contexts (e-mail bodies) are fully legitimate and do not camouflage any message fooling an adversary. However, only Bob and Alice will be able unravel the hidden message because they know the rules of the game.

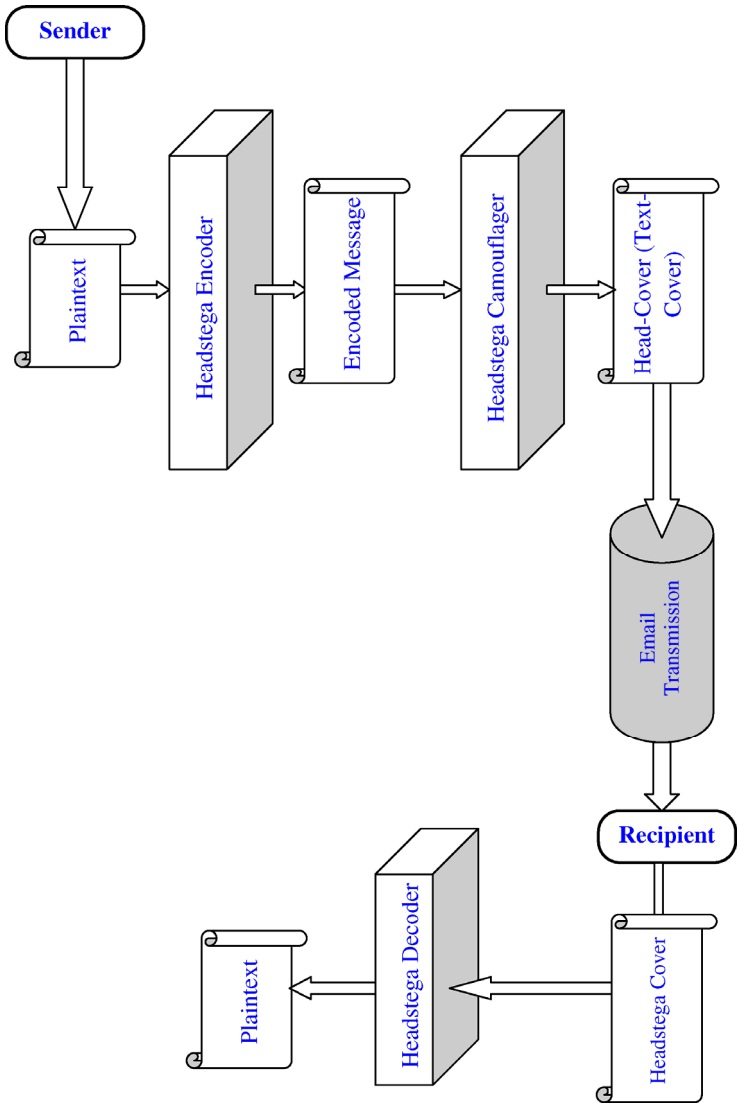
The above scenario demonstrates how Headstega methodology can be used. Headstega methodology takes advantages of the common frequent exchange of e-mails, which is popular worldwide and generates a high volume of traffic, to conceal both a message and its transmittal, as illustrated earlier by above scenario of Bob and Alice. The core idea of Headstega methodology is basically camouflaging data in the natural and legitimate e-mail headers such as recipient's e-mail addresses, names, subject, etc. Obviously, such steganographic covers linguistically and logically is legitimate. The Headstega's algorithm works as follows. Mainly, the overview of Headstega system architecture consisted of two modules, as shown in Figure 1. First module is the message encoding, which encodes a message in an appropriate and required form for the camouflaging process (the second module). The process of generating a head-cover, by Module 2, may influence the process of how a message should be encoded. For example, a message may be encode by slicing its binary string into a particular length of bits, such as four bits, seven bits or any required bit length. For instance, a message: 'no more' can be converted to its binary: '011011100110111100100000011011010110111011100100110010101'. Then slicing its binary string into a particular length of bits such as four bits: '0110 1110 0110 1111 0010 0000 0110 1101 0110 1111 0111 0010 0110 0101'. The second module is the message camouflager that generates the head-cover (text-cover), which conceals the encoded message. The head-cover may be in a form of recipient's e-mail addresses, recipient's names, subjects, etc. Implementing such camouflager scheme may involve employing some other components in order to ease automation process of generating head-cover like the following:

- E-mail Generator systems, such as iContact.
- Internet search engine, such as the free search engine of Google Internet Search Engine, Yahoo Internet Search Engine , Live Search Internet Search Engine, etc.

- E-mail accounts providers such as the free Yahoo Mail, Gmail, MSN Hotmail, etc.

Once the Headstega system is implemented, the covert communications will be accomplished in three steps. Firstly, it encodes a message using the predetermined steganographical encoder from Module 1. Secondly, Module 2 camouflages the steganographic code (encoded message) which is generated by Module 1. Finally, it e-mails the message to it's litigate recipient. The above modules are detailed in the following sections.

