Jokestega: automatic joke generation-based steganography methodology

Abdelrahman Desoky

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

Abstract: This paper presents a novel steganography methodology, namely Automatic *Joke* Generation Based *Stega*nography Methodology (Jokestega), that pursues textual jokes in order to hide messages. Basically, Jokestega methodology takes advantage of recent advances in Automatic Jokes Generation (AJG) techniques to automate the generation of textual steganographic cover. In a corpus of jokes, one may judge a number of documents to be the same joke although letters, locations, and other details are different. Generally, joke and puns could be retold with totally different vocabulary, while still retaining their identities. Therefore, Jokestega pursues the common variations among jokes to conceal data. Furthermore, when someone is joking, anything may be said which legitimises the use of joke-based steganography. This makes employing textual jokes very attractive as steganographic carrier for camouflaging data. It is worth noting that Jokestega follows Nostega paradigm, which implies that joke-cover is noiseless. The validation results demonstrate the effectiveness of Jokestega.

Keywords: steganography; linguistic steganography; text steganography; information hiding; information security.

Reference to this paper should be made as follows: Desoky, A. (2012) 'Jokestega: automatic joke generation-based steganography methodology', *Int. J. Security and Networks*, Vol. 7, No. 3, pp.148–160.

Biographical notes: Abdelrahman Desoky is currently a CEO of The Academia Planet and an independent consultant, researchers and instructor for both academia and practice sectors. He received PhD from the University of Maryland and MSc from the George Washington University; both degrees are in Computer Engineering. He is a Scientist and Computer Engineering Doctorate with over 20 years experience in the computer field. He is an experienced educator at both the graduate and undergraduate level. He is an author and of Security book entitled *Noiseless Stenography: The Key of Covert Communications*.

1 Introduction

Linguistic steganography is the scientific art of avoiding suspicion in covert communications by concealing data in a textual cover. When using any steganographic technique if suspicion is raised, the goal of steganography is defeated regardless of whether or not a plaintext is revealed. Contemporary linguistic steganography approaches found in the literature are not fully capable of passing both computer and human examination (Desoky, 2010a). Such shortcoming is attributed to the fact that these approaches may introduce detectable flaws (noise), such as incorrect syntax, lexicon, rhetoric, grammar and the content of the linguistic-cover may be meaningless and semantically incoherent (Desoky, 2008; Desoky, 2010a; Desoky, 2010b; Desoky, 2011b). Obviously, these detectable flaws can raise suspicion during covert communications unless there is a legitimate excuse such as flaws made by a person with a speech or writing impediment. Not enough attention is given to these issues until Nostega paradigm and its methodologies were introduced (Desoky, 2008; Desoky, 2009a; Desoky, 2010a; Desoky, 2010c; Desoky, 2012).

This paper presents a novel steganography methodology that follows Nostega paradigm (Desoky, 2008; Desoky, 2009a; Desoky, 2010a; Desoky, 2012), namely Automatic *Joke* Generation-Based *Stegan*ography Methodology (Jokestega) that exploits textual jokes in order to conceal messages. Jokes are a type of wordplay that offers joy. Jokes can be in speech and text (Reiter and Dale, 2000; Manurung et al., 2004; O'Mara et al., 2004c; Friedland and Allan, 2008; Waller et al., 2009). When jokes are represented in text, the classic linguistic structures e.g. grammar, spelling, rules, etc. are unessential to be obeyed. This linguistic feature of textual jokes elevates the steganographic rooms when employing joke-based steganography. The tremendous use of jokes renders an adversary's job impractical to investigate the use of steganographic techniques in all textual jokes where emails and Internet nowadays are folded by jokes. Such common behaviour of human legitimises the communicating parties to

Copyright © 2012 Inderscience Enterprises Ltd.

establish a covert channel without any suspicious pattern to transmit hidden messages. This was the motive of developing Automatic Joke Generation-Based Steganography Methodology (Jokestega). Jokestega encodes a message then assigns it to steganographic carriers, such as an entire joke, a letter, a word or even non-linguistic elements and symbols. For example, a symbol of smiley face can be assigned a tow digits value as follows: '☺' = 00, ':-)' = 01, '☹' = 10, ':-(' = 11, etc. in order to camouflage data. Since such use is completely legitimate, Jokestega neither hides data in a noise (errors) nor produces noise while a message is concealed in a joke-cover (text-cover). In addition, the recent advances in the field of Automatic Joke Generation (AJG) ease the automation of textual joke-cover. This also makes it more attractive to be used in steganography. Jokestega is resilient against contemporary attacks including an attack by an adversary who knows Jokestega, i.e. Sumstega is a public methodology. Simply, an adversary cannot distinguish between an ordinary joke that contains no hidden data and a joke-cover that contains hidden data. Moreover, Jokestega resilient against distortion and comparison attacks.

The remainder of this paper is organised as follows. Section 2 briefly provides some background and related work discussion about both fields: Automatic-Joke Generation and linguistic steganography. Section 3 introduces the Jokestega methodology. Section 4 demonstrates implementation of Jokestega. Section 5 discusses steganalysis validation of Jokestega. Finally, Section 6 concludes the paper and highlights directions for future research.

2 Background and related work

This section presents a brief overview of the Automatic Joke Generation Systems (AJGS) field and a review of prior work on linguistic steganography that are related to Jokestega.

2.1 Automatic joke generation systems

A joke is a fabricated in a form of untrue short story or expression with a humorous twist (Reiter and Dale, 2000; Ritchie et al., 2007; Friedland and Allan, 2008; Manurung et al., 2008a; Manurung et al., 2008b). Jokes can be in many different forms, e.g. a short story, question and answers, etc. that make humorous. Jokes may employ mockeries, sarcasm, wordplay, etc. in such way that makes comic. Jokes are different from both regular slang linguistics and classic linguistics. Therefore, when jokes are represented in text, the structure linguistics of a particular language would not be obeyed while textual jokes still recognised as legitimate text because jokes have their own linguistics. The purpose of using jokes is to entertain friends, relatives, colleagues, audience, etc. Generally, the expected response is laughter. However, if this does not occur then the joke is a fallen-flat or bombed. The use of jokes is a part of human culture and is traced back to the ancient civilisations.

The field of AJGS has enjoyed significant advances in recent years and is still promising more in the near future (Reiter and Dale, 2000; Manurung et al., 2006b; Black et al., 2007; Friedland and Allan, 2008). AJGS employ a procedure that may be based on one or more of the following: knowledge base, artificial intelligence, computational linguistics, natural language generation and other related techniques to achieve its goal (Manurung et al., 2006a; Manurung et al., 2006b; O'Mara et al., 2006). Some famous examples of AJGS are JAPE and STANDUP. JAPE generates question-answer-based jokes and puns (Reiter and Dale, 2000; Joke Generator Project STANDUP, 2003; Friedland and Allan, 2008). This system JAPE named by abbreviating 'Joke Analysis and Production Engine'. The STANDUP is an improved system to generate jokes. STANDUP, jokes generators, is implemented in Java language (Manurung et al., 2004; O'Mara et al., 2004a; Waller et al., 2005; Ritchie et al., 2006). It was devoted for children with communication disabilities, e.g. because of cerebral palsy. This system, STANDUP, named by abbreviating 'System to Augment Non-speakers' Dialog Using Puns'.

Since the focus of this paper is linguistic steganography and it is not the field Automatic Jokes Generation (AJG) and also due to space constraints, the related work is covered in this paper is in a balance of linguistic steganography.

