Sumstega: summarisation-based steganography methodology

Abdelrahman Desoky

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

Abstract: The demand for reading while no one has time to read everything has fuelled the necessity for automatic summarisation systems in business, science, World Wide Web, education, news, etc. Thus, the popular use of summaries by a wide variety of people creates a high volume of traffic for accessing and generating summaries. Such huge traffic makes an adversary's job impractical to investigate all of them and allows communicating parties to establish a secure covert channel to transmit steganographic covers. This renders summaries an attractive steganographic carrier. Therefore, summarisation-based steganography methodology (Sumstega), presented in this paper, takes advantage of the automatic summarisation techniques to generate summary-cover. Sumstega neither hides data in a noise nor produces noise. Instead, Sumstega manipulates the parameters and factors of automatic summarisation techniques in order to embed data without noise, which retains adequate rooms for concealing data. The validation demonstrates the capability of achieving the steganographic goal.

Keywords: steganography; linguistic steganography; information hiding; information security; secure communications; covert communications.

Reference to this paper should be made as follows: Desoky, A. (2011) 'Sumstega: summarisation-based steganography methodology', *Int. J. Information and Computer Security*, Vol. 4, No. 3, pp.234–263.

Biographical notes: Abdelrahman Desoky is a Scientist with over 18 years experience in the computer field. He is the author of security book entitled *Noiseless Stenography: The Key of Covert Communications*. Further, he is an author and main contributor of numerous stenography papers that are published in prominent journals. He received his PhD from the University of Maryland and his MSc from the George Washington University; both degrees are in Computer Engineering. His earlier studies included a Professional Postgraduate Studies Certificate (PGS) in Computer Science from the Cairo University and a Bachelor of Science (BSc) in Agricultural and Cooperative Sciences from the Higher Institute for Agricultural and Co-operation.

1 Introduction

Steganography is the science and art of camouflaging the presence of covert communications. The origin of steganography is traced back to early civilisations

[Desoky, 2010c, in process (a)]. The ancient Egyptians communicated covertly using the hieroglyphic language, a series of symbols representing a message. 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. Fundamentally, the steganographic goal is not to hinder the adversary from decoding a hidden message, but to prevent an adversary from suspecting the existence of covert communications (Desoky, 2008a, 2009a, 2010c, in process). 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, 2008a, 2009a, 2010c, in process). Contemporary approaches are often classified based on the steganographic cover type into image, audio, graph (Desoky and Younis, 2006, 2008; Desoky, 2009a, in process), or text. When linguistics is employed for hiding data and generating the steganographic cover, an approach is usually categorised as linguistic steganography to distinguish it from non-linguistic techniques, e.g., image, audio, etc. Linguistic steganography has become more favourable in recent years since the size of non-linguistic-covers is relatively large and is burdening the traffic of covert communications (Desoky, 2008a, 2009a, 2010c, in process).

Most of the published steganography approaches hide data as noise in a cover that is assumed to look innocent. For example, the encoded message can be embedded as an alteration of a digital image or an audio file without noticeable degradation (Martin et al., 2005). Another example is hiding a message in a text-cover by modifying the format and style of an existing text (Desoky, 2010c, in process).

However, such alteration of authenticated covers can raise suspicion and the message is detectable regardless of whether or not a plaintext is revelled (Desoky, 2009a, 2010c, in process).

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 hardrive, etc. (Anderson et al., 1998; ScramDisk, 2008), 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, 2010c, in process).

On the other hand, a similar argument is made in the literature about linguistic 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, (Grothoff et al., 2005a, 2005b; Stutsman et al., 2006), confusing approach (Topkara et al., 2007), and abbreviation-based (Shirali-Shahreza et al., 2007). The vulnerability and concerns of these linguistic approaches, as explained in Section 2, can be summarised as follows. First, the linguistic-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. Second, the content of the cover may be meaningless and semantically incoherent, and thus may draw suspicion. Third, 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-linguistics approaches as well. Fourth, 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. Fifth, while these approaches may fool a computer examination, they often fail to pass human inspections. A successful linguistic steganography approach must be capable of passing both computer and human examinations. These concerns have motivated the development of the summarisation-based steganography methodology (Sumstega), introduced in this paper.

The necessity of using summaries in business, science, education, news, World Wide Web, etc., is because people do not have enough time for reading long documents. This necessity allows the communicating parties to establish an innocent covert channel to transmit a hidden message rendering an adversary's job impractical to investigate all of them. The automatic summarisation's aim is to represent the core contents of a long document(s) in a significantly smaller document(s) than its original input (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000). The summarisation systems employs the parameters and factors of automatic summarisation techniques (PFAST) such as the weight (e.g., weight of frequency, location, semantic), paraphrasing, truncation, reordering, semantic and information equivalency, etc., in order to generate summaries. Sumstega exploits summarisation techniques and its PFAST to achieve the steganographic goal by concealing data in a summary-cover that looks legitimate and then transmits it covertly among other legitimate summary's traffics. For example, Sumstega may generate possible variations of legitimate summaries (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007), and then Sumstega manipulates these possible variations of legitimate summaries to naturally embed data in a summary-cover. Virtually, it forms the elements (e.g., sentences, words, etc.) of a summary-cover from possible different of legitimate summaries for the same document to conceal data in such a way that a summary-cover can fool both human and machine examinations. Consequently, a legitimate sender will covertly transmit the summary-cover through a covert channel that is summary traffics-based.

The main advantages of Sumstega are as follows. First, the tremendous amount of summary in electronic and non-electronic format makes it impossible for an adversary to investigate all of them. This makes it extremely favourable as a steganographic cover in covert communications. Second, Sumstega is resilient against contemporary attacks including an attack by an adversary who familiars with Sumstega (Sumstega is a public methodology). Third, Sumstega does not apply a particular pattern (noise) that an adversary may look for. Fourth, the concealment process of Sumstega has no effect on the linguistics of the generated cover (summary-cover). Therefore, a summary-cover is linguistically legitimate comparing to its peer summaries and is thus capable of passing both computer and human examinations. Fifth, Sumstega can be applied to all languages. Sixth, there is adequate room for concealing data in summaries. The implementation and steganalysis validation demonstrate that Sumstega methodology is capable of achieving the steganographical goal.

The remainder of this paper is organised as follows. Section 2 discusses the related work. Section 3 is an overview of automatic summarisation techniques that can be employed by Sumstega. Section 4 introduces Sumstega methodology in detail. Section 5 demonstrates the implementation of Sumstega. Section 6 demonstrates the steganalysis validation. Finally, Section 7 concludes the paper.

2 Related work

The output of both linguistic steganographical schemes and text summarisation 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 text summarisation systems is to represent the essence contents of long document(s) in a significant smaller document(s) than its original input. Linguistically, the generated summary by the text summarisation systems may not appear perfect but it is legible. Linguistically, the generated text (summary) by the presented scheme, namely Sumstega, retains superior quality than the output of contemporary approaches, as demonstrated in this paper.

The following subsections present a brief review of prior work on linguistic and non-linguistic steganography and automatic summarisation.

2.1 Linguistic steganography

Linguistic steganography approaches conceal data in a linguistic-based textual cover. Linguistic steganography approaches can be categorised as follows.

2.1.1 Series of characters and words

During World War I, the Germans communicated covertly using a series of characters and words known as null-cipher (Desoky, in process; 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 that is not legitimate. Applying a brute force attack may reveal the entire message.

2.1.2 Statistical-based

Wayner 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 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. The output from regular mimic functions is gibberish rendering it extremely suspicious (Wayner, 1992, 2002). 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.

2.1.3 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

(Chapman and Davida, 1997, 2002, 2007; Chapman et al., 2001). 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 in the last decade, for more information refer to Desoky (2010c, in process).

2.1.4 Noise-based

Grothoff et al. have introduced the translation-based steganographic scheme (Grothoff et al., 2005a, 2005b; Stutsman et al., 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. states 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, 2005b, Stutsman et al., 2006). On the contrary, Sumstega can be applied to all known languages without any exceptions while the generated summary-cover is linguistically legitimate.

Another noise-based approach has been proposed by Topkara et al. that employs typos and ungrammatical abbreviations in a text, e.g., emails, 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, Sumstega neither employs errors nor uses noisy text to conceal data.