Figure 1 Illustrates the architecture of Headstega system (see online version for colours)



3.1 Message encoding (Module 1)

Implementing the message encoder follows a two-step process. Firstly, determining the encoding parameters that will play the roll of steganographic carriers. Secondly, defining a steganographic coding based on these parameters. A parameter (steganographic carrier) in this context means some aspects of text used that can be referred to steganographical values throughout a head-cover (text-cover). In other words, the steganographic carriers used by Headstega are the textual elements that are commonly used in the e-mail headers, such as recipient's e-mail addresses, names, subject, etc. will be assigned steganographical code values such as a particular binary bit string, e.g. '0000', '0001', '0010', etc. to conceal data. The definition of the steganographic code would depend on the selected parameters. For example, encoding a message by using a recipient's e-mail addresses is different from encoding it using recipient's names or subject, etc. The coding module of Headstega exploits theses type of options and determines the parameter(s) that will be employed for concealing messages. The popularity of certain parameters is an important factor in the selection. Nonetheless, unusual appearance of certain type of elements may draw suspicion. For example, a husband sending e-mails to his wife contains a private speech will be sent only to her not to a group of people. Such an e-mail, which contains a private speech from a husband to his wife and 'Cc' a group of people, if occurred is an unusual manner that can raise suspicion. Headstega methodology counters all of these concerns by simply imposing on the implementation of Headstega system to be made aware of such issues or attacks. In addition, it is also crucial for justifying the interaction among the communicating parties to establish a covert channel for delivering the steganographic cover (head-cover) which is done by the communication protocol (Module 3). Note that the encoding parameters may influence the covert channel or *vice versa*, which is responsible for justifying the interaction among the communicating parties.

Headstega does not impose any constraint on the message encoder scheme as long as it generates a set of data values that can be embedded in a head-cover. Given the availability of numerous encoding techniques in the literature that fit (Desoky, 2010, in process; Petitcolas, 1999), the balance of this section will focus on an example that will be used in Section 4 to demonstrate the applicability of Headstega. In the presented examples in Section 4, the encoding is done as follows. A message is first converted to a binary string. The string can be a binary of cipher text or a compressed representation. The binary string is then partitioned into groups of m bits. The value of m is determined based on the encoding parameters that Headstega exploits. In the textual of e-mail headers, such as the recipient's e-mail addresses, recipient's names, subject fields, etc. may be exploited for concealing data. For example, if the head-cover will be four possible elements (steganographic carriers), the binary message is partitioned it into groups of two bits, e.g. 00, 01, 10 and 11, corresponding to the possible options. Again, this encoding scheme is just for illustration and many alternate and more sophisticated schemes can be employed, as demonstrated in Section 4.

3.2 Message camouflager (Module 2)

As mentioned earlier, the high demand and popularity of using e-mails by a wide variety of people render e-mails attractive steganographic carriers. Headstega takes advantage of such demand and popularity to camouflage data only in e-mail headers and not in the

context (not in the body of e-mails) in a form of recipient's e-mail addresses, names, abbreviations, subject, etc. in order to embed messages without generating any suspicious pattern. From a steganographical point of view, reusing or altering an existing text to hide data is not a recommended practice since an adversary can reference the original text and detect the differences. In addition, the reuse of same piece of text more than once may increase vulnerability of the covert communications. If an adversary intercepts the communications and oversees a similar piece of text over and over again between communicating parties, suspicion may be raised because the adversary will wonder of such use. Inversely, this is not a concern in Headstega methodology because reusing recipient's e-mail addresses, names, abbreviations, subject, etc. are a common practice for sending e-mails, as shown in Figure 2. Such strong feature, in Headstega system, eases the automation of a text-cover (head-cover). Note, Headstega system considers all aspects that can ensure the success for the covert communications and avoids all aspects that can cause failure. The following is fundamental idea of headstega camouflager algorithm that automates the generation process of head-cover (text-cover). Simply, Headstega system generates head-cover (text-cover) such as e-mail addresses, names, individual titles, abbreviations, etc. by either fabricating that textual e-mail headers or generating it from actual collection of legitimate e-mail headers. For example, a steganographic carrier may look as follows:

- *'Individual Titles'<USER_ID@DOMAIN_NAME.Extension>*

Individual titles, such as Prof., Dr., CEO, etc. can be employed to conceal data.

- *'Abbreviations'<USER_ID@DOMAIN_NAME.Extension>*

Abbreviations of labs, division, groups, teams, etc. can be employed to conceal data.

- *'Names'<USER_ID@DOMAIN_NAME.Extension>*

Names such as First Name, Middle Name, Last Name, Nicknames, etc. can be employed to conceal data.

