The Journal of Systems and Software 85 (2012) 2385–2394

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A compression-based text steganography method

Esra Satir[a,*], Hakan Isik[b]

[a] Selcuk University, Technical Education Faculty, Computer and Electronic Education, Konya, Turkey
[b] Selcuk University, Technology Faculty, Department of Electronic Engineering, Konya, Turkey

ARTICLE INFO

Article history:
Received 29 August 2011
Received in revised form 30 April 2012
Accepted 7 May 2012
Available online 23 May 2012

Keywords:
Steganography
Text steganography
Data compression
LZW algorithm

ABSTRACT

In this study, capacity and security issues of text steganography have been considered to improve by proposing a novel approach. For this purpose, a text steganography method that employs data compression has been proposed. Because of using textual data in steganography, the employed data compression algorithm has to be lossless. Accordingly, LZW data compression algorithm has been chosen due to its frequent use in the literature and significant compression ratio. The proposed method constructs – uses stego keys and employs Combinatorics-based coding in order to increase security. Secret information has been hidden in the chosen text from the previously constructed text base that consists of naturally generated texts. Email has been chosen as communication channel between the two parties, so the stego cover has been arranged as a forward mail platform. By means of the proposed scheme, capacity has been reached to 7.042% for the secret message containing 300 characters (or 300·8 bits). Finally, comparison of the proposed scheme with the other contemporary methods in the literature has been carried out. Experimental results show that the proposed scheme provided a significant increment in terms of capacity.

1. Introduction

The proliferation of network technologies and digital devices makes the delivery of digital multimedia fast and easy. However, distributing digital data over public networks such as Internet is not reliable because of copyright violation, counterfeiting, forgery, and fraud. Therefore, methods for protecting digital data, especially sensitive data, are extremely essential (Chang and Kieu, 2010). Although the use of electronic documents is widespread, very few people can recognize that these documents contain "hidden data". The reason for using the word "hidden" is that these data are normally located within a file, but cannot be identified using common methods. Hidden data can be classified into two kinds. The first is automatically created by the application, and the second is created and concealed by an individual for specific purposes (Park and Lee, 2009). Secret data can be protected by cryptographic methods, conventionally. However, transmitting the encrypted secret data by cryptosystems is prohibited by some dictatorial governments, or the meaningless form of the encrypted data may attract the attention of interceptors (e.g., wardens or sensors) that are designed to stop any secret communications (Chang and Kieu, 2010). Alternatively, confidential data can be protected by employing information hiding techniques. Generally, information hiding includes digital watermarking and steganography (Chang and Kieu, 2010). Watermarking is different from steganography in its main goal. Watermarking is used for copyright protection, broadcast monitoring, transaction tracking, and similar activities. A watermarking scheme alters a cover object, either imperceptibly or perceptibly, to embed a message about the cover object (e.g., the owner's identifier). It can be observed as steganography that is concentrating on high robustness and very low or almost no security (Gutub and Fattani, 2007). In contrast, steganography is used primarily for secret communications (Chang and Kieu, 2010). Steganography is the art of writing secret data in such a way that no one except the intended receiver knows about the existence of secret data. Successful steganography depends upon the carrier medium not to raise attention (Sajedi and Jamzad, 2010).

There are three main issues to be considered when studying steganographic systems: capacity (or bitrate), security and robustness (Al-Haidari et al., 2009). Capacity refers to the amount of data bits that can be hidden in the cover medium. Security relates to the ability of an eavesdropper to figure the hidden information easily. Robustness is concerned about the resist possibility of modifying or destroying the unseen data (Gutub and Fattani, 2007). In steganography for digital systems, the cover media used to hide the message can be text, image, video or audio files (Aabed et al., 2007).

We propose a compression based text steganography method in order to improve capacity and security. Namely, the problem is to obtain a significant incerement in the amount of secret data that is aimed to be hidden in cover medium while we desire to complicate

* Corresponding author.
E-mail address: esatir@selcuk.edu.tr (E. Satir).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.05.027

the extraction procedure of the secret data. In the propsed method, secret data has been embedded in the chosen text from the previously constructed text base. The text base contains naturally generated texts like notification texts, abstracts of articles, etc. which can be used for a group speech. While embedding, originality of the chosen text has been protected by only camouflaging the secret information. Email has been chosen as communication channel between the two parties, so the stego cover has been arranged as a forward mail platform. While arranging the stego cover as a forward mail platform, we use the previously arranged email address list for choosing the email addresses. Meanwhlie, this email address list has been used as a global stego key that is shared both by the sender and the recipient berofehand.