2.2 Linguistic steganography

Linguistic steganography approaches conceal data in a linguistic-based textual cover. Linguistic steganography approaches can be categorised as follows: Series of Characters and Words, Statistical Based, Word Replacement, Noise-Based and Nostega-Based. Series of Characters and Words approach is also known as null-cipher (Kahn, 1996), which was used by the Germans during World War I (WWI). Suspicion is raised because it violates the linguistic rules of a language used by forcing series of characters and words.

Statistical Based technique is known as mimic functions approach (Wayner, 1992; Wayner, 2002). Mimic functions, as the name suggests, attempts to imitate the statistical profile of normal text. The output of regular mimic functions is gibberish rendering it extremely suspicious (Grothoff et al., 2005a; Grothoff et al., 2005b; Stutsman et al., 2006; Desoky, 2010a). However, the combination of mimic functions and CFG slightly improved the readability of the text (Wayner, 1992; Wayner, 2002). Yet, the text-cover still contains numerous flaws such as incorrect syntax, lexicon, rhetoric and grammar. Furthermore, the content of the text-cover is often meaningless and semantically incoherent.

Word Replacement approach, is called NICETEXT or synonyms-based that, uses a big dictionary (Chapman and Davida, 1997; Chapman et al., 2001; Chapman and Davida, 2002). NICETEXT employs a piece of text to manipulate the process of embedding a data in a form of synonym substitutions. This process preserves the meaning of text-cover as its original text that contains no hidden message every time it is used. The synonyms-based approach attracted the attention of numerous researchers in the last decade (Desoky, 2010a). Although the text-cover of synonym-based approach may look legitimate from a linguistics point of view given the adequate accuracy of the chosen synonyms, reusing the same piece of text to hide a message is a steganographical concern. If an adversary intercepts the communications and oversees the same piece

of text that has the same meaning over and over again with just different group of synonyms between communicating parties, he will question such use.

Noise Based technique is simply hides data in the linguistic errors (noise). There are few approaches that are Noise Based, which are as follows: Translation-based scheme, Confusing Approach, SMS-based scheme. Translation-based steganographic scheme (Grothoff et al., 2005a; Grothoff et al., 2005b; Stutsman et al., 2006; Meng et al., 2011) hides a message in errors (noise) that are encountered in a Machine Translation (MT). Grothoff et al. stated that one of the concerns is that the continual improvement of machine translation may narrow the margin of hiding data (Grothoff et al., 2005; Stutsman et al., 2006). Furthermore, translation-based approach as confirmed by Grothoff et al. cannot be used with all languages due to huge differences in the essential linguistics structures (Grothoff et al., 2005; Stutsman et al., 2006). Confusing approach is a noise-based approach that employs typos and abbreviations in text of emails, blogs, forums and any other similar type of noisy text in order to hide messages (Topkara et al., 2007). Similarly, Shirali-Shahreza and Shirali-Shahreza (2007) presented an abbreviation-based scheme, which is known also as SMS-based approach that camouflages messages using Short Message Service (SMS) of mobile phones. Nonetheless, these techniques 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.

Recently, a novel paradigm in steganography research, namely Noiseless Steganography Paradigm (Nostega) has been introduced (Desoky, 2008; Desoky, 2009a), in which a message is hidden in a cover as data rather than noise. A group of methodologies have been developed based on the Nostega paradigm. First one of these methodologies is the Summarisation-Based Steganography Methodology (Sumstega) (Desoky et al., 2008). Sumstega exploits automatic summarisation techniques to camouflage data in the auto-generated summary-cover (text-cover) that looks like an ordinary and legitimate summary. The second linguistic steganographic scheme that is also based on Nostega paradigm is the List-Based Steganography Methodology (Listega) (Desoky, 2009b). Listega manipulates itemised data to conceal messages in a form of textual list. The third linguistic steganography methodology, Notes-Based Steganography Methodology (Notestega) (Desoky, 2009c) that takes advantage of the recent advances in automatic notetaking techniques to generate a text-cover. Notestega pursues the variations among both human notes and the outputs of automatic notetaking techniques to conceal data. The fourth linguistic steganography methodology, Mature Linguistic Steganography Methodology (Matlist) (Desoky, 2011b) 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 fifth linguistic steganography methodology, unlike all other approaches, the Normal Linguistic Steganography Methodology (NORMALS) neither generates noise nor uses noisy text to camouflage data. NORMALS 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 (Desoky, 2010b). As a result, NORMALS is capable of fooling both human and machine examinations. Unlike Matlist, NORMALS is capable of handling non-random series domains. The sixth linguistic steganography methodology is the Educational-Centric Steganography Methodology (Edustega). Noiselessly, Edustega methodology exploits educational documents such questions and answers of exams, examples, puzzles, competitions to camouflage data.

It is worth noting that the presented Jokestega methodology in this paper follows this new paradigm. However, it is unlike all other techniques, Jokestega exploits jokes by taking advantage of the recent advances in the field of Automatic Joke Generation (AJG) to camouflage data without introducing suspicious pattern.

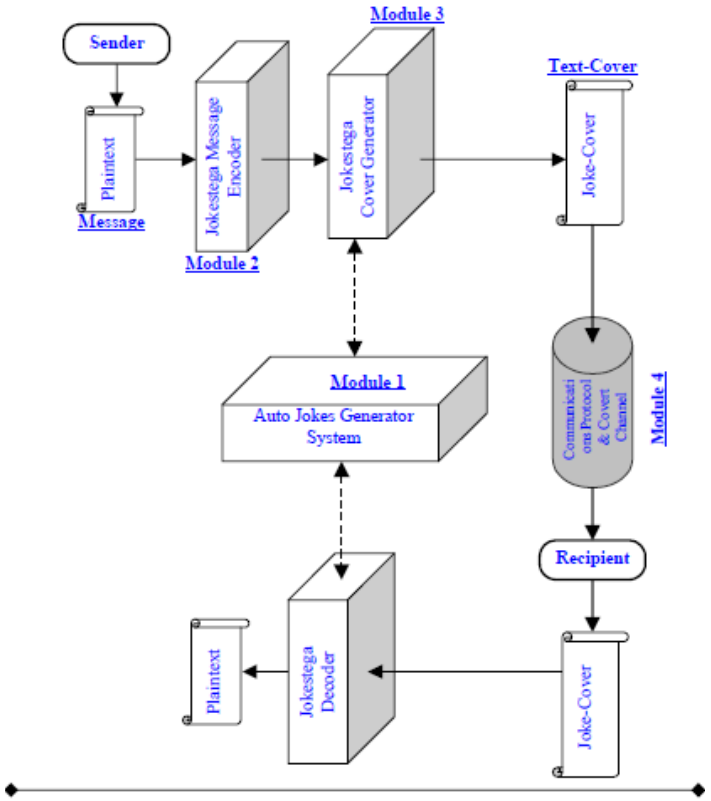
3 3 Jokestega methodology

Jokes in general are very popular and widely used all over the world by people of all ages. The excessive of jokes by a wide variety of people creates a high volume of traffic, which makes an adversary's job impractical to investigate all of them. Such huge traffic, due to the normal frequent exchange of jokes in all kind of formats e.g. electronic, printed formats, audio, etc. allows communicating parties to establish a covert channel to covertly transmit hidden messages. Yet, the textual jokes not only is capable of concealing messages but also it has an adequate room for concealing data, rendering joke-cover (text-cover) to retain superior bit-rate to contemporary linguistic steganography approaches, as will be shown later. It is also can be applied to all languages. Consequently, textual jokes are an attractive steganographic carrier. Therefore, the novel Automatic Joke Generation Based Steganography Methodology (Jokestega), which is presented in this paper, takes advantages of recent advances in the field of Automatic Joke Generation (AJG) to securely communicate covertly. Jokestega is based on Nostega paradigm (Desoky, 2008; Desoky, 2009a), which implies that it neither hides data in a noise (errors) nor produces noise. Instead, it camouflages data in the textual jokes by manipulating, mainly but not limited to words, letters, non-linguistics elements (e.g. symbols), etc., in order to embed data without generating any suspicious pattern.