- *USER_ID@ DOMAIN_NAME.Extension>*

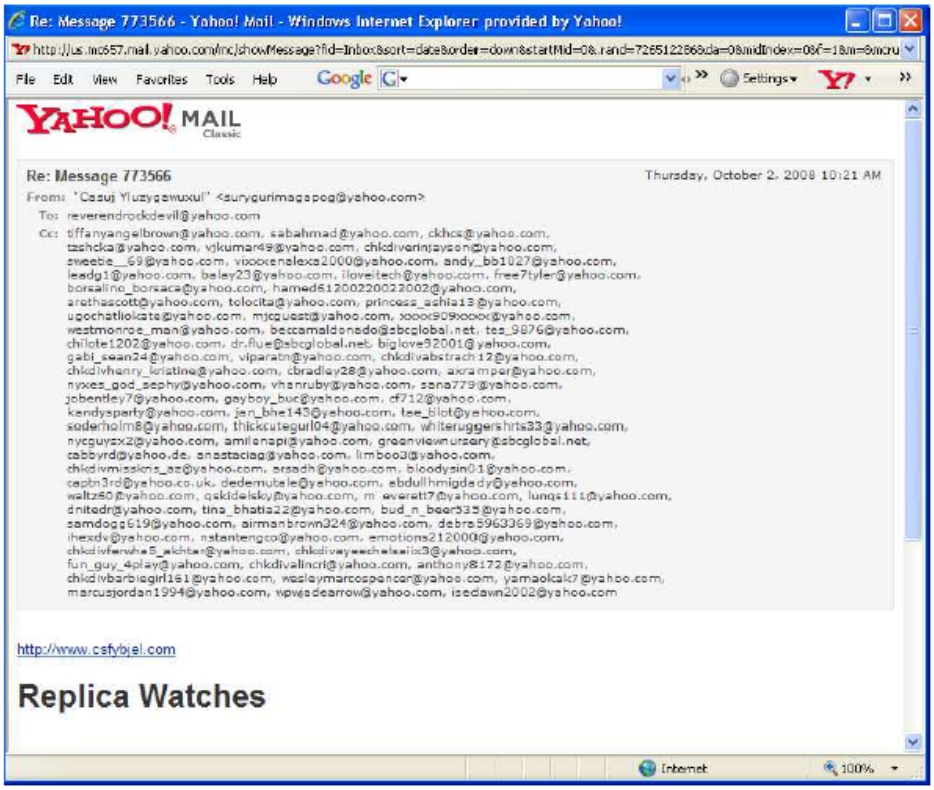
The USER ID such as particular alphabet character, numerical representations can be embedded to conceal data in available e-mail addresses such as *X_mary_23@DOMAIN_ NAME.Extension.*

- And any other legitimate text may also be used.

Such a huge collection of textual e-mail headers can be employed to conceal data. Implementing such bank of e-mail headers is accomplished by collecting the required text. Generally, the texts are initially generated by humans, like any actual (existing) e-mail address, and then the texts can be collected in an application that manages textual e-mail headers and allows fast storage and retrieval of that data, e.g. database. Then, it forms the head-cover from the collection that is updatable which gives it more robustness. by choosing a e-mail headers that are capable of concealing the encoded messages. In addition, if an e-mail address is used for concealing data then available e-mail address (one's that are not yet taken) can be generated in order to employ it by Headstega. It is like creating a new e-mail account(s) and the system indicates the availability of the e-mail address (Google E-mail, 2008; MSN Hotmail; Yahoo Email). Thus, text generated in this manner along with the way it is often used (embedded in the

e-mail headers, e.g. e-mail address in the 'To' or 'Cc' fields) is linguistically legitimate because there is no textual structures to be obeyed. Yet, the reuse of such text, which is a common practice as the use of usual re-communications, e.g. re-e-mailing, further legitimises the text reusability. For instance, if a message will be concealed by using numerical values in e-mail addresses, a set of authenticated e-mail addressees that matches the symbols (bit string) used in the encoded message have to be picked. The way of embedding these picked steganographic carriers (e-mail headers) in the head-cover is handled based on a predetermine protocol by the communicating parties. It is worth noting that justifying the head-cover for sending e-mails is essential. For example, if someone advertising for a particular or general product by e-mails, the scope of market target will be vary. Some other styles may require a higher level of sophistication in order to generate a head-cover that a sender may mix steganographic carriers (encoded e-mail addresses) among other legitimate non-encoded e-mail addresses. This can be accomplished by following a particular sequence, such as odd number, even number, every other 3, etc. or any other way that is a preagreed between sender and receiver. Obviously, the basic configuration of the communication protocol should include how a recipient can only decode the right covers.

Figure 2 Shows the common practice of sending e-mails to a group of people (see online version for colours)



Note: Obviously, the e-mail in this figure does not contain a hidden message and was just an innocent and common practice by people who are interested in marketing and promoting business.

4 Headstega implementation

This section demonstrates the feasibility of Headstega methodology and its distinct capability of achieving the steganographical goal with higher bitrate than contemporary textual steganography approaches. It is worth noting that the focus of this section is balanced on showing how Headstega achieves the steganographical goal rather than making it difficult for an adversary to decode an encoded message. Employing a hard encoding system or cryptosystem to increase the protection of a message is obviously recommended and straightforward using any contemporary encoder or cryptosystem. Similarly, employing compression to boost the bitrate can easily be accomplished by using the contemporary techniques in the literature. This section shows just few examples of possible implementations following the steps outlined in the previous section.

4.1 Headstega configuration

This section first explains how Headstega modules are employed and configured to construct the overall Headstega system used by the communicating parties. In this paper, Headstega encodes a message in a form that suits the camouflaging process. The steganographical code in this Headstega configuration works as follows. Each element (e.g. recipient's e-mail addresses, names, subject fields, etc.) of head-cover conceals particular *m* bits according to the steganographic code defined. In the presented examples, the length of bit string (*m* bits), that can be concealed in a particular element, either four or/and seven bits. The coding is not dependent on the element though. Instead, the first letter of an element in the head-cover contains a steganographic value according to Table 1, which is illustrated later in this section. For example, when an element starts with the letter 'B' it is concluded that the element conceals '0001'. On the other hand, the coding that uses seven bits, the length of bit string (*m* bits), camouflage data in the numerical values in the elements such as binary value of '0000100' can be concealed in the e-mail address like love4u@yahoo.com. 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.