For the first purpose, capacity increment, we prefer to use data compression techniques. In a data compression process the aim is to decrease the redundancy of a given data description (Galambos and Bekesi, 2002). Generally, data compression algorithms are classified as lossless or lossy. Lossless data compression involves a transformation of representation of the original data set such that it is possible to reproduce exactly the original data set by performing a decompression transformation and it is used when the original and the decompressed files must be identical (in compressing text files, executable codes, word processing files, etc.). Lossy data compression involves a transformation of representation of the original data set such that it is impossible to reproduce exactly the original data set, but an approximate representation is reproduced by performing a decompression transformation. This type is used on the Internet and especially in streaming media and telephony applications (Al-Bahadili, 2008). In case of textual information, while performing a compression/decompression process we must recover exactly the original data. In case of pictures or voices – without getting into deep trouble – it is allowed to get an approximation of the original information (Galambos and Bekesi, 2002). In our problem, we have to protect the originality because of dealing with textual data. So we have to use a lossless data compression technique. Accordingly, we propose to employ LZW data compression algorithm because of its good compression ratio and frequent usage in the literature. The LZW algorithm first reads the data and tries to match a sequence of data bytes as large as possible with an encoded string from the dictionary. The matched data sequence and its succeeding character are grouped together and then added to the dictionary for encoding later data sequences (Liang et al., 2008). For the second purpose, security improvement, we propose to employ stego-keys. We can classify the employed stego keys into two classes according to their missions. One of them is the constructed stego keys during embedding phase of the proposed scheme and the other is the previously constructed global stego key which is shared both by the sender and the receiver beforehand. Meanwhile, employing Combinatorics-based coding in order to support the desired randomness (see Jun et al., 2011 for additional information) provides a positive contribution to the security. Combinatorics-based coding is predictable to the receiver but quite random to an observer who tries to analyze the steganographic cover, rendering the steganographic cover more resilient (Desoky, 2009). For this purpose, Latin Square has been employed (see Easton and Gary Parker, 2001 and Colbourn, 1984 for additional information). By basing on Bailey and Curran (2006), we can say that LZW coding also provides contribution to security. Evaluation procedure has been performed via capacity measurements. Capacity has been measured in terms of percent, by calculating the rate of secret data that is embedded in the stego cover. Besides, a general evaluation has been performed in terms of capacity by comparing the proposed scheme with the other comtemporary methods in the literature.

The rest of this paper has been organized as follows: Section 2 provides a brief overview of the related text steganography methods in the literature. The proposed method has been explained in Section 3, in depth by mentioning embedding phase, construction and usage of stego keys and extracting phase. In Section 4, we presented the analysis of capacity calculation. In Section 5, we provided the performed experiments and the obtained experimental results for the proposed method. Finally, we summarized the most relevant conclusions of this work in Section 6.

2. Related work

Several attempts have been performed to design text steganography methods for different languages like English, Chinese, Persian and Arabic, etc. In this section, some of the previous studies are explained.

Wayner (1992, 2002) introduced the mimic functions approach. In this method, the inverse of Huffman Code is employed by inputting a data stream of randomly distributed bits. The aim of this operation is to produce the text that obeys the statistical profile of a particular normal text. Thus, the generated text by mimic functions is resilient against statistical attacks. The output of a regular mimic functions is gibberish. Accordingly, this makes the text extremely suspicious (Desoky, 2009). Maher (1995) proposed a text data hiding program that is called TEXTO. TEXTO was designed to transform uuencoded or PGP ASCII-armored ASCII data into English sentences. It is convenient for exchanging binary data, especially encrypted data. Here, the secret data is replaced by English words. Namely, TEXTO works like a simple substitution cipher (Wang et al., 2009a).

Chapman and Davida (1997, 2001, 2002) introduced a steganographic scheme that consists of two functions called NICETEXT and SCRAMBLE. NICETEXT transforms cipher text to the text that looks like natural language. There are synonyms-based approach which attracted the attention of many researchers in the last decade like Winstein (1999, 2008), Nakagawa (Nakagawa et al., 2001) and Murphy et al. (Murphy and Vogel, 2007). In synonym-based approach, the cover text may look legitimate from a linguistics point of view given the adequate accuracy of the chosen synonyms. But reusing the same piece of text to hide a message can raise suspicion (Desoky, 2009).