2.1.5 Nostega-based

Recently, the new paradigm in steganography research, namely noiseless steganography paradigm (Nostega) has been introduced, in which the message is hidden in the cover as data rather than noise (Desoky, 2008a, 2009a, 2010c, in process). A number of methodologies have been developed based on the Nostega paradigm. One of these methodologies is the list-based steganography methodology (Listega) (Desoky, 2009b). Listega manipulates itemised data to conceal messages in a form of textual list. The second linguistic steganography methodology, notes-based steganography methodology (Notestega) that takes advantage of the recent advances in automatic notetaking techniques to generate a text-cover (Desoky, 2009c, 2010c, in process). Notestega pursues the variations among both human notes and the outputs of automatic notetaking techniques to conceal data. The third linguistic steganography methodology, mature linguistic steganography methodology (Matlist) (Desoky, 2010a, 2010c, in process) 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. The fourth linguistic steganography methodology, Normal Linguistic Steganography Methodology (NORMALS) (Desoky, 2010b) employs Natural Language Generation (NLG) techniques to generate noiseless (flawless) and legitimate text-cover by manipulating the inputs' parameters of NLG system in order to camouflage data in the generated text. Unlike Matlist, NORMALS is capable of handling non-random series domains. The fifth linguistic steganography methodology is the email-headers-based steganography methodology (Headstega) (Desoky, in press). The frequent exchange of emails is widely popular and generates a high volume of traffic that allows communicating parties to establish a covert channel without a suspicious pattern rendering emails an attractive steganographic carrier to transmit hidden messages. This was the motive of developing email-headers-based steganography methodology (Headstega) (Hobson et al., 2007). It encodes a message then assign it to steganographic carriers, e.g., recipient's e-mail addresses, names, subject fields, etc., in order to camouflage data.

It is worth noting that the presented Sumstega methodology in this paper follows this new paradigm (Nostega) by exploiting automatic summarisation techniques to camouflage data without generating any suspicious pattern.

2.2 Non-linguistic steganography

Non-linguistic steganography approaches can be categorised based on its file type such as text, image, audio, and graph. Textual steganography, which is based on non-linguistic techniques, hides data by textual format manipulation (TFM) process (Desoky, 2010c). 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 versus 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 (Desoky, 2010c).

On the other hand, 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, 2004b), spread spectrum coding (Bender et al., 1996; Kirovski and Malvar, 2001), phase coding (Bender et al., 1996), and echo hiding (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 (Martin et al., 2005). 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 a major threat to the covert communication. Basically, an adversary can detect the presence of a hidden message by comparing a particular image-cover 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, 2008). 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 hardrive used. TCP/IP packets used to transport information across the internet have unused space in the packet headers (Handel et al., 1996). The TCP packet header has six unused (reserved) bits and the IP packet header has two reserved bits. There are tremendous packets are transmitted over the internet can convey and transmit a secret data. However, these techniques are vulnerable to distortion attacks (Desoky, 2010c).

Recently, a graph steganography (Graphstega) methodology has been developed (Desoky, 2008b). 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 (Desoky and Younis, 2009) exploits popular games, like 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 intrinsically in the cover as noiseless data rather than noise.

2.3 Automatic summarisation

Automatic summarisation is the scientific art of representing the essence of a long document(s) in a significantly smaller document(s) than it's original by employing computer programs. The field is traced back to the 1950s (Luhn, 1958), and in the recent years has enjoyed significant progress and is still promising more in the Mani and Maybury (1999), Mani (2001), Marcu (2000) and Jones (2007). Automatic summarisation systems employ a procedure that may be based on one or more of the following: statistical process, knowledge base, artificial intelligence, computational linguistics, and other related techniques to achieve its Mani and Maybury (1999), Mani (2001), Marcu (2000) and Jones (2007). Some examples of automatic summarisation systems are autosummarise (Microsoft Word)[2], SweSum (Hassel and Dalianis), Inxight summariser (Inxight Software Incorporation, 2000), IBM intelligent miner (IBM intelligent miner, 1999), DimSum (Mani and Maybury, 1999; SRA Corporation), etc. Automatic summarisation approaches may categorise into three types: high level, low level, and hybrid approaches (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007).

High level approaches are also referred to as shallow approaches (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007), depending mainly on extraction approaches and reordering techniques while they attempt to represent the extracted essence in as good a shape as possible. The majority of these approaches produce summary that is entirely a subset of its original. These approaches employ techniques such as frequency and location weight of sentences, words, etc. To illustrate, the summary is as if a set of important sentences is highlighted, copied, and then pasted in a desirable order to form a summary. These approaches from the point view of implementation are desirable because it is significantly easier and low-cost than low-level approaches. Low-level approaches also are referred as deep approaches (Mani and

Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007), which the need of knowledge base and other related techniques, such as artificial intelligence and NLG, are essential to generate an abstract. Therefore, these approaches are significantly sophisticated to implement which makes the cost more than the cost of high-level approaches. Low-level approaches employ techniques such as extraction, paraphrasing rule, reordering, semantic equivalency, information equivalency etc., to generate summaries. Hybrid approaches, which produce a compaction-based summary, are useful for handling multi-document(s) input. Yet, hybrid approaches may use some reordering and discourse techniques for refining an output (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007).

Note that Section 3 demonstrates some of summarisation techniques that can be used by Sumstega methodology to conceal data.

3 Sumstega carriers

The aim of this section is to explore some examples of automatic summarisation techniques to demonstrate possible steganographic summarisation carriers (SSC) that are capable of concealing data while retaining a summary-cover to be plausible, ordinary, and legitimate. It is imperative to study automatic summarisation techniques to explore these plausible SSC before implementing Sumstega scheme. Investigating the manipulation of PFAST in order to generate plausible SSC, which are, practically, all possible different legitimate summaries for the same document. It is well-known that summarisation systems naturally produce different legitimate summaries for the same document (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007). Examples of PFAST may be the weight (e.g., weight of frequency, location, semantic), paraphrasing, truncation, reordering, semantic and information equivalency, etc. Sumstega can then be tuned to exploit the PFAST in order to generate adequate SSC that can camouflage message without violating the pattern of a summary. Virtually, Sumstega embeds data by substituting a set of elements (e.g., sentences, words) of a particular summary with other legitimate elements from peer summaries in such a way that the summary-cover looks like any other legitimate summary. Rather than from the implementation of the automatic summariser point view – because it is out of scope of this paper – the next subsections are from a steganographical point view that can be used by Sumstega methodology to conceal data. Note that all of the following examples are confirmed by the experimental results and observations of both the literature of automatic summarisation field (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007) and Sumstega experimental research work, as shown in this paper.

3.1 Extraction

Mainly, extraction techniques (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007; Leite et al., 2007; Okazaki et al., 2003; Gonenc and Cicekli, 2007; Liang et al., 2007; Yu et al., 2007; Nomoto, 2007) are based on the sentence level to produce summary that is entirely a subset of its original document(s). To illustrate, the summary is as if a set of important sentences is highlighted, copied, and then pasted in a desirable order to form a summary. Different implementations of the same extraction techniques can generate variations of summary (different alterations). Similarly, different extraction techniques can also generate variations of summary. Extraction techniques may use the

weight of frequency or location of sentence, word, etc., to generate a summary. Obviously, different elements (e.g., words, sentences) may have same or similar weight and when summariser needs to select only one element out of these different elements then selecting any one of them (these different elements) can be legitimate. Thus, Sumstega can select legitimate elements that have the required steganographic code (encoded message) to generate a plausible summary-cover. To emphasise, two automatic summarisers can extract different sentences while they summarise the same document(s). For instance, when requesting from autosummarise (Microsoft Word) and automatic text summariser (LTRC, IIIT) to summarise a document(s) from the news (Time Magazine, 2007) in only one sentence the output of both summarisers was different. autosummarise (Microsoft Word) extracted this sentence as shown in Sample 1.

Sample 1: Illustrates the output of autosummarise (Microsoft Word). AutoSummarise is an extraction-based summariser and some techniques such as superfluous terms, sentence truncation, text compaction, deletion macro-rule, and construction macro-rule may be involved in the extraction procedure.

Sample 1

> *Police, mistaking de Menezes for Osman, trailed him into Stockwell tube station and down the escalator onto a platform.*

However, automatic text summariser (LTRC, IIIT) extracted different sentence as shown in Sample 2.

Sample 2: Illustrates the output of automatic text summariser (LTRC, IIIT), which is different from Sample 1. Automatic text summariser (LTRC, IIIT) is extraction-based summariser, too, and some techniques such as superfluous terms, sentence truncation, text compaction, deletion macro-rule, and construction macro-rule may be involved in the extraction procedure.

Sample 2

> *London was on high alert on the morning that police surveillance teams stationed outside an apartment block in South London spotted de Menezes leaving his building on his way to work.*

From a point view of automatic summarisation techniques both autosummarise (Microsoft Word) and automatic text summariser (LTRC, IIIT) are generally based on the same technique, which is extraction, but they are differently implemented. Note that these are just examples and to show the feasibility that Sumstega scheme can generate numerous different paths of legitimate virtual summaries to generate summary-cover.