In presented Headstega system, in this paper, head-cover is mainly a textual data of e-mail headers. The camouflage module generates a text-cover (head-cover) may employ internet search engines, such as google.com, E-Mails Generator, free e-mail account providers (Google Email; MSN Hotmail; Yahoo Email) in order to generate head-cover that can conceal data. The selected elements, that are generated to camouflage data, are picked or generated based on either the first letter of the element, the numerical values that match the steganographic code value of an encode message (the bit string of a message) or both. As it will be shown in the examples below, the use of first letters or the use of such numerical values does not impose constraints on the employed implementation. Based on the presented Headstega configuration each element (steganographic carrier) may conceal 4–11 bits. Once the communicating parties are agreed upon the Headstega system configuration, the intended parties are ready to communicate covertly with each other using Headstega. The following demonstrates examples of head-cover.

Table 1 The steganographic code for camouflaging four bits for the elements that are employed in e-mail headers such as e-mail addresses

Index	*Binary*	*Letters*
1	0000	A
2	0001	B
3	0010	C
4	0011	D
5	0100	E
6	0101	F
7	0110	G
8	0111	H
9	1000	I
10	1001	J
11	1010	K
12	1011	L
13	1100	M
14	1101	N
15	1110	O
16	1111	P
17	0000	Q
18	0001	R
19	0010	S
20	0011	T
21	0100	U
22	0101	V
23	0110	W
24	0111	X
25	1000	Y
26	1001	Z

4.2 Headstega example

This section shows how Headstega system can be used to conceal messages. Therefore, the following describes the actual process of encoding a message, concealing the encoded message in the generated text-cover (head-cover) and it demonstrates the samples of head-cover. Note the presented implementation uses the domain 'www.test.xyz' which is a non-existing domain for two reasons. Firstly, it is to avoid breaching a provider's e-mail policy. Secondly, it is to avoid any liability of someone using such e-mails for spamming. Obviously, e-mail addresses from an existing e-mail provider such as free Yahoo Mail, Gmail, MSN Hotmail, etc. can be used.

4.2.1 Sample head-cover

The presented sample in this paper conceals up to 11 bits in the examples of e-mail address. This is accomplished by concealing 4 bits in the e-mails according to the first letter of each one according to steganographic code in Table 1. In addition, 7 bits is concealed in each e-mail addresses by selecting or generating e-mails that contain the required numerical values in order to embed a message. The presented sample is demonstrated in Table 2 and Figure 3.

Table 2 Details encoding of the message '2night@8AM Use my secret key' employing e-mail addresses based on Table 3 along with embedding numerical values

			The plaintext of the message is:'2night@8AM Use my secret key'					
			The plaintext of 1st part of the message is: '2night@8AM'		*The plaintext of 2nd part the message is: 'Use my secret key'*			
Index	*Binary string of the ASCII representation of a message sliced in 4 bits length*	*Decimal*	*1st letter of e-mail address*	*Index of the raw used from Table 3*	*Binary string of the ASCII representation of a message sliced in 7 bits length*	*Decimal value will be embedded in e-mail address*	*Headstega coverusing both the first letters and numerical values in the e-mail addresses*
1	0011	3	D or T	1	0101010	42	dream42@test.xyz
2	0010	2	D or T	2	1011100	92	tarak92@test.xyz
3	0110	6	I or Y	3	1101100	108	yvonne108@test.xyz
4	1110	14	R	4	1010010	82	rob82@test.xyz
5	0110	6	K or A	5	0000011	3	adam3@test.xyz
6	1001	6	L or B	6	0110101	53	lawrence53@test.xyz
7	0110	6	M or C	7	1110010	114	mariya114@test.xyz
8	0111	7	O or E	8	0100000	32	omar32@test.xyz
9	0110	6	O or E	9	0111001	57	edward57@test.xyz
10	1000	8	R or H	10	1011001	89	harry89@test.xyz
11	0111	7	R or H	11	0101100	44	honey44@test.xyz
12	0100	4	P or F	12	0110111	55	paula55@test.xyz
13	0100	4	Q or G	13	0010011	19	qhelp19@test.xyz
14	0000	0	N or D	14	0010101	21	nancy21@test.xyz
15	0011	3	R or H	15	1101000	104	rashama104@test.xyz
16	1000	8	X or N	16	0100000	32	xrob32@test.xyz
17	0100	4	U or K	17	0110101	53	kennedy53@test.xyz

4.3 Bitrates

The aim of this section is to compare the bitrate of contemporary textual steganography approaches to that achieved by Headstega. The bitrate is defined as the size of the hidden message relative to the size of the cover. The average bitrate of the presented Headstega system used in this paper is roughly 3.38% up to 7.67%. It is worth noting that the bitrate differs from one element to another, from one implementation to another, etc. as observed. To put this bitrate figure in perspective, the bitrate of contemporary textual steganography approaches has been investigated. The following reports on the findings, categorising them based on the pursued approaches whereas Table 3 provides a concise summary of these findings.