Sun et al. (2004) proposed a scheme that uses the left and right components of Chinese characters. The proposed scheme is called L-R scheme. In L-R scheme, the mathematical expression of all Chinese characters is introduced into the text data hiding strategy. It chooses those characters with left and right components as candidates to hide the secret information. During the embedding phase, if the secret information is "0", the L-R scheme keeps the candidate character's original appearance; otherwise, the character's appearance must be modified by adjusting the space between the left and right components of the current candidate character (Wang et al., 2009a). In order to increase the hiding capacity of Sun et al.'s L-R scheme, Wang et al. (2009a) revised their scheme by adding the up and down structure of Chinese characters as an extra candidate set. Besides, a reversible function to Sun et al.'s L-R scheme has been added to make it possible for receivers to obtain the original cover text and use it repeatedly for later transmission of secrets after the initial hidden secrets have been extracted (Wang et al., 2009a).

Since communications via chat room become more popular in people's lives; Wang and Chang proposed another new text steganography method. The proposed method embeds secret information into emotional icons (also called emoticons) in chat rooms over the Internet. In this method, firstly the sender's emoticon table should be unanimous with the receiver's emoticon table. Next, the sender and the receiver classify those emoticons in the emoticon table into several sets according to their meaning (like cry, smile laugh) and every emoticon belongs to one set. The order number of

an emoticon, counting from 0, in its set is the secret bits that will be embedded. Thus, the proposed steganographic scheme uses a secret key to control the order of emoticons in each constructed set. Only the sender and the receiver keep this key. The embedding capacity has also been improved due to the tremendous numbers of emoticons used in many kinds of chat rooms (Wang et al., 2009b).

Grothoff et al. introduced a new approach that is called translation-based steganographic scheme. This scheme hides a message in the errors (noise), which are naturally encountered in a machine translation (MT). The secret message is embedded by performing a substitution procedure on the translated text using translation variations of multiple MT systems (Desoky, 2009). Another noise-based approach was proposed by Topkara et al. in 2007. Here, typos and ungrammatical abbreviations in a text, e.g., emails, blogs, forums, etc., are employed for hiding data. However, this approach is sensitive to the amount of noise (errors) that occurs in a human writing (Desoky, 2009).

In 2009, Desoky proposed a method called Listega. Listega takes advantage of using textual list to camouflage data by exploiting itemized data to conceal messages. Simply, it encodes a message then assigns it to legitimate items in order to generate a cover text in a form of list. Listega establishes a covert channel among communicating parties by employing justifiably reasons based on the common practice of using textual list of items in order to achieve unsuspicious transmission of generated covers (Desoky, 2009).

Por et al. (2012) proposed a data hiding method based on space character manipulation called UniSpaCh. UniSpaCh is proposed to embed information in Microsoft Word document using Unicode space characters. In addition, white spaces are considered to encode payload because they appear throughout the document (i.e., available in large number), and the manipulation of white spaces has insignificant effect to the visual appearance of document. UniSpaCh embeds payload into inter-sentence, inter-word, end-of-line and inter-paragraph spacings by introducing Unicode space characters (Por et al., 2012).

3. The proposed LZW based text steganography method

In this section, embedding phase, construction and usage of stego keys and extracting phase of the proposed method have been explained.

3.1. Embedding phase

Before explaining embedding procedure, let's mention the following variables:

S: Secret message
T: Text base
Text: A text in the text base
$\overrightarrow{\Delta D}$: Relative distances
A: Set of email address extensions
D: Matrix of relative distances
E: Matrix of exceedings
R: Matrix of reconstructed relative distances
K_1: Global stego key
K_2: Set of chosen and modified email addresses

$$S = \{a_1, a_2, \cdots, a_m\}$$

$$D = D_{48,m} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,m} \\ \vdots & & \vdots \\ d_{48,1} & \cdots & d_{48,m} \end{bmatrix}$$

$$T = T_{4,83,000} = \begin{bmatrix} t_{1,1} & \cdots & t_{13,000} \\ \vdots & & \vdots \\ t_{48,1} & \cdots & t_{4,83,000} \end{bmatrix}$$

$$E = E_{48,m} = \begin{bmatrix} e_{1,1} & \cdots & e_{1,m} \\ \vdots & & \vdots \\ e_{48,1} & \cdots & e_{48,m} \end{bmatrix}$$

$$Text = \{b_1, b_2, \cdots, b_n\}$$

$$R = R_{48,m} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,m} \\ \vdots & & \vdots \\ r_{48,1} & \cdots & r_{48,m} \end{bmatrix}$$

$$\overrightarrow{\Delta D} = (c_1, c_2, \ldots, c_m)$$

$$K_1 = \{j_1, j_2, \cdots, j_{676}\}$$

Since K_1 consists of the combinations of each pair of letters, the maximum index has been computed as $26 \times 26 = 676$. We can represent K_1 as follows:

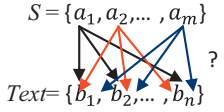
K_1 = {aa...@hotmail.com, ab...@hotmail.com, ac...@hotmail.com, ..., zv...@hotmail.com, zy...@hotmail.com, zz...@hotmail.com}

$$A = \underset{(Binary\ index=}{} \underset{000,}{\{hotmail.com}, \underset{001,}{gmail.com}, \underset{010,}{yahoo.com}, \underset{011,}{msn.com},$$

$$\underset{100,}{windowslive.com}, \underset{101,}{mail.com}, \underset{110,}{myspace.com}, \underset{111)}{mynet.com}\}$$

Initially, let's mention that S is a set that contains characters of secret message: a. *Text* represents a text in text base and it contains characters of text: b. T is a matrix that contains all *Texts* in text base. Here 48 is the number of *Texts* in text base. 3000 is the maximum character number of a *Text* in text base. If a *Text* has fewer characters than 3000; the corresponding elements of T are assigned as 0.

Step 1. S contains characters of secret message; a and *Text* contains characters of text; b. We look for a situation, where $a = b$. Accordingly, $\overrightarrow{\Delta D}$ is a vector of which elements (c) are differences between the indexes of b elements where the character mapping forms. We can express this operation as follows:



Since characters are ASCII codes;

$$a_1 = b_1 \rightarrow a_1 - b_1 = 0$$
$$a_1 = b_2 \rightarrow a_1 - b_2 = 0$$
$$\vdots$$
$$a_1 = b_n \rightarrow a_1 - b_n = 0$$

Let's assume that $a_1 = b_2$. In this case, the value we need for $\overrightarrow{\Delta D}$ is the index of b, namely *2*. In the second step we consider the following elements of S and *Text*:

$$a_2 = b_3 \rightarrow a_2 - b_3 = 0$$
$$a_2 = b_4 \rightarrow a_2 - b_4 = 0$$
$$\vdots$$
$$a_2 = b_n \rightarrow a_2 - b_n = 0$$

Let's assume that $a_2 = b_4$. In this case, the value we need for $\overrightarrow{\Delta D}$ is the difference of the current index of b (*4*) and the previous index of b (*2*), namely $4 - 2 = 2$. This operation forms iteratively through *Text* till the end of secret message, in order to construct $\overrightarrow{\Delta D}$.

Step 2. We construct $\overrightarrow{\Delta D}$ for every *Text* in T. Then we hold every $\overrightarrow{\Delta D}$ in order to form D. Accordingly, we obtain:

$$D = D_{48,m} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,m} \\ \vdots & & \vdots \\ d_{48,1} & \cdots & d_{48,m} \end{bmatrix}$$

D is a matrix of $48 \times m$. We construct D in order to choose the most appropriate *Text* in T for LZW coding.

Step 3. In this step let's examine whether elements of D exceed 26. If yes, we obtain E and R as follows;

$$E = E_{48,m} = D \backslash 26 \quad (1)$$

$$R = R_{48,m} = D \bmod 26 \quad (2)$$

Here, the aim is to benefit Latin Square (please refer to Annex 1) without exceeding its boundaries. If d does not exceed 26, notice that the corresponding e will be 0 and the corresponding r will be equal to d.

Step 4. We estimate the number of dual pattern repetition for every line of the constructed R in the previous step and we obtain:

$$P = \begin{bmatrix} p_1 \\ \vdots \\ p_{48} \end{bmatrix}$$

We chose the maximum p value in P. Accordingly; we choose the lines of E and R which correspond to the line index of this calculated maximum p in P. Let's denote these lines as $\vec{E}$ (exceeding vector for the reconstructed $\overrightarrow{\Delta D}$) and $\vec{R}$ (reconstructed $\overrightarrow{\Delta D}$). Meanwhile, in T, we choose *Text* as cover text (T^*) which corresponds to the line index of maximum p value. Here, the aim is to increase the performance of LZW coding.

Step 5. In this step, $\vec{R}$ is compressed by employing LZW coding.

5.1 The integers between *1* and *26* have been used to construct the initial LZW dictionary (These codes will be employed in case of meeting no repetition.).
5.2 LZW dictionary is updated for every met symbol or symbol string by considering the repetition cases. The concerning symbol or symbol string is encoded by considering the corresponding index in the dictionary.

As the result of LZW coding, we have $\overrightarrow{R'}$ and $\left\|\overrightarrow{R'}\right\| < \left\|\overrightarrow{R}\right\|$. Then we represent each element of $\overrightarrow{R'}$ in base 2, namely we perform the operation: $\left(\overrightarrow{R'}\right)_2$ Then we concatenate every element in order have a bit stream.