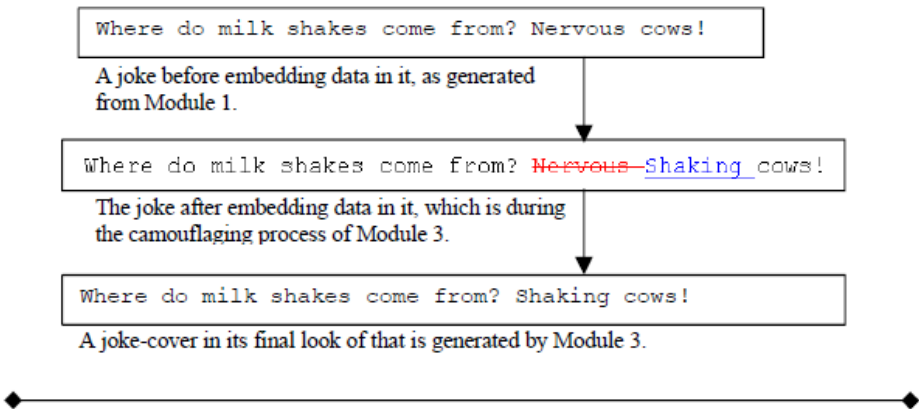
The fundamental algorithm of Jokestega system is consisted of four modules: the Automatic Jokes Generator System, Message Encoder, Camouflager and Covert Channel. These modules are ultimate goal is to define a Jokestega system configuration for the communicating parties to use. The four modules are highlighted as follows. First, the communicating parties opt to either implement Automatic Jokes Generator System or use one that is already built. Second, a Message Encoder that is capable of encoding a message in an appropriate form for the camouflaging process (Module 3) is implemented.

In this paper for the simplicity, encoding a message may be done by converting a message into binary representation of its ASCII code and slicing it to a particular length of digits e.g. 3, 4, 5, etc. For instance, a message after encoding it may look like the following: 00011, 10101, 00000, 11110, 11111 and so on. Note that any other techniques of encoding may be used and the employed encoding technique in this module is just an example to make it simple for illustration. Third, a Camouflager is the Module 3 that conceals messages in the generated textual jokes (original text that contains no hidden data) by embedding data that represent the required message to be camouflaged. This process can be accomplished by numerous ways. However, the chosen technique in this paper for embedding data in textual jokes is that it pursues the common variations among the outputs of AJGS to conceal data. Examples of ordinary variations among textual jokes may include: different text of same joke, altering some words without changing the core of a joke, taking advantage of using non-classic linguistic rules of writing, etc. Fourth, communicating parties opt to establish a Covert Channel (Module 4) which is the means for hidden delivery of steganographic cover. These modules are elaborated in the following sections.

Figure 1 (A) An illustration of the interaction of the various Jokestega modules and how the outputs of the individual modules are used for covert communications between two parties; (B) Illustrates the virtual stages of generating joke-cover (see online version for colours)



(A)



(B)

3.1 Automatic jokes generator system (Module 1)

The aim of this section is to discuss the Jokes Generator System (Module 1). AJGS produces an original text (textual jokes) that contains no hidden messages. The output of this module will be generated by the request of the Camouflager (Module 3), which is capable of embedding the steganographic code (the encoded message) in the generated text by the AJGS (Module 1), in order to conceal a message. The output of this procedure is in a form of legitimate jokes. Obviously, the fact about textual jokes is known to everyone that it is untrue, funny and legitimate. From a steganography point of view, an untrue content of text may raise suspicion. However, when using textual jokes, this is not a concern because it is legitimate the use of untrue information when someone is joking. Furthermore, reusing or altering an existing text to hide data is not a recommended practice since an adversary can reference the original text and detect the differences. In addition, the reuse of same piece of text more than once may increase vulnerability of the covert communications. If an adversary intercepts the communications and oversees a similar piece of text being exchanged between communicating parties over and over again, suspicion may be raised because the adversary will wonder of such use. However, this is not a concern for Jokestega because reusing and modifying such text (textual jokes) are common practices because it is not a serous text like such as medical or court documents. Such Jokestega's strong feature eases the automation of a joke-cover (steganographic textual cover). In addition, it is a trivial task that communicating parties to use contemporary AJGS, as demonstrated in Section 4. Examples of AJGS include:

- MIT Project, Chuck Norris Joke Generator.[1]
- Jokes2000.[2]
- The Joke Generator dot Com.[3]
- Online Joke Generator System (pickuplinegen).[3]

3.2 Message encoder (Module 2)

Implementing a steganographic Message Encoder (ME Module 2), the message encoder, follows a two-steps process: first, determining the encoding parameters in the topic picked by Module 1; second, defining a steganographic coding based on these parameters. A parameter in this context means some aspects of textual joke(s) that can be referred to steganographical values throughout a joke-cover (text-cover). In textual jokes, the common variations among the outputs of AJGS can be used to conceal data. Examples, different text of same joke, altering some words without changing the core of a joke, taking advantage of using non-classic linguistic rules of writing, inserting symbols, etc. can be exploited for camouflaging data. The definition of the steganographic code would depend on the selected parameters. For example, encoding a message using symbols (e.g.: '☺' = 00, ':-)' = 01, '☺' = 10, ':-(' = 11, etc.) is different from encoding it using the order in which the various jokes appear and so on. The coding module of Jokestega exploits these options and determines the parameter(s) that will be employed for concealing data. The selection criteria may be driven by the size of the message, the popular joke styles, the availability of existing jokes and any other factors.

Jokestega does not impose any constraint on the message encoding scheme (ME Module 2) as long as it generates a set of data values that can be embedded in a joke-cover. Given the availability of numerous encoding techniques in the literature that fit (Desoky and Younis, 2008; Desoky, 2009a; Desoky and Younis, 2009; Desoky, 2010a; Desoky, 2012), the balance of this section will focus on an example that will be used in Section 4 to demonstrate the applicability of Jokestega. In the example, the encoding is done as follows. A message is first converted to a binary string. The string can be a binary of cipher text or a compressed representation. The binary string is then partitioned into groups of m bits. The value of m is determined based on the encoding parameters that Jokestega exploits. For instance, if the joke-cover will be in a form of group of jokes, the binary message is partitioned it into groups of two bits, e.g., '0000', '0001', '0010', '0011' and so on up to '1111' corresponding to the possible choices. Again, this encoding scheme is just for illustration and many alternatives and more sophisticated schemes can be employed, as stated earlier and demonstrated in Section 4.

3.3 Camouflager (Module 3)

The aim of this section is to discuss and describe the Camouflager (Module 3). Once a message is encoded using Module 2 (the Message Encoder); a Jokes Generator System (Module 1) first produces an original text (textual jokes) that contains no hidden messages and then the Camouflager (Module 3) embeds the steganographic code (encoded message), in order to conceal a message. The output of this procedure is in a form of legitimate textual jokes such as demonstrated in the implementation section. Such text can be in a form of family jokes, adult jokes, academic jokes, etc. in order to embed data without generating any suspicious pattern. Note that the modules in this paper may be implemented differently with a different sequence too.