1 The statistical-based approach, namely mimic functions: an experiment has been conducted using 30 samples generated using Spam Mimic. An average bitrate of 0.90% is observed.

2 Synonym-based approaches:

   i For the NICETEXT scheme, the samples in Chapman and Davida (1997, 2002, 2007) are used to estimate the bitrate, which is found to be approximately 0.29%.

   ii The Winstein's (2008a,b) scheme roughly hides about six bits per sentence, which yields a bitrate of approximately 0.5% based on the sentences listed in the these publications. However, this rate cannot be generalised since not every sentence in the text-cover conceals data. In addition, the size of sentences will affect the bitrate because there are short and long sentences. Nonetheless, the 0.5% figure is assumed given that it is based on the samples developed by the authors.

   iii The capability of the scheme of Murphy and Vogel (2007) again is reported as the number of bits per sentence. Based on the samples provided in their publication, the achievable bitrate is roughly 0.30% per sentence.

   iv Nakagawa et al. (2001) have provided two samples for their scheme. The samples achieve bitrate of 0.06% and 0.12%, respectively. However, it has been noted that when tried in a real application, only a bitrate of 0.034% could be reached.

3 Noise-based approaches:

   - The bitrate for the translation-based scheme reported in Stutsman et al. (2006) is roughly 0.33%.
   - Based on the examples in Topkara et al. (2007), the confusing scheme approximately achieves a bitrate of 0.35%.
   - The technique of the SMS-based approach (Shirali-Shahreza et al., 2007) is said to be capable of hiding few bits in a file of several kilobytes, which yields an extremely low bitrate.

Table 3 The bitrate of contemporary textual steganography approaches

Approach	*Bitrate (%)*	*Comment*
Mimic functions (Wayner, 1992, 2002)	0.90	Based on 30 samples generated at www.spamimc.com
NICETEXT (Chapman and Davida, 1997, 2002, 2007)	0.29	Based on the samples in the cited papers
Winstein (2008a,b)	0.5	Based on the samples in the cited papers, and also confirmed in Murphy and Vogel (2007)
Murphy and Vogel (2007)	0.30	Average per sentence (as reported in Murphy and Vogel (2007))
Nakagawa et al. (2001)	0.12	As reported in Nakagawa et al. (2001) bitrate achieved in real application is only 0.034%
Translation-based (Stutsman et al., 2006)	0.33	Noted by the authors in the cited papers
Confusing (Topkara et al., 2007)	0.35	Based on the samples in the cited papers

Comparing the achieved bitrate by Headstega which is roughly 3.38% up to 7.67% vs. the bitrate achieved by the contemporary textual approaches in Table 3, it is obvious that Headstega achieves superior bitrate than all comparable approaches, making it a very effective steganography approach.

5 Steganalysis validation

The aim of this section is to show the resilience of Headstega to possible attacks. Again the success of steganography is qualified with its ability for avoiding an adversary's suspicion of the presence of a hidden message. It is assumed that an adversary will perform all possible investigations. In addition, the adversary is also aware of Headstega, as a public methodology, but he does not know the Headstega configuration that the sender and recipient employ for their covert communication.

5.1 Traffic attack

One of the possible attacks an adversary may pursue is to inspect the communications traffic of images, graphs, audio files, etc. in order to detect the existence of covert communications if occurred. For example, the intelligence community has a number of tools at their disposal for analysing traffic on the internet, tracking access to websites, monitoring checked out literature from public libraries, etc. The main goal of a traffic attack is to detect unusual or questionable association between a sender and recipient. Traffic analysis intuitively can identify who communicates with whom. The relationship between the communicating parties will be then qualified based on the contents of the message. Traffic attacks can be a threat for most contemporary steganographic techniques regardless of the steganographic cover types (e.g. image, graph, audio file, text, etc.) used. In the context of Headstega, the profile of users and e-mail subjects are checked rather than its validity and the consistency of its textual e-mail body. If someone e-mails and receives e-mails without a justifiable reason, e.g. e-mails for marketing military airplanes, obviously suspicion can be raised and further investigation may be warranted.

The additional investigations will involve a thorough analysis of a steganographic cover, as detailed in the next sections.

Traffic analysis is deemed ineffective with Headstega. Headstega camouflages the transmittal of a hidden message (head-cover) to appear legitimate and thus suspicion is averted. Basically, Headstega is based on Nostega paradigm (Desoky, 2008a, 2009b) which implies that Headstega by default ensures that the involved parties establish a covert channel by having a justifiable relationship with each other rendering the communications traffic innocent and to look like any ordinary communications. Analysing the traffic between them will not reveal any questionable association and will not trigger any further investigation. In addition, Headstega imposes on the communicating parties to use innocent domains, e.g. contexts, martial, etc. that retain high demand by a wide variety of people. Such domains create a high volume of traffic that makes it impractical for an adversary to investigate all traffics. The voluminous traffic allows the communicating parties to establish a covert channel in order to transmit a head-cover without drawing attention, rendering Headstega an attractive steganographical methodology to be used. Finally, it is noted that if further investigations on a head-cover are triggered by traffic analysis, they would not be successful, as elaborated next. In Headstega, differentiating between a head-cover that contains a hidden message and another peer textual e-mail headers without a hidden message is infeasible.