3.2 Abstraction

Summaries that are generated by abstraction techniques have different legitimate elements (e.g., words, sentences, partial sentences, etc.) from its original document(s) (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007; Cremmins, 1996;

Nomoto, 2007). Steganographically, such elements along with others can obviously play an essential role for embedding a message in the generated legitimate summary by Sumstega methodology. Abstraction techniques are most likely complemented by other summarisation techniques to generate summaries such as:

- extraction
- paraphrasing rule
- lexical substitution
- wording prescription
- superfluous terms
- sentence truncation
- text compaction
- deletion macro-rule
- construction macro-rule
- generalisation macro-rule
- reordering sentence aggregation
- latent semantic analysis
- semantic equivalency
- information equivalency.

Some of these techniques are shown by virtual examples in Samples 3, 4, 5, 6, 7, and 8. For instance, the goal of revision techniques is to improve the generated summary. Revision techniques may accomplish its goal with or without referencing the source document(s) (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007; Cremmins, 1996; Nomoto, 2007). When revision techniques function without taking into account its original source document(s), they will alter the generated summary to improve it. This may be accomplished by adding some external elements to the generated summary. These external elements are neither from the summary nor from its original source document(s). On the other hand, when revision techniques function by taking into account its original source document(s), they will also alter the generated summary to improve it, which may be accomplished by adding some internal elements to the generated summary. These internal elements may be from the summary, from its original source document(s), or both. In either case, such elements can definitely be employed to embed data in the generated summary. It is worth noting that the revision techniques are, most likely, used by abstraction-based summariser, as shown by virtual examples in Samples 5, 6, 7, and 8.

Sample 3: Illustrates the original document during virtual extraction procedure. Some techniques such as superfluous terms, sentence truncation, text compaction, deletion macro-rule, construction macro-rule may be involved in the extraction procedure.

Sample 3 (see online version for colours)

~~*Automatic summarization is the scientific art of representing the essence of a long document(s) in a significantly smaller document(s) than its original by employing computer programs. The field is traced back to the 1950's. However, the field of automatic summarization has enjoyed significant progress in recent years and is still promising more in the future.*~~ *Automatic summarization systems employ a procedure that may be based on one or more of the following: statistical process, knowledge base, artificial intelligence, computational linguistics, and other related techniques to achieve its goal.* ~~*Some examples of automatic summarization systems are AutoSummarize, SweSum, Inxight Summarizer, IBM Intelligent Miner, DimSum, and more.*~~ *Automatic summarization approaches may categorize into three types: high level, low level, and hybrid approaches.*

Sample 4: Illustrates the abstract after virtual reorder procedure of the extracted text. The abstract started with the second extracted sentence and ends with the first extracted sentence.

Sample 4

Automatic summarization approaches may categorize into three types: high level, low level, and hybrid approaches. Automatic summarization systems employ a procedure that may be based on one or more of the following: statistical process, knowledge base, artificial intelligence, computational linguistics, and other related techniques to achieve its goal.

Sample 5: Illustrates the abstract (in Sample 4) during virtual revision procedure. Some techniques such as paraphrasing rule, lexical substitution, wording prescription, superfluous terms, sentence truncation, text compaction, deletion macro-rule, construction macro-rule, generalisation macro-rule, reordering sentence, discourse, aggregation, latent semantic analysis, semantic equivalency, information equivalency, and information retrieval may be involved to generate abstracts. All underlined words are added to the abstract during revision procedure. Additionally, the highlighted words are external elements that are not existed in the original document input.

Sample 5 (see online version for colours)

~~*Automatic summarization*~~ Summarizers ~~*approaches may categorize into*~~ *are three types:* ~~*high level*~~ shallow, deep ~~*low level*~~, *and hybrid* ~~*approaches*~~. ~~*Automatic summarization systems employ*~~ They use ~~*a procedure that may be based on one or more of the following:*~~ *statistical*~~ *process*~~, *knowledge base, artificial intelligence,* and *computational linguistics*~~, and other related~~ *techniques*~~ *to achieve its goal*~~ .

Sample 6: Illustrates the abstract (in Sample 4) after virtual revision procedure. All underlined words are embedded to the abstract during the revision procedure. Additionally, the highlighted words are external elements that do not exist in the original document input.

Sample 6 (see online version for colours)

> Summarizers are three types: shallow, deep, and hybrid. They use statistical, knowledge base, artificial intelligence, and computational linguistics techniques.

Sample 7: Illustrates the abstract (in Sample 4) during a different virtual revision procedure other than Samples 5 and 6. Some techniques such as paraphrasing rule, lexical substitution, wording prescription, superfluous terms, sentence truncation, text compaction, deletion macro-rule, construction macro-rule, generalisation macro-rule, reordering sentence, discourse, aggregation, latent semantic analysis, semantic equivalency, information equivalency, and information retrieval may be involved to generate abstracts. All underlined words are added to the abstract during revision procedure. Additionally, the highlighted words are external elements that do not exist in the original document input.

Sample 7 (see online version for colours)

> Automatic ~~summarization~~ Summarizers ~~approaches may categorize into~~ are three types: extractor, abstractor ~~high level, low level~~, and hybrid ~~approaches~~. ~~Automatic summarization systems employ a procedure that may be~~ They are based on ~~one or more of the following: statistical process, knowledge base,~~ artificial intelligence~~, computational linguistics, and other related~~ techniques ~~to achieve its goal~~.

Sample 8: Illustrates the abstract (in Sample 4) after different virtual revision procedure other than Samples 5 and 6. It is noted that both abstracts of Samples 6 and 8 are different in words, in sentences, and even slightly in meaning. All underlined words are embedded to the abstract during revision procedure. Additionally, the highlighted words are external elements that do not exist in the original document input.

Sample 8 (see online version for colours)

> Automatic Summarizers are three types: extractor, abstractor, and hybrid. They are based on artificial intelligence techniques.

3.3 Multi-document

Multi-document summarisation techniques are capable of handling multiple documents to generate the required summary (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007; Cremmins, 1996; Nomoto, 2007; Mana-Lopez et al., 2004; Sekine and Nobata, 2003; Afantenos et al., 2007; Koumpis and Renals, 2005). The demand of modern age such as Word Wide Web and data mining rendered the field of multi-document summarisation very active and imperative. From the point of view of Sumstega methodology, it is argued that the multi-documents input may play a critical role for easing the task of generating a mature summary. For example, the use of domain-specific subject and knowledge base can be used for generating a summary where some linguistics that does not exist in the 'original documents input' may be used in the

generated summary. However, from a linguistics point of view, it is most likely can be more accurate to use the linguistics of the input documents rather than using other linguistics that do not exist in the original input documents. For instance, when a journalist is having a discussion with an author of a book and the journalist uses a speech from the author's text, it is called 'using the same language' because he is using the author's words to prove a point in order to convince him. It is argued that the multi-document summarisation techniques may be feasible to play a role in resolving some of these problematic issues, e.g., linguistic flaws such as the flow of text-cover, etc., of contemporary linguistic steganography approaches. This may be accomplished by employing the linguistics from the multi-documents input to generate a mature text-cover (summary-cover). Since multi-document summarisation techniques are the extension of single document summarisation techniques, the demonstrated samples and examples in this section are sufficient for understanding how multi-document summarisation techniques can be used (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007; Cremmins, 1996; Nomoto, 2007; Mana-Lopez et al., 2004; Sekine and Nobata, 2003; Afantenos et al., 2007; Koumpis and Renals, 2005).

3.4 Cross-lingual

Summarisation techniques are not only capable of handling monolingual-documents, but they are also capable of handling multilingual-documents (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007). Cross-lingual summarisation techniques can handle several languages where the input and output documents are in different languages (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007). Both cross-lingual summarisation techniques and MT techniques may intersect. However, from the point view of Sumstega methodology, cross-lingual summarisation techniques are employed differently than in the translation-based steganography approach. This is because the translation-based steganography approach is errors-based. In other words, it hides data in the errors (noise), and it generates more noise to hide data. On the other hand, Sumstega neither camouflages data in a noise nor generates noise when concealing data in summary-cover (Sumstega cover). Instead, when Sumstega employs cross-lingual summarisation techniques, it conceals the data in the natural varietal elements (e.g., words, sentences, partial sentences, etc.) that are produced by the natural and legitimate process of the summarisation techniques. Obviously, cross-lingual techniques can increase the room for concealing data in Sumstega cover (summary-cover). An example of this technique will be similar to the demonstrated samples in this section.