As stated earlier, reusing or altering an existing text to hide data is not a recommended practice, from a steganography point of view, because an adversary may reference the original text and detect the differences, which easily may raise suspicious. In addition, the reuse of same piece of text more than once may increase vulnerability of the covert communications. If an adversary intercepts the communications and oversees a similar piece of text being exchanged between communicating parties over and over again, suspicion may be raised because the adversary will wonder of such use. However, this is not a concern with Jokestega because reusing and modifying jokes are common practices and after all it is just joking for fun and nothing is serious. For example, the common variations among the outputs of AJGS or imposing natural alterations can be used to conceal data by text substitution procedure without generating detectable noise avoiding raising suspicion. Some examples of jokes that are generated By AJGS, which is called 'Standup' software as show in Table 1.

Table 1 It illustrates virtual variations in blue that can be used to camouflage data

No	*Original textual jokes contain no hidden data by standup system*		*Embedding data may use*
1	What do you get when you cross a car with a sandwich?	A traffic jam➔	Subway is faster or subway eat fresh
2	What do you call a strange rabbit?	A funny bunny➔	Rob it, honey bunny, happy bunny
3	What do you call a frog road?	A main toad➔	A fake road
4	What do you call artist who is a minister?	A pastor master➔	A pastor toaster

In a corpus of jokes, one may judge a number of documents to be the same joke although letters, locations and other details are different (Reiter and Dale, 2000; Joke Generator Project STANDUP, 2003; Manurung et al., 2006a; Friedland and Allan, 2008). Generally, joke and puns could be retold with totally different vocabulary, while still retaining their identities. Linguistically, there are published systems that their task is identifying the 'same joke', for more information refer to (Ritchie et al., 2007; Friedland and Allan, 2008; Manurung et al., 2008b; Waller et al., 2009). Such features are capable of concealing data without raising suspicion.

In regard of message size, concealing long messages is generally a challenge for most known steganography approaches. Jokestega can hide long messages by simply employing more jokes by splitting the message over multiple jokes in a joke-cover. When someone tells a joke, it initiates and legitimises the others to joke-back. Joking-back is a common behaviour of human, which allows the communicating parties to repeatedly transmit joke-covers that conceal hidden messages back-and-force using Jokestega.

3.4 Establishing covert channel (Module 4)

Jokestega naturally camouflages the delivery of a hidden message in a way that makes it appear legitimate and innocent. To employ Jokestega, the communicating parties first need to define and agree on the basic configuration of the covert channel. This step includes determining the following: (a) a legitimate relationships among or between the communicating parties that justify their interaction wit each other and (b) how the cover will be delivered from a sender to the recipient. Plotting a suitable scenario can play an essential role for securing the steganographic communications by establishing an appropriate covert channel for delivering a hidden message. The chosen scenario must facilitate the process of embedding data without generating noise in order to achieve the steganographical goal. Since Jokestega mainly manipulates jokes to camouflage messages, any scenario such as relationships among or between the communicating parties (e.g. colleagues of a particular profession) that allows the employing of jokes can be used. The second important configuration parameter is 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. The fact that Jokestega employs noiseless-based means for hiding data enables great flexibility in delivering the steganographic cover to its recipient. Options may include web post and download, email transmission, etc. A sender may mix a joke-cover among other legitimate documents; obviously, the basic configuration of the covert channel should include how a recipient can only decode the right covers. For instance, the communicating parties may agree on putting joke-covers among others similar documents by designating a particular sequence, such as odd number, even number, every other 3 or any other order. The core of covert channels is how to prevent the association between a sender and recipient from drawing suspicion and to render it innocent communications. For example, exchanging emails would automatically imply a relationship between the communicating parties. Similarly, downloading files from a web site indicates an interest in the accessed material. Due to the 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 the transmitted materials. For example, if a profession for one of the communicating party is an elementary English teacher and yet he sends or receives college level chemistry exams, then suspicion will likely be raised. Therefore, it is very important to rationalise the exchange of steganographic cover in order to avoid attracting any attention that may trigger an attack. The communicating parties need to agree on how to justify their interest in the education documents of the selected topic. This may include defining a role, such as mentoring or tutoring that a sender plays, a profession or simple an interest that justify a peer relationship.

4 Jokestega implementation

The aim of this section is to demonstrate possible implementation example to show how Jokestega methodology can be used. It is worth noting that his section shows just few examples of possible implementations following the steps outlined in the previous section. Jokestega implementation is detailed in following subsections as follows.

4.1 Jokestega system

This section explains an implementation example of how Jokestega modules are employed and configured to construct the overall Jokestega system used in this paper by the communicating parties.

AJGS and Camouflager (Modules 1 and 3): The reason that these modules (Modules 1 and 3) are described in the same section is because the fact that both modules are highly interrelated to each other. The AJGS Module 1 produces an original text that contains no hidden messages. Then, the Jokestega Camouflager (Module 3) embeds the steganographic code (encoded message), that is generated by the Message Encoder (ME Module 2), in the generated joke by the AJGS (Module 1), in order to conceal a message. The output of this procedure is in a form of legitimate and ordinary jokes. Such text embeds data without generating any suspicious pattern is capable of fooling an adversary. Nonetheless, in this Jokestega

configuration example, Jokestega Camouflager module employs online AJGS (Binsted, 1996; Binsted et al., 1997; Reiter and Dale, 2000; O'Mara and Waller, 2003; Friedland and Allan, 2008), online samples,[1–3] online dictionaries,[4–6] and Microsoft Thesaurus (built-in Microsoft Word 97)[7] to embed the data and generate the joke-cover. The dictionaries and thesaurus are mainly exploited in order to pick appropriate vocabulary for the choices for a joke. Since one the options is to conceal data in a keyword of a joke, the first letter of the keyword in a given joke would conceal data in a length of 4 bits such as values from '0000' up to '1111' using the letters from 'A' to 'Z' (Desoky and Younis, 2008). For instance, in the first time using Table 1 if the steganographic value that needs to be embedded is '0000' then the correct keyword of a joke will be started by the letter 'A'. Classified the jokes by joke's keywords (e.g. Vampire, Teacher, Hamburger, etc.) may ease both process of hiding and revealing. However, if the steganographic value that needs to be embedded is equal '0011' then the correct keyword of a joke will be started by the letter 'D' and so on. Furthermore, the use of first letters by Jokestega system does not impose constraints on the employed vocabulary. Based on this Jokestega configuration each joke may conceal at least four bits. Obviously, more bits of data may be concealed by simply embedding symbols such as the common symbols that are used by users nowadays (e.g.: '☺', ':-)', '☹', ':-(', etc.). Such symbols can conceal data the length of bits would depend on the maximum number of symbols used. For example, if the maximum number of symbols is 16 then the maximum length of bits is 4 bits which is from '0000' up to '1111'. On the other hand, if the maximum number of symbols is 64 then the maximum length of bits is 7 bits which is from '0000000' up to '1111111' and so on.

Jokestega Encoder (Module 2): It is worth noting that the focus of this paper is in the balance for showing how Jokestega achieves the steganographical goal rather than making it difficult for an adversary to decode an encoded message. Employing a hard encoding system or cryptosystem to increase the protection of a message is obviously recommended and straightforward using any contemporary encoder or cryptosystem. Similarly, employing compression to boost the bit-rate can easily be accomplished by using the contemporary techniques in the literature. Nonetheless, Jokestega encodes a message in a form that suits the camouflaging process. Encoding a message may be converted into binary representation of its ASCII code and slicing it to a particular length of digits e.g. 3, 4, 5, etc. For instance, a message after encoding it may look like the following: 00011, 10101, 00000, 11110, 11111 and so on. In this implementation example of this paper, a message is converted into binary representation of its ASCII code and slicing it to a length of 4 digits. This will be assigned to the first letter of joke's keyword according to Table 2. For example, when the joke's keyword starts with the letter 'B', it is concluded that the joke conceals '0001' as shown in Table 2. However, when the joke's keyword starts with the letter 'C', it is concluded that the question conceals '0010' and son as shown in Table 2.