5.2 Contrast and comparison attacks

One of the intuitive sources of noise that may alert an adversary is the presence of contradictions in a head-cover. Examples of these contradictions include finding suspicious repetition of an element. Also, if a head-cover contains errors, it is not expected to be numerous. Such contradictions may raise suspicion about the existence of a hidden message, especially when they are present in the same document. Automating the generation of a head-cover through the use of data banks makes the cover very resilient to this type of attacks. As demonstrated in Section 4, the use of a tool, such as yahoo mail, gmail, Hotmail, google.com, etc. allows the appropriate textual cover generation that avoids suspicious. Meanwhile, noise in the context of comparison attacks reflects alteration of authenticated or previously used documents. The goal of the adversary is to find any incorrect and inconsistent data that may imply the manipulation of contents of a head-cover in order to embed a hidden message. However, since reusing e-mail headers, e.g. e-mail addresses, are common practices, comparison attacks are deemed ineffective.

5.3 Linguistics attacks

Linguistics examination distinguishes 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 feature that can help in detecting or suspecting the existence of a hidden message. These examinations are used to determine whether or not the text that is under attack is abnormal. Generally, the text used in e-mail headers is not sophisticated document and it is easy of such scheme to retain the textual normality of head-cover. In addition, the produced set of textual elements meets the expected properties of a normal human

language because it is initially generated by human and any alteration is done is more of cosmetic, e.g. changing the order of elements. Yet, there is no linguistic structure to be obeyed in e-mail headers and thus does not generate any noise (linguistic flaws). As a result, the generated cover as demonstrated in the implementation section is normal text. Furthermore, if there are errors in the cover generator engine, it should not be a concern for two reasons. Firstly, it applies to all other textual e-mail headers that contain no hidden messages. Secondly, nothing is concealed in errors. In addition, an engine error of such cover generator is most likely fixable. Therefore, Headstega is capable of passing any linguistic attack by both human and machine examinations.

On the other hand, a statistical attack refers to tracking the profile of the used text. A statistical signature (profile) of a text refers to the frequency of words and characters used. An adversary may use the statistical profile of a particular topic of documents that contains no hidden message and compare it to a statistical profile of the suspected head-cover to detect any differences. An alteration in the statistical signature of a particular document may be a possible way of detecting a noise that an adversary would watch for. Unlike image steganography, tracking statistical signatures is an ineffective means for attacking textual steganography (Desoky, 2009b; Grothoff et al., 2005a,b; Stutsman et al., 2006). Nonetheless, Headstega is resistant to statistical attacks because it is simply opt to use legitimate text that is generated naturally by human. In addition, the generated textual cover (head-cover) by Headstega keeps the same profile of its other peer documents that contains no hidden message. Basically, most alterations introduced by Headstega are non-linguistic and do not produce any flaws (noise), as demonstrated in the implementation section, deeming statistical attacks on head-cover ineffective.

6 Conclusion

The high demand for using e-mails by a wide variety of people allows the communicating parties to establish a covert channel to transmit hidden messages (head-cover) rendering e-mails an attractive steganographic carriers. Such features motivate the development of the e-mail-*head*ers-based *stega*nography methodology (Headstega). Headstega conceals data only in textual e-mail headers. Headstega neither hides data in a noise (errors) nor produces noise. Instead, it camouflages data in legitimate elements of only textual e-mail headers by mainly exploiting elements, such as recipient's e-mail addresses, names, subject, abbreviations, etc. in order to embed data without generating any suspicious pattern. The presented implementation achieves bitrate up to 7.67%. Such bitrate is superior to contemporary text steganography approaches found in the literature, confirming the effectiveness of Headstega methodology. The steganalysis validation has shown Headstega methodology is capable of achieving the steganographic goal.

References

Anderson, R.J., Needham, R. and Shamir, A. (1998) 'The steganographic file system', *Proceedings of the Second International Workshop on Information Hiding*, Vol. 1525 of *Lecture Notes in Computer Science*, Springer, pp.73–82.

Ansari, R., Malik, H. and Khokhar, A. (2004) 'Data-hiding in audio using frequency-selective phase alteration', *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '04)*, Vol. 5, Nos. 17–21, pp.389–392.

Atallah, M.J., Raskin, V., Crogan, M., Hempelmann, C., Kerschbaum, F., Mohamed, D. and Naik, S. (2001) 'Natural language watermarking: design, analysis, and a proof-of-concept implementation', in I.S. Moskowitz (ed.), *Information Hiding: Fourth International Workshop, Lecture Notes in Computer Science*, Vol. 2137, Springer, pp.185–199.

Atallah, M.J., Raskin, V., Hempelmann, C.F., Topkara, M., Sion, R., Topkara, U. and Triezenberg, K.E. (2002) 'Natural language watermarking and tamperproofing', in F.A.P. Petitcolas (ed.), *Information Hiding: Fifth International Workshop, Vol. 2578 of Lecture Notes in Computer Science*, Springer, pp.196–212.

Bennett, K. (2004) 'Linguistic steganography: survey, analysis, and robustness concerns for hiding information in text', Technical Report CERIAS Tech Report 2004–13, Purdue University.

Bender, W., et al. (1996) 'Techniques for data hiding', *IBM Systems Journal,* Vol. 35, Nos. 3 and 4, pp.313–336.