Step 6. We divide the obtained bit stream into the groups each of which contains 12 bits. In every group, the first 9 bits will be handled for constructing the userside of email address. The remaining 3 bits will be handled for modifying the email address extension (for e.g. hotmail.com). Let G_1 be the first 9 bits. By performing the following operations, we obtain two integers:

$$x = (G_1)_{10} \backslash 26 \tag{3}$$

$$x = (G_1)_{10} \bmod 26 \tag{4}$$

These integers will then be used in order to choose email addresses from K_1. K_1; global stego key; is a set that consists of the previously generated email address list. x and y are converted to letters by employing *Latin Square* (refer to Annex 1). Then these two letters are mapped to one email address by employing K_1.

Let G_2 be the last 3 bits of each group.

$$z = (G_2)_{10} \tag{5}$$

As mentioned before, z will be used to modify email address extension by employing A. Email address extension is determined by using binary indexes of elements in A (this modification is handled as a stego key which is a part of K_2).

Step 7. Finally, we modify these chosen email addresses in order to complete construction of K_2 set by using $\vec{E}$ (estimated in step 4). This modification is performed by adding exceeding number to the chosen email address before "@" character and also handled as a stego key. Since there are not any rules or constraints while forming an email address, exceeding numbers seem as a natural part of the email address (If there is no exceeding we do not modify the corresponding email address's userside.). Thus, K_2 is a set that consists of the chosen and modified email addresses. Namely, many

```
Get S
Get T
For each Text in T
        Calculate ΔD⃗
        Generate a line of D by getting ΔD⃗
End for
For each ΔD⃗ in D
        For each c in ΔD⃗
                If c > 26 then
                        e = int (c/26)
                        r = c mod 26
                else
                        e = 0
                        r = 0
                Endi f
                Generate E⃗ by getting e
                Generate R⃗ by getting r
        End for
        Generate a line of E by getting E⃗
        Generate a line of R by getting R⃗
End for
For each R⃗ in R
        Calculate p
        Generate P⃗ by getting p
End for
Find maximum p and its index in P⃗
Get the line of R as R⃗ which corresponds to the index of maximum p
Get the line of E as E⃗ which corresponds to the index of maximum p
Get the line of T as T* which corresponds to the index of maximum p
Generate R⃗' via LZW coding
Bit Stream = (R⃗')2
Generate 12 bit groups by seperating Bit Stream
For each 12 bit groups in Bit Stream
        Get G1
        x = (G1)10\26
        y = (G1)10mod 26
        Generate letter equivalents for x and y via Latin Square
        Get the corresponding email address from K1
        Get G2
        z = (G2)10
        Generate a stego key by getting email address extension from A according to the value of z
End for
i = 0
For each 3 elements of E⃗
        Generate a stego key by adding 3 elements to K2[i] before "@"
        i= i+1
End for
Generate Stego Cover by getting T* and K2
```

Fig. 1. Pseudo codes for the embedding phase.

elements of the set K_2 are arranged as stego keys. Besides, let's mention that $s(K_2) < s(K_1)$.

Step 8. We construct stego cover by using both T^* as cover text and K_2 set. Thus, the medium in order to conceal the secret message is shown as a forward mail platform. Here, notice that definite email addresses are employed as stego keys according to the exceeding numbers (estimated in step 4) and with the last 3 bits (z) that determines email address extension (estimated in Step 6). However, it is not possible to know which of them are stego keys without having K_1. K_1 is shared only by the sender and the recipient beforehand. Pseudo codes of the embedding phase have been provided in Fig. 1.

3.2. Construction and usage of stego keys

The proposed method constructs and uses stego keys in order to increase security. We can classify the used stego keys into two classes according to their missions. One of them is the set of chosen and modified email addresses; K_2. This has been performed by embedding overflow information before "@" character and choosing the email address extension according to z via A. Construction of K_2 has been carried out in the embedding phase (in Steps 6 and 7). K_2 is used to embed the information, which shows the correct position of the hidden character. Thus, these constructed stego keys seem as a natural part of the stego cover which has been arranged as a forward mail platform. This email platform is only a simulation. Namely, the mail will not be sent to the chosen and arranged email addresses. It will be sent to the main recipient, only.

The other one is a global stego key; K_1, like the employed ones according to the studies of Lou et al. titled as "A novel adaptive steganography based on local complexity and human vision

sensitivity" (Lou et al., 2010) and Wang et al. titled as; "Emoticon-based Text Steganography in Chat" (Wang et al., 2009b). It is a set consisting of the previously constructed email address list that is shared by both sender and recipient beforehand. The purpose of using a global stego key is to detect the correct position information, which is embedded in the modified email addresses. Namely, K_2 needs a global key in order to solve and to detect correct position information of the secret message.