To increase the resilience to attacks, Jokestega uses some randomness to the steganographical coding through the use of a combinatorics operations to define the mapping of symbols to bit strings. The steganographical code in this Jokestega configuration works as follows. Based on a predetermined protocol, the presented implementation example of Jokestega system in this section adds counter value like an index value 'i' to each steganographic bit string (e.g. 0000 + i, 0001 + i, 0101 + i, etc). To emphasise, it adds value of '0' 1st time, '1' 2nd time, '2' 3rd time and so on. To illustrate, when using Table 2, in order to conceal data in a joke's keyword starts by the letter 'A' implies '0000' 1st time, '0001' 2nd time used and so on. The auto receiver to reveal the hidden message will check the joke's keyword against Table 1 and it is index value 'i' (is it 1st time, '1' 2nd time, '2' 3rd time and so on) to find out their steganographic code values. The use of these tables is illustrated later in this section. Again, the encoding used in this paper is just an example for simple implementation example to make it easy for the reader to follow and understand. However, more sophisticated encoding techniques can be used.

Table 2 The steganographic code for camouflaging four bits in the joke's keyword

Steganographic code values	
Binary values	*First letter of the joke's keyword*
0000	A
0001	B
0010	C
0011	D
0100	E
0101	F
0110	G
0111	H
1000	I
1001	J
1010	K
1011	L
1100	M
1101	N
1110	O
1111	P
0000	Q
0001	R
0010	S
0011	T
0100	U
0101	V
0110	W
0111	X
1000	Y
1001	Z

The Covert Channel (Module 4): As indicated earlier, the configuration of the covert channel includes the convincing scenario of the relationship between the sender and receiver and how a joke-cover can be delivered. A scenario that makes it easy for the communicating parties to legitimise their communications in order to deliver joke-cover that contains hidden messages as shown in this paper.

4.2 Joke-cover example

This section shows few examples for how the Jokestega configuration discussed above can be used by the communicating parties to conceal messages. Therefore, the following describes how a message is encoded and processed by the Jokestega system prior to generating the joke-cover. The following shows joke-cover samples that conceal data are demonstrated.

- The plaintext is: 'Stop'.
- The Jokestega Encoder converts the message to a concatenated binary string using the ASCII representation of the individual characters, as follows: '01010011011101000110111101110000'.
- The encoder will then divide the above binary message into slices of sizes that matches those supported by the steganographic coding. The result is shown as follows: '0101 0011 0111 0100 0110 1111 0111 0000'. It should be noted that the binary string could have been encrypted or compressed prior to this step.
- Jokestega Camouflager then will generate the joke-cover (text-cover) that conceals the binary of a message. For the simplicity for the reader, a joke-cover is generated by selecting a group of jokes that contains the joke's keywords starts by first letters according to Table 2 as shown in Table 3.
- As shown in Figure 2, a sample of virtual joke-cover that contains 8 jokes, which conceals 32 bits using Table 2.
- As shown in Figure 3, a sample of virtual joke-cover contains 5 jokes that conceal 32 bits by using both joke's keywords and symbols, using Table 2 and 4. This is also concluded in Table 5.
- Another plaintext example is: 'war'.
- The Jokestega Encoder converts the message to a concatenated binary string using the ASCII representation of the individual characters, as follows: '011101110110000101110010'.
- The encoder will then divide the above binary message into slices of sizes that matches those supported by the steganographic coding. The result is shown as follows: '0111 0111 0110 0001 0111 0010'. It should be noted that the binary string could have been encrypted or compressed prior to this step.
- As shown in Figures 4 and 5, a sample of virtual joke-cover contains 4 jokes that conceal 24 bits by using jokes semantic embedding and substitutions. This illustrates the transformation from Figure 4 to Figure 5, using Table 2.

Obviously, a joke-cover can be sent as group of jokes, can be among other text in emails, can posted in a particular website blog or anyway that the communicating parties would agree up on it as a legitimate scenario for transmitting hidden messages.

Figure 2 Shows virtual joke-cover contains eight jokes that conceal 32 bits

- Where is Dracula's American office? The Vampire State Building.
- Teacher: When do astronauts eat? Pupil: At launch time!
- Can a hamburger marry a hot dog? Only if they have a very frank relationship!
- I'm not ugly. I could marry anyone I pleased! But that's the problem - you don't please anyone.
- Have you seen www.square.com? No, I haven't got around to it.
- What do you call 4 blondes laying on the beach? A: Public access.
- Why did the little pig hide the soap? He heard the farmer yell, "Hogwash!"
- Why is the moon like a dollar? It has four quarters.

Figure 3 Shows virtual joke-cover contains six jokes that conceal 32 bits by using symbols

- Where is Dracula's American office? The Vampire State Building. ☺
- Teacher: When do astronauts eat? Pupil: At launch time! :-)
- Can a hamburger marry a hot dog? Only if they have a very frank relationship! :0)
- I'm not ugly. I could marry anyone I pleased! But that's the problem - you don't please anyone. :-) ☺
- Have you seen www.square.com? No, I haven't got around to it. ☺

Figure 4 Shows virtual joke-cover contains four jokes that conceal 24 bits by using joke-substitutions (see online version for colours)

- Where is Dracula's American house ~~office~~? The Vampire State Building.
- Teacher: When do astronauts eat? Pupil: Hmm At launch time!
- ~~Can~~ Would a hamburger marry a hot dog? Only if they ~~have~~retain a very frank relationship!
- I'm not ~~ugly~~ hideous. I ~~could~~ can marry anyone I pleased! But that's the problem - you don't please anyone.

Figure 5 Shows the final look of virtual joke-cover, from Figure 4, contains four jokes that conceal 24 bits

- Where is Dracula's American house? The Vampire State Building.
- Teacher: When do astronauts eat? Pupil: At hunger time!
- Would a hamburger marry a hot dog? Only if they retain a very frank relationship!
- I'm not homely. I can marry anyone I pleased! But that's the problem - you don't please anyone.

Table 3 Encoded message using first letter of joke's keyword

Binary	*Letter*	*Joke's keyword*
0101	F or V	Vampire
0011	D or T	Teacher
0111	H or X	hamburger
0100	E or U	ugly
0110	G or W	www.square.com
1111	P	Public
0111	H or X	Hogwash
0000	A or Q	quarters

Table 4 Steganographic code values of symbols

Binarry	*Symbols*	
00	☺	☹
01	:0)	:0(
10	:0))	:0((
11	:-) or !	:-(or !