Bergmair, R. (2004) 'Towards linguistic steganography: a systematic investigation of approaches, systems, and issues', Final Year Project, The University of Derby, April.

Bergmair, R. and Katzenbeisser, S. (2004) 'Towards human interactive proofs in the text-domain', *Proceedings of the 7th Information Security Conference (ISC'04), Springer Lecture Notes in Computer Science*, September.

Bolshakov, I.A. (2004) 'A method of linguistic steganography based on collocationally-verified synonymy', in J.J. Fridrich (Ed.), *Information Hiding: 6th International Workshop, Volume 3200 of Lecture Notes in Computer Science*, Springer, May, pp.180–191.

Bolshakov, I.A. and Gelbukh, A. (2004) 'Synonymous paraphrasing using wordnet and internet', in F. Meziane and E. Elisabeth Metais (Eds.), *Natural Language Processing and Information Systems: 9th International Conference on Applications of Natural Language to Information Systems, NLDB 2004*, *Volume 3136 of Lecture Notes in Computer Science*, Springer, June, pp.312–323.

Calvo, H. and Bolshakov, I.A. (2004) 'Using selectional preferences for extending a synonymous paraphrasing method in steganography', in J.H. Sossa Azuela (Ed.), *Avances en Ciencias de la Computacion e Ingenieria de Computo – CIC'2004: XIII Congreso Internacional de Computacion*, October, pp.231–242.

Chand, V. and Orgun, C.O. (2006) 'Exploiting linguistic features in lexical steganography: design and proof-of-concept implementation', *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, Vol. 6, p.126b. IEEE, January.

Chapman, M. and Davida, G.I. (1997) 'Hiding the hidden: a software system for concealing ciphertext as innocuous text', *Proceedings of the International Conference on Information and Communications Security, Vol. 1334 of Lecture Notes in Computer Science*, Beijing, PR China: Springer, November, pp.335–345.

Chapman, M. and Davida, G.I. (2002) 'Plausible deniability using automated linguistic steganography', in Davida, G. and Frankel, Y. (Eds.), *International Conference on Infrastructure Security (InfraSec '02)*, Vol. 2437 *of Lecture Notes in Computer Science*, Springer, pp.276–287.

Chapman, M. and Davida, G.I. (2007) 'Nicetext system official home page', Available at: http://www.nicetext.com, Accessed on 3 August 2007.

Chapman, M., et al. (2001) 'A practical and effective approach to large-scale automated linguistic steganography', *Proceedings of the Information Security Conference (ISC '01)*, *Volume 2200 of Lecture Notes in Computer Science*, Malaga, Spain: Springer, pp.156–165.

Cvejic, N. and Seppanen, T. (2004a) 'Increasing robustness of LSB audio steganography using a novel embedding method', *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, Las Vegas, Nevada, pp.533–537.

Cvejic, N. and Seppanen, T. (2004b) 'Reduced distortion bit-modification for LSB audio steganography', *Proceedings of the 7th International Conference on Signal Processing (ICSP 04)*, Vol. 3, Beijing, China, pp.2318–232.

Desoky, A. (2008a) 'Nostega: a novel noiseless steganography paradigm', *Journal of Digital Forensic Practice*, Vol. 2, No. 3, pp.132–139.

Desoky, A. (2008b) 'Graphstega: graph steganography methodology', *Journal of Digital Forensic Practice*, Vol. 2, No. 1, pp.27–36.

Desoky, A. (2009a) 'Listega: list-based steganography methodology', *Int. J. Information Security*.

Desoky, A. (2009b) 'Nostega: a novel noiseless steganography paradigm', PhD Dissertation, University of Maryland, Baltimore County.

Desoky, A. (2009c) 'Notestega: notes-based steganography methodology', *Information Security Journal: A Global Perspective*, Vol. 18, No. 4, pp.178–193.

Desoky, A. (2010) 'Comprehensive linguistic steganography survey', *Int. J. Information and Computer Security*, Vol. 4, No. 2, pp.164–197.

Desoky, A. (in press) 'Matlist: mature linguistic steganography methodology', *Journal of Security and Communication Networks*.

Desoky, A. (in process) *Noiseless Steganography: The Key to Covert Communications*. Information Security Publisher/Taylor & Francis Group.

Desoky, A. and Younis, M. (2006) 'PSM: public steganography methodology', Technical Report TR-CS-06-07, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County.

Desoky, A. and Younis, M. (in press) 'Chestega: chess steganography methodology', *Journal of Security and Communication Networks*.

Desoky A., et al. (2008) 'Auto-summarization-based steganography', *Proceedings of the 5th IEEE International Conference on Innovations in Information Technology*, Al-Ain, UAE.

Email Generators. Available at: http://www.icontact.com. Accessed on 26 September 2008.

Google Email. Available at: https://www.google.com/ accounts/ServiceLogin?service= mail&passive=true&rm=false&continue=http%3A%2F%2Fmail.google.com%2Fmail%2F%3Fhl%3Den%26ui%3Dhtml%26zy%3Dl&bsv=1k96igf4806cy&ltmpl=default&ltmplcache=2&hl=en. Accessed on 26 September 2008.

Google Internet Search Engine. Available at: http://www.google.com. Accessed on 26 September 2008.