4 Sumstega methodology

To illustrate Sumstega, consider the following scenario. Bob and Alice are on a spy mission. Before they start 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. To make this work, they developed a steganographical summariser that is capable of generating legitimate various summaries. Then, It predetermines a particular single unique course of generating summaries to generate an original summary-cover (unaltered), which it contains no hidden message at this moment. Then, it embeds a message by performing a summarisation substitution

procedure on the original summary-cover using these legitimate variations of the generated summaries. This process is done in such a way that the summary-cover appears as an ordinary summary. Yet, they establish a business relationship Bob and Alice are journalists working for the same corporation. They generate summaries of real news data to make their covert communications more legitimate. When Bob wants to send a covert message to Alice, Bob either posts summary-cover online for authorised clients and staff to access or he sends them via email. Covert messages transmitted in this manner will not look suspicious because Bob and Alice are journalists and their interaction is legitimate and innocent. The use of automatic summary in such a profession is natural given the space constraints and time a reader may dedicate for reading. Moreover, Bob and Alice are not the sole recipients. There are other non-spy journalists, staff, and clients who send and receive such documents, further warding off suspicion. However, only Bob and Alice will be able unravel the hidden message because they know the rules of the game. They reveal a message by comparing the summary-cover that contains a hidden message to the unaltered original summary, which is agreed on its path in advance, then decode all substituted pieces (e.g., sentences, words, etc.) according to the predetermined encoding system to be used.

The above scenario illustrates how Sumstega methodology can be used effectively. Sumstega methodology is demonstrated in the remainder of this section in detail.

4.1 Sumstega architecture

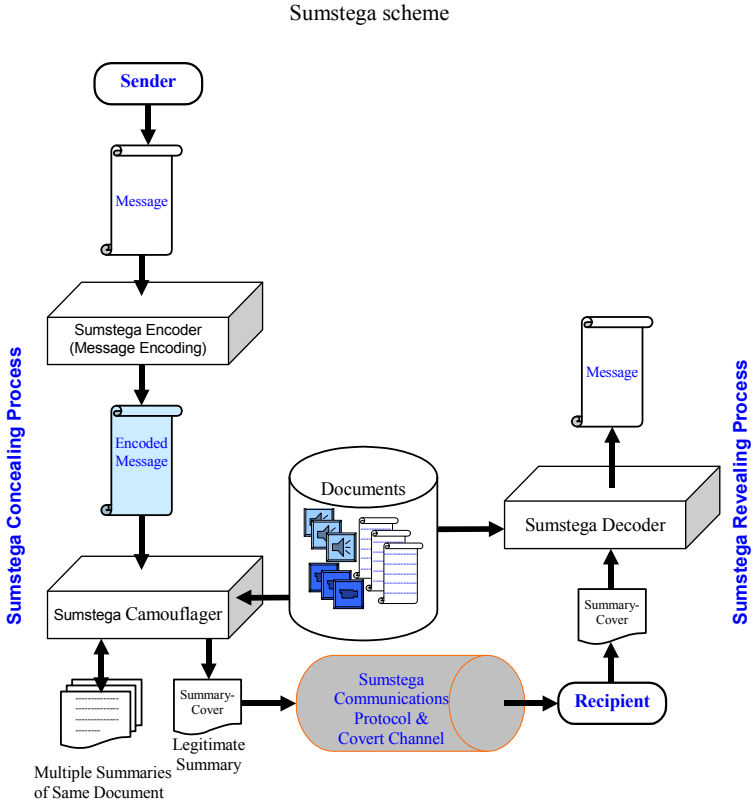
The core idea of Sumstega methodology is that the camouflage process of data has to be accomplished in the natural and legitimate variations that are produced by the process of the automatic summarisation techniques. As demonstrated in Section 3, different automatic summarisation techniques, implementations, or both generate output variations (different summaries) of the same input(s). It is like multiple summaries of the same document(s) that are generated by different people where everyone will summarise the document(s) differently regardless of similarities in the meaning (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007). Therefore, Sumstega methodology takes advantage of such variations to conceal data. As stated earlier, it manipulates PFAST, as shown in Section 3, in order to generate output variations that can be employed for embedding data in the generated summaries without violating the pattern of automated summaries. It generates summary-cover that looks legitimate by exploiting PFAST such as the weight (e.g., weight of frequency, location, semantic), paraphrasing, truncation, reordering, semantic and information equivalency, etc. In addition, Sumstega methodology imposes on the communicating parties to establish a covert channel in order to transmit summary-covers. The following is an overview of the Sumstega architecture, which consisted of four modules, as shown in Figure 1:

- *Sumstega encoder (Module 1)*: encodes a message in an appropriate and required form for the camouflaging process (Module 3).
- *Sumstega camouflager (Module 2)*: generates variety of legitimate summaries, as demonstrated in Section 3, to be employed by this camouflaging process to generate a summary-cover, in which data are embedded.
- *Sumstega communications protocol (Module 3)*: configures the basic protocol of how a sender and recipient would communicate covertly. Obviously, it includes the covert

channel for delivering a summary-cover to the recipient and the decoder scheme to unravel a hidden message.

The above modules are detailed in the following subsections.

Figure 1 Illustrates Sumstega architecture (see online version for colours)



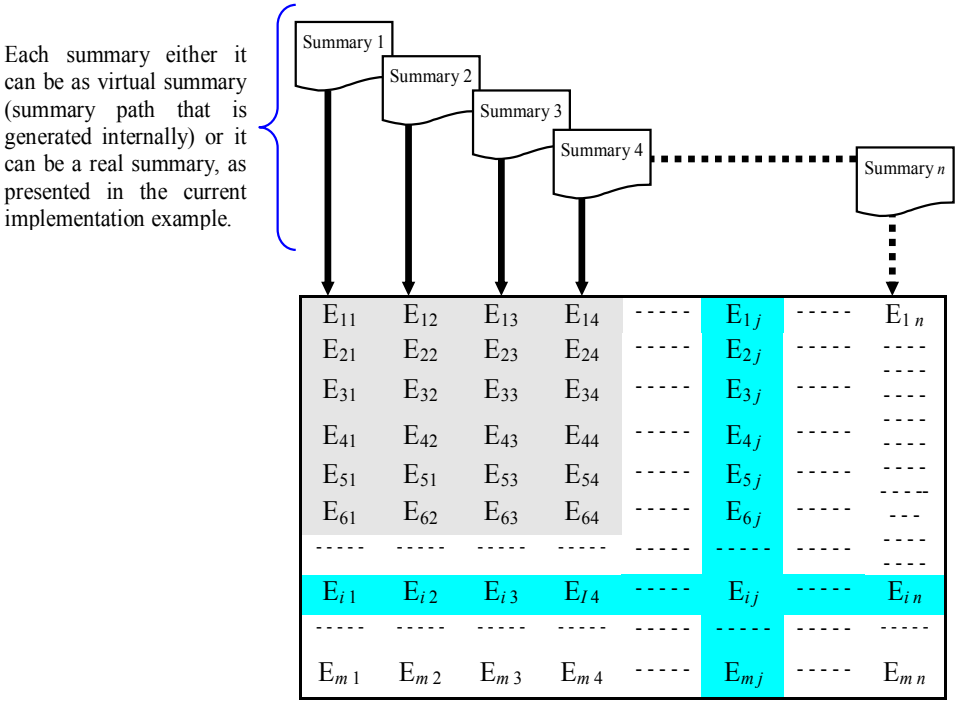
4.2 Sumstega encoder (Module 1)

Sumstega encoder encodes a message in an appropriate and required form for the camouflaging process (Module 2). In general, Sumstega does not impose any constraint on the message encoder scheme as long as it generates a steganographical code that can be embedded in a summary-cover. However, the selection and the implementation of the most appropriate encoding scheme are factored by other requirement such as the need of encryption, compression, etc. Implementing Sumstega encoder can be accomplished either by constructing the required encoder from scratch or by employing contemporary steganographic encoding techniques to encode messages. Given the availability of numerous steganographical encoding techniques, including encryption and compression techniques, in contemporary literature that can be employed by Sumstega methodology, the balance of the discussion in this paper is focused on the generation of Sumstega cover (summary-cover) rather than message encoding. In this paper, the implementation of Sumstega encoder is mainly based on the number of different summaries and the type of

different elements (e.g., words, sentences) that can be employed to generate steganographic code, regardless of whether or not that other techniques may be included, e.g., encryption, compression, etc. Since the focus of this paper is steganography and the use of encryption and compression techniques are not part of the contribution, such techniques neither are discussed nor are used in this article.