Table 5 Encoded message using first letter of joke's keyword and symbols

Binary	*Letter*	*Joke's Keywords*	*Binary*	*Symbols*
0101	F or V	Vampire	00	☺
1101	D or T	Teacher	11	:-)
0100	H or X	hamburger	01	:0)
1011	E or U	ugly	11	:-)
0111	G or W	www.square.com	00	☺
			00	☺

4.3 Jokestega performance

The bit-rate for this implementation example may achieve up to 8 bits per a short joke. In this paper, a short joke can be in a length of: sentence, question and answer, expressions (e.g. funny bunny) and so on. This imply that the bit-rate of Jokestega would vary from a particular joke to another because would depend on the size of jokes used and from one implementation to another. In this paper, the achieved bit-rate of the implementation example is accomplished by encoding one keyword per a joke, common non-linguistic elements (e.g. symbols) or minor alterations of few words (e.g. one or two words) as natural version of a joke. Obviously, the more steganographic careers are employed, the higher bit-rate will definitely be achieved. In regard of message size, the size of a message is a concern for most if not all steganography approaches. However, in the presented Jokestega scheme, Jokestega camouflages a long message. When a message is long, then Jokestega generates a longer text-cover (joke-cover). Simply, Jokestega distributes the required message to be camouflaged in a group of jokes using either single or multiple transmission(s), post them somewhere or whatever a predetermined scenario is. Nonetheless, the following shows some examples of Jokestega carriers to conceal data and legitimise the use of steganographic technique.

- *Joking behaviour:* Obviously, the joking behaviours are intrinsic in human. It is something like it is inborn in all people regardless of their nations. Note that joking is a normal behaviour where someone doing or telling something totally untrue and funny. Such behaviour legitimises the use of steganography where the need of fabricated text (untrue text) may be essential to conceal data.
- *Embedding non-linguistic elements:* A use of non-linguistic elements such as symbols are popular in textual jokes such as: ☺, ☹, !, :), :-), etc., which legitimately allows a steganographic system to conceal data by embedding such symbols.
- *Rhyme-substitution-based:* In jokes, the use of rhyme words is a common practice. Examples, funny bunny can be cutie bunny, sweetie bunny, honey bunny and so on. The joking behaviours of human can legitimise such use of rhyming that can be used to conceal data. In jokes, the use of rhyme-substitution-based steganographic technique is fully different from other technique such as the use of synonym-substitution-based technique. The rhyme-substitution-based steganographic technique does not attempt to preserve same linguistic meaning like in synonym-substitution-based technique.
- *Antonyms-substitution-based:* Joke is a joke! In other words, textual jokes are not a serious text and just for fun. This legitimises the use of antonyms-substitution-based steganographic technique. For example, the use of 'fatty skinny', 'tall short', 'hot cold', 'intelligent stupid', 'fast slow' and so on. Note that some of antonyms may be also rhyme, which makes sound funny as the goal of jokes. For instance, 'happy bunny' can be 'unhappy bunny', 'fatty' can be 'skinny' and so on. Such antonyms that are rhyme give more jokingly attitude, which legitimises their use.
- *Meaning-substitution-based:* When a joke is retold, someone may tell the meaning of a joke using deferent vocabulary and text. This legitimises the employing of meaning-substitution to camouflaging data.
- *Other common steganographic carriers:* Examples, Text-substitution-Based, Semantic-Substitution-Based and Synonyms-Substitution-Based can also be used while the accuracy of the substitution will be a serious concern due to the fact that someone just joking.

Comparing the bit-rates as shown in Table 6, it is obvious that Jokestega methodology achieves much more superior bit-rate than all comparable approaches, making it a very effective steganography system. The high bit-rate also enables the use of reasonable cover sizes, which it is a major concern for all steganography approaches whether it is linguistic or non-linguistic technique.

Table 6 The bitrate of contemporary text steganography approaches of the 'non Nostega-based' versus Jokestega (the presented system in this paper)

Approach	*Bit-rate*	*Comment*
Mimic functions (Wayner, 1992; Wayner, 2002)	0.90%	Based on 30 samples generated at www.spamimc.com
NICETEXT (Chapman et al., 1997; Chapman and Davida, 2002)	0.29%	Based on the samples in the cited papers
Winstein (1999)	0.5%	Based on the samples in the cited papers, and also confirmed by Murphy and Vogel (2007)
Murphy and Vogel (2007)	0.30%	Average per sentence (as reported by Murphy and Vogel (2007))
Nakagawa et al. (2001)	0.12%	As reported by Nakagawa et al. (2001), Bitrate achieved in real application is only 0.034%
Translation-based (Stutsman et al., 2006)	0.33%	Noted by the authors in the cited papers
Confusing (Topkara et al., 2007)	0.35%	Based on the samples in the cited papers
Jokestega (the presented system)	1–2%	Based on the presented implementation examples that conceals up to 8 bits per a short joke.

5 Steganalysis validation

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

5.1 Traffic attack

One of the possible attacks an adversary may pursue is to analyse the communications traffic and the access patterns to publicly available or exchanged documents, images, graphs, files, etc. For example, the intelligence community has a number of tools at their disposal for analysing traffic on the internet, tracking access to web sites, monitoring checked out literature from public libraries, etc. The main goal of a traffic attack is to detect unusual or questionable association between a sender and a recipient. Traffic analysis intuitively can identify who communicates with whom. The relationship between the communicating parties will be then qualified based on the contents of the message. Traffic attacks can be a threat for most contemporary steganographic techniques regardless of the steganographic cover type (e.g. image, graph, audio file, text, etc.). In the context of Jokestega, the domain of the cover is checked rather than its validity and the consistency of its contents. Then, if someone sends, receives and accesses some materials without a legitimate reason for doing so, e.g. a History teacher sending a Physics assignment to one of his students, suspicion can be raised and further investigation may be warranted. The additional investigations will involve a thorough analysis of a steganographic cover, as detailed in the next subsections. Traffic analysis is deemed ineffective with Jokestega. Jokestega camouflages the transmittal of a hidden message (joke-cover) to appear legitimate and thus suspicion is averted. Basically, Jokestega ensures that the involved parties establish a covert channel by having a well-plotted relationship with each other. Such scenario appears innocent and convinces an adversary of its legitimacy. Therefore, analysing the traffic between such communicating parties will not reveal any questionable association and will not trigger any further investigation. In addition, the high demand for joking e.g. making up jokes, telling and retelling, joking blogs and websites, etc., by a wide variety of people world wide, creates a high volume of traffic that makes it impractical for an adversary to investigate all traffics of jokes. Such voluminous traffic allows the communicating parties to establish a covert channel in order to transmit a joke-cover without drawing attention, rendering jokes an attractive steganographic carrier. Finally, it is noted that if further investigations on a joke-cover were triggered by traffic analysis, they would not be successful, as elaborated next. In Jokestega, differentiating between a joke-cover that contains a hidden message and another peer jokes without a hidden message is impossible.

5.2 Contrast and comparison attacks

One of the intuitive sources of noise that may alert an adversary is the presence of contradictions in a joke-cover. Examples of these contradictions include finding unusual repetition of a joke or a set of jokes, uncommon alteration of jokes, excessive naive jokes, etc. Also, if a joke-cover is not expected because of the profile of communicating parties (e.g. relationship of professor and student) and if there is, it is not expected to be numerous. Such contradictions would surely raise suspicion about the existence of a hidden message, especially when the type of jokes is presented repeatedly. The vulnerability of Jokestega to contrast attacks is generally very limited and depends on how the cover is generated. Automating the generation of a joke-cover through the use of contemporary AJGS makes the cover very resilient to this type of attacks. As demonstrated in Section 4, the use of such system[2, 3] allows the selection of appropriate jokes that not only match the encoded messages but also ensure the clarity of the wording and the suitability of the joke-cover.