Grothoff, C., et al. (2005a) 'Translation-based steganography', Technical Report CSD TR# 05-009, Purdue University (CERIAS Tech Report 2005-39).

Grothoff, C., et al. (2005b) 'Translation-based steganography', *Proceedings of Information Hiding Workshop (IH 2005)*, Barcelona, Spain: Springer-Verlag, June, pp.213–233.

Gruhl, D., Lu, A. and Bender, W. (1996) 'Echo hiding', *Proceedings of First International Workshop on Information Hiding*, *Lecture Notes in Computer Science*, Vol. 1174, Springer, Cambridge, UK, pp.295–316.

Handel, T.G. and Sandford, M.T. (1996) 'Data hiding in the OSI network model', *Information Hiding: First International Workshop, Proceedings* of *Lecture Notes in Computer Science*, Springer, Vol. 1174, pp.23–38.

Kahn, D. (1996) *The Codebreakers: The Story of Secret Writing* (revised ed.). Scribner.

Kirovski, D. and Malvar, H. (2001) 'Spread-spectrum audio watermarking: requirements, applications, and limitations', *Proceedings of the 4th IEEE Workshop on Multimedia Signal Processing*, Cannes, France, pp.219–224.

Live Search: Internet Search Engine. Available at: http://www.live.com. Accessed on 26 September 2008.

Martin, A., Sapiro, G. and Seroussi, G. (2005) 'Is image steganography natural?', *IEEE Transactions on Image Processing*, Vol. 14, No. 12, pp.2040–2050.

MSN Hotmail. Available at: https://login.live.com/ login.srf?wa=wsignin1.0&rpsnv=10&ct=1222960929&rver=4.5.2130.0&wp=SAPI&wreply=https:%2F%2Faccount.live.com%2Fsummarypage.aspx%3Fmkt%3DEN-US&lc= 1033&id=38936.Accessed on 26 September 2008.

Murphy, B. and Vogel, C. (2007) 'The syntax of concealment: reliable methods for plain text information hiding', *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*.

Nakagawa, H., Sampei, K., Matsumoto, T., Kawaguchi, S., Makino, K. and Murase, I. (2001) 'Text information hiding with preserved meaning – a case for Japanese documents', *IPSJ Transaction*, Vol. 42, No. 9, pp.2339–2350. Originally published in Japanese. A similar paper by the first author in English Available at: http://www.r.dl.itc.u-tokyo.ac.jp/nakagawa/academic-res/finpri02.pdf. Accessed on 4 June 2008.

Niimi, M., Minewaki, S., Noda, H. and Kawaguchi, E. (2003) 'A framework of text-based steganography using SD-form semantics model', *IPSJ Journal*, Vol. 44, No. 8, Available at: http://www.know.comp.kyutech.ac.jp/STEG03/STEG03-PAPERS/papers/12-Niimi.pdf, Accessed on 3 June 2008.

Petitcolas, F.A.P. (1999) 'Information hiding – a survey', in R.J. Anderson and M.G. Kuhn (Eds.), *Proceedings of the IEEE*, July, Vol. 87, No. 7, pp.1062–1078.

ScramDisk. *Free Hard Drive Encryption for Windows 95 & 98*, Available at: http://www.scramdisk.clara.net, Accessed on 3 August 2008.

Shirali-Shahreza, M.H. and Shirali-Shahreza, M. (2006) 'A new approach to Persian/Arabic text steganography', *Proceedings of 5th IEEE/ACIS International Conference on Computer and Information Science (ICIS-COMSAR 2006)*, Hawaii: Honolulu, July 2006, pp.310–315.

Shirali-Shahreza, M., et al. (2007) 'Text steganography in SMS', *International Conference on Convergence Information Technology*, Nos. 21–23, pp.2260–2265.

Spam Mimic. Available at: http://www.spammimic.com. Accessed on 31 July 2007.

Stutsman, R., et al. (2006) 'Lost in just the translation', *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06)*, Dijon, France, April.

Topkara, M., Topkara, U. and Atallah, M.J. (2007) 'Information hiding through errors: a confusing approach', *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*.

Topkara, U., Topkara, M. and Atallah, M.J. (2006) 'The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions', *MM&Sec '06: Proceeding of the 8th Workshop on Multimedia and Security*, New York, NY, USA: ACM Press, pp.164–174.

Wayner, P. (1992) 'Mimic functions', *Cryptologia*, Vol. XVI/3, pp.193–214.

Wayner, P. (2002) *Disappearing Cryptography* (2nd ed.). Morgan Kaufmann, pp.81–128.

Winstein, K. (2008a) 'Lexical steganography through adaptive modulation of the word choice hash', *January 1999. Secondary Education at the Illinois Mathematics and Science Academy*. Available at: http://alumni.imsa.edu/~keithw/tlex/lsteg.ps, Accessed on 15 April 2008.

Winstein, K. (2008b) *Lexical Steganography*. Available at: http://alumni.imsa.edu/~keithw/tlex, Accessed on 3 August 2008.

Yahoo Email. Available at: https://login.yahoo.com/config/login_verify2?&.src=ym. Accessed on 26 September 2008.

Yahoo Internet Search Engine. Available at: http://search.yahoo.com/web?fr=fptb-msgr. Accessed on 26 September 2008.