3.3. Extracting phase

Step 1. Let's get the stego cover. Then compare each element of K_2 to each element of K_1. The purpose of this operation is to find out whether there are differences between the compared email addresses. We consider the userside of each email address before "@" character. In case of any difference, we extract the different information. Notice that this extracted numerical information are elements of $\vec{E}$. If these compared email addresses are not different from each other, there is no overflow and the concerning element of $\vec{E}$ will be 0.

Step 2. We now investigate the elements of K_2 more clearly; the first two characters of each email address. We convert them to numbers by employing *Latin Square*. Thus we obtain x and y. Meanwhile, we have to investigate the email address extension to obtain z. For this purpose, we estimate z via A by employing the binary index number of each element in A. Then, we can calculate G_1 and G_2 for each group of 12 bits by using the following equations:

$$G_1 = (x \cdot 26)_2 \tag{6}$$

$$G_2 = (z)_2 \tag{7}$$

By concatenating these obtained G_1 and G_2 values, we have $(\vec{R'})_2$, the compressed bit stream via LZW.

Step 3. We have to decompress $\vec{R'}$ via LZW coding in order to obtain $\vec{R}$.

3.1 The integers between *1* and *26* have been used to construct the initial LZW dictionary.
3.2 LZW dictionary is updated for every symbol or symbol string that is met. The concerning symbol or symbol string is decoded by considering the corresponding index in the dictionary.

At the end of this decompression, we obtain $\vec{R}$.

Step 4. We have to estimate the initial $\vec{\Delta D}$ by employing $\vec{R}$ and $\vec{E}$. If we denote the elements of $\vec{R}$ and $\vec{E}$ as r and e, respectively (In embedding phase, we denoted the elements of $\vec{\Delta D}$ as c.):

$$c = r + (26 \cdot e) \tag{8}$$

Step 5. By using elements of $\vec{\Delta D}$ we can extract the elements of S through T^*, in the stego cover. By advancing c at a time through elements of T^*, we detect the index number where character mapping forms (the place where $a = b$). Thus we can extract the concerning element of S. This operation repeats consecutively for every element of $\vec{\Delta D}$. Finally, we concatenate the extracted elements and we obtain S. Pseudo codes of the extracting phase have been provided in Fig. 2.

```
Get Stego Cover
For each k in K2
        If k= j (element of K1 ) then
                e= 0
                Generate E⃗ by getting e
        Else
                e = k − j
                e′ = e\100
                e″ = [e − (e′ · 100)]\10
                e‴ = e mod 10
                Generate 3 elements of E⃗ by getting e′, e″, and e‴
        End if
End for
For each k in K2
        Find x and y via Latin Square
        Find z via A
        G1 = (x · 26)2
        G2 = (z)2
        Generate (R⃗′)2 by getting G1 and G2
End for
Generate R⃗ via LZW coding
i =0
For each r in R⃗
        c = r + (26 · E⃗[i])
        i =i+1
        Generate ΔD⃗ by getting c
End for
For each c in ΔD⃗
        a = T*[c]
        Generate S by getting a
End for
```

Fig. 2. Pseudo codes for the extracting phase.

4. Analysis of capacity estimation for the proposed method

Bitrate or capacity is defined as the size of the hidden message relative to the size of the cover (Desoky, 2009). In this case, we can formulate bitrate as follows:

$$C = \frac{bits\ of\ secret\ message}{bits\ of\ stego\ cover} \tag{9}$$

In Table 1, some information regarding with secret messages (S), character numbers of secret messages (n), numbers of dual pattern repetition (p) and bitrates (or capacity – C) has been provided. p values have been calculated by counting the repetition number of each pair in $\vec{R}$ and then getting sum of them. The reason of considering dual pattern repetition is the possibility of meeting patterns with triad, quad, etc. repetitions. As the length (n) of S increases, it can be seen that dual pattern repetition number (p) increases, too. This has a positive contribution to LZW coding which performs compression process by basing on symbol repetition.

Table 1
n, p and C informations of 12 sample secret messages.

	Secret message (S)	n	p	C (%)
S_1	the import	10	11	0.679
S_2	the importance and s	20	22	1.153
S_3	the importance and size of tex	30	17	2.34
S_4	the importance and size of text data hav	40	25	3.642
S_5	the importance and size of text data have increase	50	34	4.071
S_6	the importance and size of text data have increased at an ac	60	36	3.926
S_7	the importance and size of text data have increased at an accelerating	70	41	4.176
S_8	the importance and size of text data have increased at an accelerating pace beca	80	44	4.507
S_9	the importance and size of text data have increased at an accelerating pace because the re	90	62	4.777
S_{10}	the importance and size of text data have increased at an accelerating pace because the reliance on	100	69	5.481
S_{11}	the importance and size of text data have increased at an accelerating pace because the reliance on text based	110	84	5.268
S_{12}	the importance and size of text data have increased at an accelerating pace because the reliance on text based web infor	120	96	5.527

Table 2
$\overrightarrow{R}$ of 12 sample secret messages.