From a steganography point of view, reusing or altering an existing text to hide data is not a recommended practice since an adversary can reference the original text and detect the differences. In addition, the reuse of same piece of text more than once may increase vulnerability of the covert communications. If an adversary intercepts the communications and oversees a similar piece of text being exchanged between communicating parties over and over again, suspicion may be raised because the adversary will wonder of such use. However, this is not a concern for Jokestega because reusing and modifying jokes are common human behaviour. For example, in a corpus of jokes, one may judge a number of documents to be the same joke although letters, locations and other details are different. Generally, joke and puns could be retold with totally different vocabulary, while still retaining their identities, as natural human behaviour. Therefore, Jokestega pursues the common variations among jokes to conceal data. Note that when someone is joking, anything may be said which legitimises the use of joke-based steganography. This makes employing textual jokes very attractive as steganographic carrier for camouflaging data and allows communicating parties to establish a covert channel without raising suspicion. Such Jokestega's strong feature eases the automation of a joke-cover. In addition, it is a trivial task that communicating parties to establish covert channel and well-plotted scenario to employ Jokestega methodology, as demonstrated in Section 3 and 4. Meanwhile, noise in the context of comparison attacks reflects alteration of authenticated or previously used documents. The goal of the adversary is to find any incorrect and inconsistent data that may imply the manipulation of contents of a joke-cover in order to embed a hidden message. As stated above, since reusing and modifying jokes are common practices, comparison and contrast attacks are deemed ineffective.

5.3 Linguistics attacks

A joke is a fabricated (untrue) text in a form of short story, question and answers, expression, etc., with a humorous twist (Binsted, 1996; Binsted et al., 1997; O'Mara and Waller, 2003; O'Mara et al., 2004b; O'Mara et al., 2004c; Friedland and Allan, 2008). Therefore, when textual jokes are fabricated to conceal data, suspicion-based fabrication (untrue text) can not be used as evidence about the existence of hidden messages. This is because such text by its nature supposes to be untrue text and will remain legitimate. Therefore, an adversary cannot claim or suspect an existence of hidden data because a text is untrue (fabrication). Linguistically, jokes have their own unique linguistics e.g. structures, rules, etc. To emphasise, the fabrication of such linguistics would not raise suspicion because the fair attack is comparing the linguistic of textual jokes has to be with its comparable of textual jokes. Therefore, when textual jokes are represented in a particular language, the classic linguistics of that language may not be obeyed while textual jokes still recognised as legitimate text because jokes have their own linguistics. This is one of the factors that eases a steganographic procedure to camouflage data in jokes. Obviously, the purpose of using jokes is to entertain friends, relatives, colleagues, audience, etc., and nothing is serious, which supports the use of jokes in steganography. Generally, the expected response is laughter. However, if this does not occur then the joke is a fallen-flat or bombed.

Linguistics examination distinguishes the text that is under attack from normal human language. Distinguishing the text from normal human language can be done through the examination of meaning, syntax, lexicon, rhetoric, semantic, coherence and any other feature that can help in detecting or suspecting the existence of a hidden message. These examinations are used to determine whether or not the text that is under attack is abnormal. Generally, the text used in jokes is normal. In addition, the produced text by AJGS usually meets the expected properties of a normal human language because it is initially generated by human and any alteration is done is more of cosmetic that is similar to natural joke variations. For example, changing words such as 'funny bunny' to 'happy bunny', question start by 'can' to 'would', inserting non-linguistic elements (e.g. '!', ':-)', ':-))'',:0)', '□', '□', etc.). Such steganographic carriers will not generate any noise (linguistic flaws), as demonstrated in Section 3 and 4. As a result, the generated cover as demonstrated in the implementation section is normal text. Furthermore, if there are errors in the AJGS engine, it should not be a concern for two reasons; first, it applies to all the generated text with and without a hidden message; second, nothing is concealed in errors. In addition, an engine error of AJGS is most likely fixable. Therefore, Jokestega is capable of passing any linguistic attack by both human and machine examinations.

5.4 Statistical signature

A statistical attack refers to tracking the profile of the used text. A statistical signature (profile) of a text refers to the frequency distribution of words and characters used. An adversary may use the statistical profile of a particular topic of jokes that contains no hidden message and compare it to a statistical profile of the suspected joke-cover to detect any differences. An alteration in the statistical signature of a particular topic of jokes may be a possible way of detecting a noise that an adversary would watch for. Unlike image steganography, tracking statistical signatures is an ineffective means for attacking linguistic steganography (Grothoff et al., 2005a; Grothoff et al., 2005b; Stutsman et al., 2006). Nonetheless, Jokestega is resistant to statistical attacks because it is simply opt to use legitimate text that is generated naturally by human. In addition, the generated textual cover (joke-cover) by Jokestega retains the same profile of its other peer jokes that contains no hidden message. Basically, most alterations introduced by Jokestega are non-linguistic and do not produce any flaws (noise), as demonstrated in the implementation section, deeming statistical attacks on joke-cover very ineffective. For more details about text steganography statistical experimental results refer to Desoky (2010a), Desoky (2011c), Desoky (2009a), Desoky (2012) and Friedland and Allan (2008).

6 Conclusion

A novel steganography methodology demonstrated in this paper that exploits textual jokes to conceal data. Namely, Automatic Joke Generation-Based Steganography Methodology (Jokestega). Jokestega methodology takes advantage of recent advances in the field of AJG techniques to automate the generation of textual steganographic cover. Linguistically, in a corpus of jokes one may judge a number of documents to be the same joke although letters, locations and other details are different. In addition, joke and puns commonly can be retold with totally different vocabulary, while still retaining their core identities. Therefore, Jokestega pursues the common variations among jokes to conceal data. Examples of ordinary variations among textual jokes may include: different text of same joke, altering some words and letters without changing the core of a joke, taking advantage of using non-classic linguistic rules of writing, inserting non-linguistic elements symbols and any other factors and elements. Obviously, joking behaviour intrinsically retain in all human regardless of race, nationality, religion etc., which legitimises the use of steganography. Presumably, when someone is joking, anything may be said which legitimises and convey the use of joke-based steganography. This makes employing textual jokes very attractive as steganographic carrier for camouflaging data, which it is definitely also allows communicating parties to establish a covert channel without raising suspicion to deliver a steganographic joke-cover. It is worth noting that Jokestega follows Nostega paradigm, which implies that joke-cover is noiseless. Improving the bit-rate is worth investigating it in the future.

References

Binsted, K. (1996) *Machine Humour: An Implemented Model of Puns*, PhD Thesis, University Of Edinburgh, Edinburgh, Scotland.

Binsted, K., Pain, H. and Ritchie, G. (1997) 'Children's evaluation of computer-generated punning riddles', *Pragmatics and Cognition*, Vol. 5, No. 2, pp.305–354.

Black, R., Waller, A., Ritchie, G., Pain, H. and Manurung, R. (2007) 'Evaluation of joke-creation software with children with complex communication needs', *Communication Matters*, Vol. 21, No. 1, pp.23–28.

Chapman, M. and Davida, G. (1997) 'Hiding the hidden: a software system for concealing ciphertext as innocuous text', *Proceedings of the International Conference on Information and Communications Security*, 11–14 November, Beijing, China, pp.335–345.

Chapman M. and Davida G.I. (2002) 'Plausible deniability using automated linguistic steganography', *Proceedings of* the *International Conference on Infrastructure Security (InfraSec'02)*, 1–3 October, Bristol, UK, pp.276–287.

Chapman, M., Davida, G.I. and Rennhard, M. (2001) 'A practical and effective approach to large-scale automated linguistic steganography', *Proceedings of the Information Security Conference (ISC'01)*, 1–3 October, Malaga, Spain, pp.156–165.

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

Desoky, A. (2009a) *Nostega: A Novel Noiseless Steganography Paradigm*, PhD Dissertation, University of Maryland, Baltimore County.