In the implementation example shown in this paper, a message is encoded as follows. A message is converted to a binary string. The binary string of a message can be a binary of ciphertext or compressed representation. The binary string is then partitioned into groups of m bits. The value of m is determined based on the number 'n' of different summaries that can be produced, as specified by Sumstega camouflager (Module 2). Basically, m is set to log n. If $n = 4$, i.e., four different summaries, the bit pattern 00, 01, 10, or 11 (as shown in Section 4.3 and in the implementation example in Section 4.4) will be applied to the first, second, third or fourth internally generated summaries, respectively. This if an element (e.g., word, sentence) is unique the internally generated summaries. On the other hand, multiple matches imply null data bits, e.g., if an element and its index are same in all generated summaries. Again, this encoding scheme is just for illustration and many alternatives, and more sophisticated, can be employed.

Figure 2 Illustrates Sumstega matrix (see online version for colours)



Notes: Sumstega matrix contains all the elements of each summary as shown in the matrix. The grey part (shaded part) in Sumstega matrix represents the current implementation example and the sample example of summary-cover (Sumstega cover) which are pursued by Table 1 and in Figure 3 as if only four different summaries can be generated.

4.3 Sumstega camouflager (Module 2)

Sumstega camouflager engine generates the summary-cover that conceals data by employing Module 1 along with different implementations, techniques, etc., of automatic summarisation. Technically, there are numerous ways, as expected, to implement Sumstega camouflager engine. However, in this paper Sumstega camouflager engine is implemented based on the following algorithm, which consists of seven submodules:

1 *Submodule 1:* it generates variety of legitimate summaries by employing different implementations, techniques, etc., of automatic summarisation, as demonstrated in Section 3.

2 *Submodule 2:* it predetermines one of the generated summaries by Submodule 1, which is a particular path of generating summaries, to be the mother summary (original summary). This step will ease the process only for the legitimate recipient to reveal the hidden message. Simply, it allows the decoder to compare the summary-cover to the mother summary in order to determine all alterations, which represents the hidden message. These alterations will then be assigned the values of the steganographic code to unravel the hidden message. The steganographical code is the same set of values that are used by the sender to conceal data.

3 *Submodule 3:* it maps the generated summaries by Submodule 1 into a matrix, which is called Sumstega matrix, as shown in Figure 2. Sumstega Matrix is m x n where m is the number of rows and n is the number of column. The n, which is the number of column, is the number of how many different summaries can be generated by Submodule 1. In other words, it maps one summary in each column of Sumstega Matrix. The m, which is the number of rows, is the number of how many elements of each summary. The value of m should be same for all generated summaries since it is doable to have such control especially for sentence level, e.g., sentence extraction summarisation. However, if the value of m in some cases, as exception case, is vary from one summary to another, in this case, m can be same for all generated summaries by assigning empty values, for any summary that is contains less elements than its peer summaries. This in order to render m for all summaries to have the same value. The index of rows is denoted by i while the index of column is j. Note that a mother summary will be a particular column of Sumstega matrix which Sumstega system is configured by pre-agreeing upon it.

4 *Submodule 4:* it compares only the peer elements of all summaries to determine the differences among all summaries. Mathematically, it compares only the peer elements of the same row to determine the differences among all elements of only the same row. In other words, it compares the elements that have the same value of i while the value of j changing from its initial value, which is equal 1, to its maximum value, which is equal n, in order to distinguish all different elements of the entire Sumstega matrix. For instance, the result of this step may be accomplished by marking all elements as follows: same elements, unique elements, and semi unique elements.

5 *Submodule 5:* it encodes Sumstega matrix using the general steganographic code of Sumstega encoder (Module 1). For example, it may encode the entire Sumstega Matrix by general steganographic code values as follows. The elements that are the

same in the entire row may be non-coded elements, which may assign value of null. The elements that are unique may assign a full value of the steganographical code. Finally, some elements are semi unique or partially different which may assign a partial value of the steganographical code. To emphasise, if Sumstega generates maximum of four different summaries then the full value of the steganographical code may be two bits, e.g., 00, 01, 10, or 11 and obviously the partial value of the steganographical code can be one bit either 0 or 1.

6 *Submodule 6:* it generates a summary-cover by selecting the mandatory elements that may have null values and all elements that have the same steganographic values of the encoded message. The mandatory elements are, most likely, have null values which cannot conceal data because these elements along with their indices are same in all generated summaries which are not different elements.

7 *Submodule 7:* it evaluates a summary-cover to assure that the summary-cover appears normal by using the evaluation techniques (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007).

4.4 Implementation example

This subsection demonstrates an example of an actual implementation of Sumstega methodology, discusses some important aspects of the implementation, and highlights possible directions for implementation. The purpose of the presented implementation in this paper is to show the Sumstega'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 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. Therefore, neither cryptosystem nor compression technique is used in this paper. Given the availability of numerous encoding, encryption, and compression techniques in the contemporary literature that can be employed by Sumstega methodology, the discussion in the balance of this section will focus on the generation of Sumstega cover (summary-cover) rather than the message encoding. Obviously, the technique presented in this paper, as stated earlier, is just an example of possible implementation, but Sumstega methodology can be implemented differently. In this example, Sumstega encoder (Module 1) converts a message to the binary string of its ASCII representation. Obviously, it is expected that Sumstega encoder may be implemented differently and includes a procedure of both data compression and encryption during the generation of the steganographic code, as mentioned earlier. Applying such techniques is a trivial task. As mentioned in Section 4.3, the goal is to construct a Sumstega camouflager (Module 2) that is capable of applying the above seven submodules in Section 4.3. In this example, as illustrated in Figures 1 and 2, Sumstega employs several contemporary summarisers, in particular, four summarisers (Microsoft Word 97; Hassel and Dalianis; LTRC, IIIT; Auto Summarizer, http://mskw.cipher-sys.com/Lectern/summary_submitter.asp) that are capable of generating numerous variations of summaries. Obviously, Sumstega methodology may employ more summarisers or build Sumstega summariser from scratch without employing contemporary summarisers. Consequently and according to the algorithm of Sumstega camouflager (Module 2), the presented Sumstega system maps the

elements of the generated summaries in a table, called Sumstega Matrix, and compares them to assign Sumstega code from the Steganographical code table which is detailed Table 1. Then, it selects all elements that match the encoded message, which the binary code of a message in this paper, along with the non-coded elements in order to generate a summary-cover.

Table 1 Illustrates Sumstega code example that will be employed by Sumstega camouflager (the camouflage procedure) to conceal a message.

Steganographical code					
Note	*Elements type*	*Sumstega code*			
		Summary 1	*Summary 2*	*Summary 3*	*Summary 4*
No overlap	Unique	00	01	10	11
	Semi-unique	00	01	10	11
Non-coded	Two options	0	1	1	1
	Same	Null	Null	Null	Null

Note: This is just an example.

Figure 3 Illustrates Sumstega matrix and shows the selected elements (the shaded squares) that conceal a message (see online version for colours)

Sumstega code	***All elements of summries***			
	00	*01*	*10*	*11*
	Summary 1	*Summary 2*	*Summary 3*	*Summary 4*
01	E_{11}	E_{12}	E_{13}	E_{14}
	E_{21}	E_{22}	E_{23}	E_{24}
00 01	E_{31}	E_{32}	E_{33}	E_{34}
11	E_{41}	E_{42}	E_{43}	E_{44}
	E_{51}	E_{51}	E_{53}	E_{54}
	E_{61}	E_{62}	E_{63}	E_{64}

Notes: These elements form the summary path of the Sample 9 of Sumstega cover. This is an actual process of generating summary-cover (Sumstega cover).

4.4.1 Sample of Sumstega cover

The grey part in Sumstega matrix in Figure 2, which is mapped to Figure 3, represents the current implementation example and Sample 9 of summary-cover (Sumstega cover). The presented sample generated by using an input public news article from New York Times (New York Times Magazine Online, 2008). Sample 9, the presented summary-cover, generated from the four summaries that contain five to six elements in each summary path (the generated summary). This implementation example is based on sentence extraction summarisation techniques. Therefore, in this example an element is referred to the extracted sentence, as denoted by letter E in Sumstega matrix, as shown in Figure 2. The following is example of Sumstega cover (Sample 9) conceals 8 bits of data that represent the letter 'G' which is in binary '01000111'.

Sample 9: Illustrates the summary-cover (Sumstega cover) by employing only extraction-based summarisation techniques. As shown, the presented sample of Sumstega cover has the same qualities of its comparable summaries that contain no hidden data.