	$\overrightarrow{R}$
$\overrightarrow{R_1}$	(19, 1, 1, 1, 6, 1, 1, 1, 1, 1)
$\overrightarrow{R_2}$	(19, 1, 1, 1, 6, 1, 1, 1, 1, 1, 1, 1, 8, 5, 2, 5, 24, 18, 1, 3)
$\overrightarrow{R_3}$	(8, 1, 1, 1, 5, 13, 7, 2, 14, 1, 6, 2, 3, 19, 7, 16, 23, 1, 1, 10, 16, 12, 1, 1, 1, 1, 9, 2, 1)
$\overrightarrow{R_4}$	(8, 1, 1, 1, 5, 13, 7, 2, 14, 1, 6, 2, 3, 19, 7, 16, 23, 1, 1, 10, 16, 12, 1, 1, 1, 1, 9, 2, 1, 1, 5, 15, 6, 5, 6, 3, 9, 24, 22)
$\overrightarrow{R_5}$	(8, 1, 1, 1, 5, 13, 7, 2, 14, 1, 6, 2, 3, 19, 7, 16, 23, 1, 1, 10, 16, 12, 1, 1, 1, 1, 9, 2, 1, 1, 5, 15, 6, 5, 6, 3, 9, 24, 22, 1, 2, 4, 11, 9, 1, 1, 7, 2, 18)
$\overrightarrow{R_6}$	(8, 1, 1, 1, 5, 13, 7, 2, 14, 1, 6, 2, 3, 19, 7, 16, 23, 1, 1, 10, 16, 12, 1, 1, 1, 1, 9, 2, 1, 1, 5, 15, 6, 5, 6, 3, 9, 24, 22, 1, 2, 4, 11, 9, 1, 1, 7, 2, 18, 17, 1, 2, 15, 8, 15, 6, 1, 23, 8)
$\overrightarrow{R_7}$	(1, 16, 17, 2, 19, 5, 11, 17, 13, 2, 6, 17, 6, 7, 8, 14, 21, 2, 4, 4, 1, 20, 1, 1, 1, 10, 1, 1, 1, 1, 1, 1, 4, 3, 2, 7, 3, 9, 7, 14, 13, 5, 5, 1, 4, 9, 11, 2, 13, 9, 5, 1, 18, 4, 2, 5, 8, 1, 1, 1, 1, 11, 9, 10, 11, 22, 20, 4, 9, 17)
$\overrightarrow{R_8}$	(1, 16, 17, 2, 19, 5, 11, 17, 13, 2, 6, 17, 6, 7, 8, 14, 21, 2, 4, 4, 1, 20, 1, 1, 1, 10, 1, 1, 1, 1, 1, 1, 4, 3, 2, 7, 3, 9, 7, 14, 13, 5, 5, 1, 4, 9, 11, 2, 13, 9, 5, 1, 18, 4, 2, 5, 8, 1, 1, 1, 1, 11, 9, 10, 11, 22, 20, 4, 9, 17, 11, 18, 1, 18, 2, 2, 21, 22, 4, 9)
$\overrightarrow{R_9}$	(8, 1, 1, 1, 5, 13, 7, 2, 14, 1, 6, 2, 3, 19, 7, 16, 23, 1, 1, 10, 16, 12, 1, 1, 1, 1, 9, 2, 1, 1, 5, 15, 6, 5, 6, 3, 9, 24, 22, 1, 2, 4, 11, 9, 1, 1, 7, 2, 18, 17, 1, 2, 15, 8, 15, 6, 1, 23, 8, 12, 1, 15, 1, 18, 7, 16, 8, 4, 25, 2, 1, 14, 1, 12, 2, 4, 11, 19, 4, 12, 12, 5, 7, 8, 1, 1, 1, 11, 9)
$\overrightarrow{R_{10}}$	(1, 16, 17, 2, 19, 5, 11, 17, 13, 2, 6, 17, 6, 7, 8, 14, 21, 2, 4, 4, 1, 20, 1, 1, 1, 10, 1, 1, 1, 1, 1, 1, 4, 3, 2, 7, 3, 9, 7, 14, 13, 5, 5, 1, 4, 9, 11, 2, 13, 9, 5, 1, 18, 4, 2, 5, 8, 1, 1, 1, 1, 11, 9, 10, 11, 22, 20, 4, 9, 17, 11, 18, 1, 18, 2, 2, 21, 22, 4, 9, 14, 11, 1, 7, 16, 4, 1, 1, 10, 4, 16, 4, 17, 9, 16, 1, 1, 2, 4)
$\overrightarrow{R_{11}}$	(1, 1, 1, 1, 20, 4, 13, 1, 2, 4, 17, 1, 9, 1, 6, 21, 6, 8, 1, 24, 11, 16, 7, 6, 13, 16, 2, 1, 2, 17, 4, 3, 18, 9, 1, 11, 3, 2, 4, 3, 6, 2, 9, 6, 8, 18, 5, 22, 1, 1, 1, 11, 23, 7, 4, 4, 1, 1, 11, 19, 3, 19, 5, 24, 4, 1, 11, 4, 26, 4, 6, 16, 15, 1, 6, 8, 1, 1, 11, 2, 5, 10, 1, 5, 15, 10, 2, 11, 3, 1, 1, 1, 8, 5, 11, 11, 6, 6, 4, 5, 25, 8, 4, 3, 8, 15, 1, 1, 1)
$\overrightarrow{R_{12}}$	(1, 1, 1, 1, 20, 4, 13, 1, 2, 4, 17, 1, 9, 1, 6, 21, 6, 8, 1, 24, 11, 16, 7, 6, 13, 16, 2, 1, 2, 17, 4, 3, 18, 9, 1, 11, 3, 2, 4, 3, 6, 2, 9, 6, 8, 18, 5, 22, 1, 1, 1, 11, 23, 7, 4, 4, 1, 1, 11, 19, 3, 19, 5, 24, 4, 1, 11, 4, 26, 4, 6, 16, 15, 1, 6, 8, 1, 1, 11, 2, 5, 10, 1, 5, 15, 10, 2, 11, 3, 1, 1, 1, 8, 5, 11, 11, 6, 6, 4, 5, 25, 8, 4, 3, 8, 15, 1, 1, 1, 17, 2, 7, 4, 11, 8, 9, 8, 26)