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

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

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

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

Desoky, A. (2010c) 'Headstega: email-headers-based steganography methodology', *International Journal of Electronic Security and Digital Forensics*, Vol. 3, No. 4, pp.289–310.

Desoky, A. (2011a) 'Sumstega: summarization-based steganography methodology', *International Journal of Information and Computer Security*, Vol. 4, No. 3, pp.234–263.

Desoky, A. (2011b) 'Matlist: mature linguistic steganography methodology', *Journal of Security and Communication Networks*, Vol. 4, No. 6, pp.697–718.

Desoky, A. (2011c) 'Edustega: an education-centric steganography methodology', *International Journal of Security and Networks*, Vol. 6, Nos. 2/3, pp.153–173.

Desoky, A. (2012) *Noiseless Steganography: The Key to Covert Communications*, Taylor & Francis.

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

Desoky, A. and Younis, M. (2009) 'Chestega: Chess Steganography Methodology', *Journal of Security and Communication Networks*, Vol. 2, No. 6, pp.555–566.

Desoky, A., Younis, M. and El-Sayed, H. (2008) 'Auto-summarization-based steganography', *Proceedings of the 5th IEEE International Conference on Innovations in Information Technology*, 16–18 December, Al-Ain, UAE, pp.608–612.

Friedland, L. and Allan, J. (2008) 'Joke retrieval: recognizing the same joke told differently', *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, 26–30 October, Napa Valley, California, USA, pp.883–892.

Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R. and Atallah, M.J. (2005a) *Translation-Based Steganography*, Technical Report CSD TR# 05-009, Purdue University.

Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R. and Atallah, M.J. (2005b) 'Translation-based steganography', *Proceedings of Information Hiding Workshop (IH'2005)*, 6–8 June, Barcelona, Spain, pp.213–233.

Joke Generator Project STANDUP (2003) *Joke Generator Project STANDUP*. Available online at: http://www.csd.abdn.ac.uk/research/standup (accessed on 1 August 2009).

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

Manurung, R., Low, A., Trujillo-Dennis, L., O'Mara, D., Pain, H., Ritchie, G. and Waller, A. (2004) 'Interactive computer generation of jokes for language skill development', *Conference of International Society for Humor Studies*, Dijon, France.

Manurung, R., O'Mara, D., Pain, H. Ritchie, G. and Waller, A. (2006a) 'Building a lexical database for an interactive joke-generator', *Proceedings of LREC, 5th International Conference on Language Resources and Evaluation (CD)*, 24–26 May, Genoa, Italy.

Manurung, R., Ritchie, G., O'Mara, D., Waller, A. and Pain, H. (2006b) 'Combining lexical resources for an interactive language tool', *Proceedings of ISAAC 2006, 12th Biennial International Conference of the International Society for Augmentative and Alternative Communication (CD)*, 29 July–5 August, Düsseldorf, Germany.

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D. and Black, R. (2008a) 'The construction of a pun generator for language skills development', *Applied Artificial Intelligence*, Vol. 22, No. 9, pp.841–869.

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D. and Black, R. (2008b) 'Adding phonetic similarity data to a lexical database', *Language Resources and Evaluation*, Vol. 42, No. 3, pp.319–324.

Meng, P., Shi, Y-Q., Huang, L., Chen, Z., Yang, W. and Desoky, A. (2011) 'LinL: Lost in n-best list', Proceedings of 13th Information Hiding Conference, 18–20 May, Prague, Czech Republic, pp.329–341.

O'Mara, D. and Waller, A. (2003) 'What do you get when you cross a communication aid with a riddle?', *The Psychologist*, Vol. 16, No. 2, pp.78–80.

O'Mara, D., Waller, A., Manurung, R., Ritchie, G. and Pain, H. (2004a) 'I say, I say, I say', *Australian Group on Severe Communication Impairment News*, Vol. 23, No. 2, pp.1443–9107.

O'Mara, D., Waller, A., Ritchie, G., Pain, H. and Manurung, R. (2004b) 'The role of assisted communicators as domain experts in early software design', *Proceedings of ISAAC, 11th Biennial International Conference of the International Society for Augmentative and Alternative Communication (CD)*, 6–10 October, Natal, Brazil.

O'Mara, D., Waller, A. and Todman, J. (2004c) 'The recognition and use of verbal humour by children with language impairment', *Presented as Emerging Scholar, Conference of International Society for Humor Studies*, Dijon, France.

O'Mara, D., Waller, A., Manurung, R., Ritchie, G., Pain, H. and Black, R. (2006) 'Designing and evaluating joke-building software for AAC users', *Proceedings of ISAAC 2006, 12th Biennial International Conference of the International Society for Augmentative and Alternative Communication (CD)*, 29 July–5 August, Düsseldorf, Germany.

Reiter, E. and Dale, R. (2000) *Building Natural Language Generation Systems*, Cambridge University Press, Cambridge, UK.

Ritchie, G., Manurung, R., Pain, H., Waller, A. and O'Mara, D. (2006) 'The STANDUP interactive riddle builder', *IEEE Intelligent Systems*, Vol. 21, No. 2, pp.67–69.

Ritchie, G., Manurung, R., Pain, H., Waller, A., Black, R. and O'Mara, D. (2007) 'A practical application of computational humour', in Cardoso, A. and Wiggins, G.A. (Eds): *Proceedings of the 4th International Joint Conference on Computational Creativity*, 17–19 June, London, UK, pp.91–98.

Shirali-Shahreza, M. and Shirali-Shahreza, M.H. (2007) 'Text steganography in SMS', *Proceedings of the International Conference on Convergence Information Technology*, 21–23 November, Gyeongjui, Korea, pp.2260–2265.

Stutsman, R., Grothoff C., Atallah, M.J. and Grothoff, K. (2006) 'Lost in just the translation', *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06)*, 23–37 April, Dijon, France, pp.338–345.

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*, 29 January–1 February, San Jose, CA, USA.

Waller, A., Black, R., O'Mara, D.A., Pain, H., Ritchie, G. and Manurung, R. (2009) 'Evaluating the STANDUP pun generating software with children with cerebral palsy', *ACM Transactions on Accessible Computing (TACCESS)*, Vol. 1, No. 3, pp.1–27.

Waller, A., O'Mara, D., Manurung, R., Pain, H. and Ritchie, G. (2005) 'Facilitating user feedback in the design of a novel joke generation system for people with severe communication impairment', in Salvendy, G. (Ed.): *Proceedings of HCII 2005 (CD)*, Lawrence Erlbaum, NJ, USA, Vol. 5.

Wayner, P. (1992) 'Mimic functions', *Cryptologia*, Vol. 16, No. 3, pp.193–214.

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

Notes

1 MIT Project, Online Joke Generator System. Available online at: http://scratch.mit.edu/projects/Ronan1888/4365 (accessed on 16 May 2011).

2 Online Joke Generator System. Available online at: http://www.jokes2000.com (accessed on 16 May 2011).

3 Online Joke Generator System. Available online at: http://www.thejokegenerator.com (accessed on 16 May 2011).

4 Dictionary and Thesaurus – Merriam-Webster Online. Available online at: www.merriam-webster.com (accessed on 31 July 2008).

5 Online Dictionary Net. Available online at: www.online-dictionary.net (accessed on 31 July 2008).

6 Cambridge Dictionaries Online, Cambridge University Press Available online at: www.dictionary.cambridge.org (accessed on 31 July 2008).

7 Microsoft Word 97 Available online at: http://www.microsoft.com/en/us/default.aspx (accessed on 31 July 2008).