Sample 9

```
Although the ministry did not confirm that the
drawdown would begin in March, it confirmed that
the ministry was "expecting to see a fundamental
change of mission in early 2009." The plans by
Britain — and its talks with Washington — have been
complicated by pressure from the Bush
administration to couple the British drawdown in
Iraq with an increase in British forces in
Afghanistan. The leaking of the British withdrawal
plan appeared to have been prompted, at least in
part, by President-elect Barack Obama's victory in
the election last month and his plans to draw up a
timetable for the withdrawal of American troops
from Iraq. Within 18 months of the invasion,
British commanders were complaining privately that
the Americans lacked Britain's colonial experience
in countries like Iraq, & that the heavy use of
firepower against Mr. Sadr was counterproductive.
```

4.5 Sumstega communications protocol (Module 3)

The communicating parties configure the communications protocol of Sumstega system, as shown in Figure 1, in order to communicate covertly by predetermining the following. First, the particular specifications of Sumstega system used including its decoder and the input used to generate the steganographic cover which is already known for public, e.g., news article. Second, the covert channel for transmitting securely summary-covers among communicating parties. Once communications protocol is agreed upon, the intended parties are ready to communicate covertly with each other using Sumstega. The first item is addressed by Modules 1 and 2, which are discussed in the previous subsections. The second item is a particular covert channel that mainly defines how the cover will be delivered to the recipient without raising suspicion. Covert transmittal of the steganographic cover is very crucial to the success of steganography. At the core of the cover transmittal issue is how to prevent the association between the sender and recipient from drawing suspicion. For example, exchanging email messages would automatically imply a relationship between the communicating parties. Similarly, downloading files from a website indicates an interest in the accessed material. With advances in monitoring tools for network and internet traffic, profiles of user's access pattern can be easily established. An adversary most probably will suspect the presence of a hidden message, even if the content does not look suspicious, because of the observed traffic pattern and the lack of a justification for the interest in the contents of such traffic. For example, if a sender or recipient his pretended profession is an online-news and sends or receives other suspicious documents such nuclear documents then suspicious can easily be raised. Someone works in an online-news field may send or receive only documents that are justifiable to be obtained such as news reports. Therefore, it is very important to rationalise the sending and receiving of steganographic cover in order to avoid attracting any attention that may trigger an attack. Sumstega enables an effective solution for the issue of legitimising a cover transmittal. The use of a particular domain(s) allows establishing a covert channel in a form of legitimising the association among communicating parties and thus sharing a summary-cover would appear an ordinary practice. The use of summaries is very popular all over the world such

as the example of Bob and Alice, Section 3. Thus, the transmission of the summary-covers via e-mail, posting them on web pages, etc., is a natural matter that does not raise suspicion.

4.6 Bitrate

Nonetheless, the presented implementation of Sumstega scheme may achieve bitrate roughly from 0.064% up to 0.20% and with an approximate average of 0.12%. This bitrate is limited to only the current implementation example, which employs only one type of summarisation technique, namely an extraction technique. However, there are numerous summarisation techniques (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007), as detailed in Section 3 that can be employed by Sumstega such as: abstraction, revision, discourse, paraphrasing rule, lexical substitution, semantic equivalency, information equivalency, etc. Obviously, employing such techniques can easily increase the bitrate. Unfortunately, there is no free or affordable summariser that uses these techniques and that is why the current implementation example uses extraction-based summarisers as detailed early, which are either affordable or free. Therefore, improving Sumstega's bitrate is feasible and will be investigated in future work. In regard to the message size, the size of a message is a concern for most, if not all, steganography approaches. However, in the presented implementation example of Sumstega scheme, Sumstega camouflages a long message. Note that if a particular steganographic system achieves low bitrate, it does not imply that a long message cannot be concealed by such scheme. For example, the low bitrate of the text-cover will require long text-cover to camouflage a long message. Generally, text files do not burden a network like image or audio files where the size of either image or audio is huge compared to text files. Obviously, Sumstega is capable of concealing a long message, which is not applicable to be presented in this paper due to space constraints.

5 Steganalysis validation

The aim of this section is to show the resilience of Sumstega 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 Sumstega, as a public methodology, but he does not know the Sumstega configuration that the sender and recipient employ for their covert communication.

5.1 Traffic attack

Traffic attack (Desoky, 2010c) 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, accessing, or obtaining such materials when the users have no legitimate reason to do so, then suspicion can be raised without any further investigation. For example, a medical doctor communicates using weather analysis report documents with one of his patients or vice versa. This can easily raise suspicion because a medical doctor should send medical documents not

weather analysis report documents. Furthermore, if the patient has no legitimate reason for receiving or sending such documents, then suspicion can also be easily raised. Traffic attack can be applied to any contemporary steganographic technique 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.

Sumstega methodology ensures that the communicating parties establish a secure covert channel for transmitting the hidden message covertly. In other words, Sumstega naturally camouflages the delivery of a hidden message in such a way as to appear legitimate and innocent. Thus, suspicion is averted during the transmittal of a hidden message. The scenario in Section 3 demonstrates how Bob and Alice communicated in a natural way that can avert suspicion. This scenario shows how Sumstega can be effective for camouflaging the transmittal of a hidden message. When a particular text under investigation is accessed by people who have a legitimate reason to obtain such information, suspicion is averted. This is because the professions of the intended users play the role of camouflaging the delivery of hidden messages between the intended users such as the example of Bob and Alice. On the other hand, if Bob sends information other than that related to his journalist profession, such as a medical report to Alice, suspicion will be raised without any further investigation. As long as there is a legitimate reason for sending and accessing this material, suspicion can be averted. As a result, the Sumstega 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 of the unseen.

Investigating all similar traffics are impossible because there is an astronomical amount of these traffics to suspect, rendering Sumstega favourable as a steganography methodology to be adopted.

5.2 Contrast and comparison attacks

One of the intuitive sources of noise that may alert an adversary is the presence of contradictions, which is called contrast attack (Desoky 2010c, in process), in the text. Finding such contradictions in a summary-cover of consumer prices index (CPI) report, the value of a product edging up while saying that it has decreased. It is worth noting that the traffic analysis (Desoky 2010c, in process), discussed in the previous section (Section 5.1), can also be pursued as a base for launching contrast attacks in case the data are not publicly accessible. In the later case, comparing current data (Sumstega cover/summary-cover) against a record of old data searching for any inconsistency over some period of time can be tracked. Countering against such an attack is always a challenge because it requires consistency with data previously used over an extended period of time. Contradictions would surely raise suspicion about the existence of a hidden message. Sumstega methodology, as demonstrated through the example in Section 4, is simply made naturally contrast-aware in order to avert such attacks (Desoky, 2010c).

Noise, in the context of comparison attacks, reflects an alteration of authenticated data. The goal is to find any incorrect or altered data that may imply the presence of a hidden message. When Sumstega employs public documents to generate the summary-cover (Sumstega cover), in fact it, naturally and legitimately, alters these documents by employing summarisation techniques to generate natural and legitimate

summaries to camouflage data in the natural and legitimate generate summary. Therefore, countering such an attack is in vain. To emphasise, whether the summaries are used by Sumstega methodology or for non-steganographical purposes (e.g., helping people to read long documents in a reasonable time), the generated summaries in both cases is similar and legitimate. Therefore, there is no noise to be detected by comparison attack. Definitely, suspicion is averted during such attacks. As long as an attack is known, it is feasible to be avoided simply by constructing the Sumstega scheme to be aware of contemporary attacks. For example, if the communicating parties are concerned about comparison attacks then Sumstega scheme should be made comparison-aware in order to avoid such an attack, as demonstrated in the above examples in Section 4.

5.3 Evaluation attack

There is an entire field called evaluation of automatic summarisation systems that is employed for examining summarisers to improve them (Mani and Maybury, 1999; Mani, 2001; Marcu, 2000; Jones, 2007). Adversaries may take advantage of evaluation techniques to investigate Sumstega cover (summary-cover), and if the result of the evaluation is indicating that the summary is below the acceptance level, then it is possible that the summary contains a hidden message. Steganographically, evalation techniques can play the role of well known attacks where the steganographers have to counter against. As indicated early, it is feasible to fool any attack as long as the attack model is known simply by constructing the steganographic scheme as attack-aware (Desoky, 2010c, in process). Therefore, it is essential for whoever is adopting Sumstega methodology to take advantage of evaluation techniques to assess and examine Sumstega cover (summary-cover) before the actual use of Sumstega scheme. As a result, evaluation techniques will not only benefit the field of automatic summarisation but also will benefit Sumstega methodology to be resiliently resistant against such attacks.

5.4 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 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 that is under attack is abnormal. The summaries that are naturally and legitimately generated by the contemporary automatic summariser systems differ from summaries that are generated by human and may have their linguistic issues or flaws. Obviously, an adversary cannot employ the detection of such issues or flaws to attack Sumstega cover (summary-cover) because these issues and flaws exist in the legitimate summaries which do not contain any hidden message. Therefore, these linguistic issues or flaws, from a steganographical point view, pose no concerns for three reasons. First, as mentioned before, these issues and flaws exist in the legitimate summaries that do not contain any hidden message. Second, nothing is concealed in errors. Third, Sumstega methodology does not generate any flaw during the concealment process of a message. Therefore, it is obvious that Sumstega is capable of passing such an attack by both human and machine examinations.

5.5 Statistical signature

In this paper, the statistical signature (profile) of a text refers to the frequency of words 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 since it can be easily automated and combined with traffic analysis. However, Sumstega is resiliently resistant to statistical attacks as demonstrated by the experimental results below.

Human language in general, and the English language in particular, have been statistically investigated (Zipf, 1968; Li, 1992) to discover their statistical properties. The most notable study on the frequency of words was done by George Kingsley Zipf (Zipf, 1968; Li, 1992). 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. 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 nth 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 domain-specific subjects by different authors, writing-styles, 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 Sumstega's experiment applied Zipf's law directly on Sumstega cover considering the worse case scenario that an adversary knows Sumstega methodology and knows if there is a hidden message, where the hidden message is concealed. Unlike Zipf's experiment, the Sumstega experiment applied Zipf's law on a short piece of text with a unique domain-specific subject. Based on the experimental observation, as shown below in Figure 4, Sumstega cover which contains a hidden message holds a Zipfian slope of –0.8128. On the other hand, the unaltered summary comparables, which they do not contain any hidden message, hold a similar Zipfian slope of –0.4721, –0.5095, –0.4282, and –0.4736 respectively with a roughly average of –0.47085. Furthermore, when applying Zipf's law on two different domain-specific subjects, such as smoking cessation and CPI, using their original textual documents which are neither summarised nor contained hidden message, observed the following. In the first domain (smoking cessation), there are two Zipfian regions, as shown in Table 2: the highest Zipfian region holds a Zipfian slope in the range of –0.8118 to –0.8993; and the lowest Zipfian region holds a Zipfian slope in the range of –0.5745 to –0.6942. In this experiment, the highest Zipfian region is in the range of –0.8118 to –0.8993 and is the closest to the ideal Zipfian of –1. Similarly, the above observation was also observed, as shown in Table 2, in a different domain-specific subject of CPI, where it holds a Zipfian slope with an average of –0.74835, the highest Zipfian region in the range of –0.8245 to –0.9557, and the lowest Zipfian region in the range of –0.6052 to –0.7493. The main observation for both

domains (CPI and smoking cessation) is that they do not obey Zipf's law because they are significantly far from the ideal Zipfian of –1.

Figure 4 Illustrates Zipfian for a Sumstega cover (see online version for colours)

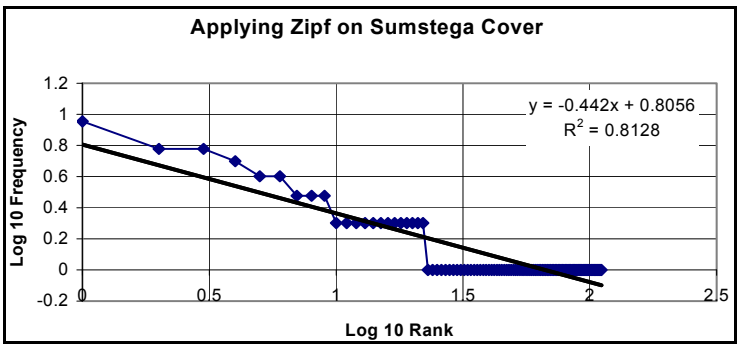


Table 2 The Zipfian distribution (logarithmic scale) for text without hidden message and without summarisation of two different domains smoking cessation and CPI

Text without hidden message of two different domains without summarisation						
Text #	*Smoking cessation*			*Consumer prices index (CPI)*		
	Equation	R^2	*Slope (–a)*	*Equation*	R^2	*Slope (–a)*
1	–0.7094x + 1.6976	0.9276	–0.7094	–0.8245x + 1.4915	0.9329	–0.8245
2	–0.6596x + 1.4729	0.9237	–0.6596	–0.8741x + 1.698	0.9467	–0.8741
3	–0.618x + 1.3766	0.9113	–0.618	–0.7412x + 1.266	0.9251	–0.7412
4	–0.7339x + 1.8687	0.9264	–0.7339	–0.8542x + 1.6855	0.9512	–0.8542
5	–0.6922x + 1.6727	0.9304	–0.6922	–0.9557x + 1.8569	0.9559	–0.9557
6	–0.6377x + 1.377	0.8922	–0.6377	–0.737x + 1.4103	0.9201	–0.737
7	–0.674x + 1.4475	0.9218	–0.674	–0.737x + 1.4103	0.9201	–0.737
8	–0.5745x + 1.3416	0.9012	–0.5745	–0.758x + 1.2825	0.9091	–0.758
9	–0.7227x + 1.6441	0.9244	–0.7227	–0.7493x + 1.428	0.9109	–0.7493
10	–0.6558x + 1.388	0.9146	–0.6558	–0.6697x + 1.4098	0.9173	–0.6697
11	–0.6141x + 1.4108	0.9145	–0.6141	–0.705x + 1.4186	0.9257	–0.705
12	–0.7221x + 1.6445	0.943	–0.7221	–0.6559x + 1.2942	0.8882	–0.6559
13	–0.8603x + 2.0621	0.9451	–0.8603	–0.7171x + 1.1889	0.9159	–0.7171
14	–0.8993x + 2.4766	0.9592	–0.8993	–0.6052x + 0.9868	0.8342	–0.6052
15	–0.899x + 2.4759	0.9591	–0.899	–0.9121x + 1.5605	0.9461	–0.9121
16	–0.6942x + 1.5498	0.9202	–0.6942	–0.8504x + 1.3719	0.9015	–0.8504
17	–0.6432x + 1.4241	0.887	–0.6432	–0.7116x + 1.3634	0.8902	–0.7116
18	–0.767x + 1.9058	0.9409	–0.767	–0.7093x + 1.363	0.9035	–0.7093
19	–0.7944x + 1.7776	0.9282	–0.7944	–0.7352x + 1.329	0.9185	–0.7352
20	–0.7018x + 1.6793	0.9279	–0.7018	–0.7085x + 1.3469	0.9021	–0.7085
21	–0.7441x + 1.9242	0.9434	–0.7441	–0.6697x + 1.4098	0.9173	–0.6697
22	–0.62x + 1.445	0.8853	–0.62	–0.6603x + 1.2676	0.8973	–0.6603
23	–0.8118x + 2.0752	0.9449	–0.8118	–0.671x + 1.3073	0.9037	–0.671
Average			–0.71518			–0.74835

Notes: The equation is a linear curve fitting of the results. R^2 is the squared error.

The conclusion of Sumstega experiment of word frequency is as follows. Zipfian slop of a Sumstega cover is –0.442, which falls in the Zipfian region of its domain. When applying Zipf's law, Sumstega cover should be similar to a Zipfian slope of the summaries of its domain-specific subject (the unaltered authenticated data summaries of the same domain that contains no hidden message), and it is not required to fully obey Zipf's law (Zipfian of –1). To emphasise, if the Zipfian slope of the Sumstega domain-specific subject (the unaltered authenticated data of the same domain that contains no hidden message) is equal to *N* value, then Sumstega cover should be either equal or close to that *N* value. Generally, it is feasible to fool any attack as long as the attack model is known, simply by constructing the steganographic scheme as attack-aware. Furthermore, it is feasible to alter a natural language in a way that can fool Zipf's law if it is required. Simply, Sumstega can be designed as Zipf-aware since the statistical model is already known.

6 Conclusions

The presented Sumstega methodology achieves legitimacy by basing the camouflage of both a message and its transmittal on a summarisation of documents. The necessity of automatic summarisation increases in business, science, World Wide Web, education, news, etc., because no one has time to read everything. This renders summarisation an attractive steganographic carrier. Yet, the high volume of traffic for accessing and generating summaries makes an adversary's job impossible to investigate all of them and allows the communicating parties the opportunity to establish an innocent covert channel to transmit hidden messages. Therefore, Sumstega takes advantage of the automatic summarisation techniques to conceal data. This is accomplished by manipulating the PFAST in order to embed a message without violating the pattern of an automated summary. The implementation and steganalysis validation demonstrate that Sumstega methodology is capable of achieving the steganographical goal.

Some of the main advantages of the Sumstega methodology over all other approaches that are demonstrated in this paper are as follows. First, the tremendous amount of summary in electronic and non-electronic format makes it impossible for an adversary to investigate all of them. This makes it extremely favourable as a steganographic cover in covert communications. Second, Sumstega is resilient against contemporary attacks including an attack by an adversary who familiars with Sumstega (Sumstega is a public methodology). Third, Sumstega does not apply a particular pattern (noise) that an adversary may look for. Fourth, the concealment process of Sumstega has no effect on the linguistics of the generated cover (summary-cover). Therefore, a summary-cover is linguistically legitimate comparing to its peer summaries and is thus capable of passing both computer and human examinations. Fifth, Sumstega can be applied to all languages. It is unlike the translation-based approach, where the continual improvement of MT will eliminate the use of the translation-based approach, the improvement in summarisation systems is promising and will make Sumstega more stable in future. Sixth, there is adequate room for concealing data in summaries. Seven, the presented bitrate is roughly from 0.064% up to 0.20% and with an approximate average of 0.12%. This bitrate is limited only to the current implementation example, which employs only one type of summarisation; namely, it is an extraction technique. Obviously, it is expected that it can be implemented differently than the presented implementation example to achieve better

overall results, as will be investigated in future work. The future direction for improving Sumstega implementation, in general, includes applying more summarisation techniques such as abstraction techniques.

References

Afantenos, S.D., Karkaletsis, V., Stamatopoulos, P. and Halatsis, C. (2007) 'Using synchronic and diachronic relations for summarizing multiple documents describing evolving events', *Journal of Intelligent Information Systems*.

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

Auto Summarizer, available at http://mskw.cipher-sys.com/Lectern/summary_submitter.asp (accessed on 25 December 2008).

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

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

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

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), Lecture Notes in Computer Science*, Springer, Malaga, Spain, Vol. 2200, pp.156–165.

Cremmins, E.T. (1996) *The Art of Abstracting*, 2nd ed., Information Resources Press, Arlington, VA.

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

Cvejic, N. and Seppanen, T. (2004b) 'Reduced distortion bit-modification for LSB audio steganography', in the *Proceedings of the 7th International Conference on Signal Processing (ICSP 04)*, Beijing, China, August, Vol. 3, pp.2318–2321.

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

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

Desoky, A. (2009b) 'Listega: list-based steganography methodology', *International Journal of Information Security*, Springer, April, Vol. 8, No. 4, pp.247–261.

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

Desoky, A. (2010a) 'Matlist: mature linguistic steganography methodology', *Journal of Security and Communication Networks*.

Desoky, A. (2010b) 'NORMALS: normal linguistic steganography methodology', *Journal of Information Hiding and Multimedia Signal Processing*, July, Vol. 1, No. 3, pp.145–171.

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

Desoky, A. (in press) 'Headstega: email-headers-based steganography methodology', *International Journal of Electronic Security and Digital Forensics*.

Desoky, A. (in process) *Noiseless Steganography: The Key to Covert Communications*, Information Security Publisher/Taylor and Francis Group, ISBN: 1439846219 and ISBN: 9781439846216.

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

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

Desoky, A. and Younis, M. (2009) 'Chestega: chess steganography methodology', *Journal of Security and Communication Networks*, March.

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

Ercan, G. and Cicekli, I. (2007) 'Using lexical chains for keyword extraction', *Inf. Process. Manage.*, Vol. 43, No. 6, pp.1705–1714.

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', in the *Proceedings of Information Hiding Workshop (IH 2005)*, Springer-Verlag, Barcelona, Spain, June, pp.213–233.

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

Hassel, M. and Dalianis, H. *SweSum – Automatic Text Summarizer*, available at http://swesum.nada.kth.se/index-eng-adv.html (accessed on 25 December 2007).

Hobson, S.P., Dorr, B.J., Monz, C. and Schwartz, R. (2007) 'Task-based evaluation of text summarization using relevance prediction', *Inf. Process. Manage.*, Vol. 43, No. 6, pp.1482–1499.

IBM Intelligent Miner (1999) Available at http://www.research.ibm.com/journal/sj/433/mack.html (accessed on 25 December 2007).

Inxight Software Incorporation (2000) *Inxight Summarizer*, available at http://www.inxight.com/products/sdks/sum (accessed on 25 December 2007).

Jones, K.S. (2007) 'Automatic summarising: the state of the art', *Inf. Process. Manage.*, Vol. 43, No. 6, pp.1449–1481.

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

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

Koumpis, K. and Renals, S. (2005) 'Automatic summarization of voicemail messages using lexical and prosodic features', *ACM Transactions on Speech and Language Processing*, February, Vol. 2, No. 1.

Leite, D.S., Rino, L.H.M., Pardo, T.A.S. and Nunes, M.d.G.V. (2007) 'Extractive automatic summarization: does more linguistic knowledge make a difference?', in *TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, pp.17–24, Association for Computational Linguistics, Rochester, New York, USA.

Li, W. (1992) 'Random texts exhibit Zipf's-law-like word frequency distribution', *IEEE Transactions on Information Theory*, Vol. 38, No. 6, pp.1842–1845.

Liang, S.F., Devlin, S. and Tait, J. (2007) 'Investigating sentence weighting components for automatic summarisation', *Inf. Process. Manage.*, January, Vol. 43, No. 1, pp.146–153.

LTRC, IIIT, *Automatic Text Summarizer*, available at http://search.iiit.net/~jags/summarizer/index.cgi (accessed on 26 December 2007).

Luhn, H.P. (1958) 'The automatic creation of literature abstracts', *IBM Journal of Research and Development*, Vol. 2, No. 2, pp.159–165.

Mana-Lopez, M.J., De Buenaga, M. and Gomez-Hidalgo, J.M. (2004) 'Multidocument summarization: an added value to clustering in interactive retrieval', *ACM Transactions on Information Systems*, April, Vol. 22, No. 2, pp.215–241.

Mani, I. (2001) *Automatic Summarization*, John Benjamins Publishing Company.

Mani, I. and Maybury, M.T. (1999) *Advances in Automatic Text Summarization*, MIT Press, Cambridge.

Marcu, D. (2000) *The Theory and Practice of Discourse Summarization and Parsing*, MIT Press, Cambridge.

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

New York Times (2008) Available at http://www.nytimes.com/2008/12/11/world/europe/11britain.html?_r=2&hp (accessed on 17 December 2008).

Nomoto, T. (2007) 'Discriminative sentence compression with conditional random fields', *Inf. Process. Manage.*, Vol. 43, No. 6, pp.1571–1587.

Okazaki, N., Matsuo, Y., Matsumura, N. and Ishizuka, M. (2003) 'Sentence extraction by spreading activation with refined similarity measure', *IEICE Transactions on Information and Systems (Special Issue on Text Processing for Information Access)*, Vol. E86-D, No. 9, pp.1687–1694.

ScramDisk (2008) *Free Hard Drive Encryption for Windows 95 and 98*, available at http://www.scramdisk.clara.net (accessed on 3 August 2008).

Sekine, S. and Nobata, C. (2003) 'A survey for multi-document summarization', in Dragomir R. Radev, Simone Teufel, Donna Harman and Paul Iver (Ed.,): *Proceedings of the HLT '03/NAACL '03 Workshop on Text Summarization*, May 31 – June 1, Association for Computational Linguistics, Edmonton, Canada.

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

SRA Corporation, *DimSum Summarizer*, available at http://sra.com (accessed on 26 December 2007).

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

TIME Magazine (2007) available at http://www.time.com/time/world/article/0,8599,1679108,00.html (accessed on 2 November 2007).

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*, January.

Wayner, P. (1992) 'Mimic functions', *Cryptologia*, Vol. XVI, No. 3, pp.193–214.

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

Yu, J., Reiter, E., Hunter, J. and Mellish, C. (2007) 'Choosing the content of textual summaries of large time-series data sets', *Natural Language Engineering*, Vol. 13, pp.25–49.

Zipf, G.K. (1968) *(Introduction by Miller, G.A.) The Psycho-Biology of Language: An Introduction to Dynamic Philology*, MIT Press, Cambridge, MA.

Notes

1 Preliminary and shorter version of this work appeared in Desoky et al. (2008c).

2 Microsoft, *AutoSummarize is built-in Microsoft Word* (in this paper version 97 used).