In Table 2, repetition details about 12 sample secret messages have been demonstrated (Notice that p is calculated by employing $\overrightarrow{R}$ for each S.). LZW coding performs compression by using the same codeword for the same repeating patterns. Accordingly, as the performance of LZW compression increases, we have a smaller $\overrightarrow{R'}$ that is compressed from $\overrightarrow{R}$. Therefore, the number of chosen email addresses via Latin square decreases. Since these email addresses are used in the stego cover with cover text, which represents denominator in Eq. (9), this increases the capacity.

The state of capacity versus character length of secret message has been demonstrated in Fig. 3. Here, vertical axis indicates capacity value in terms of percent (C%) while horizontal axis is indicating character length of secret message (n). By basing on Fig. 3, we can claim that capacity increases as the character length of secret message increases, since this represents nominator in Eq. (9). The line named as Log(capacity (%)) in the given graph demonstrates the bias curve of capacity. Logarithmic curve has been preferred due to the value of R^2 that is too close to 1.

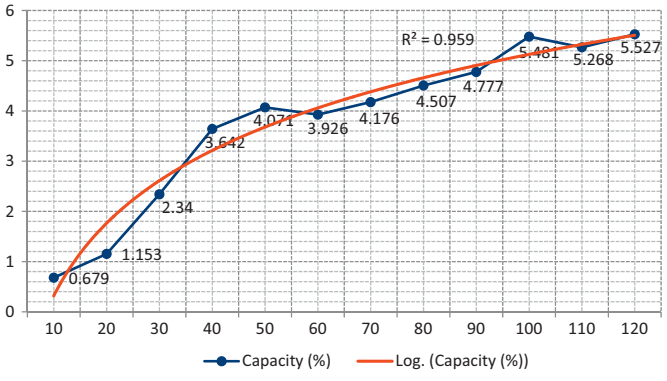


Fig. 3. Graph of capacity versus character length of secret message. (For interpretation of the references to color in text, the reader is referred to the web version of the article.)

5. Experimental results

The experiments for the proposed method have been performed in four steps by employing the following paragraph:

This paper presents a novel adaptive steganographic scheme that is capable of both preventing visual degradation and providing a large embedding capacity. The embedding capacity of each pixel is dynamically determined by the local complexity of the cover image, allowing us to maintain good visual quality as well as embedding a large amount of secret messages. We classify pixels into three levels based on the variance of the local complexity of the cover image. When determining which level of local complexity a pixel should belong to, we take human vision sensitivity into consideration. This ensures that the visual artifacts appeared in the stego image is imperceptible, and the difference between the original and stego image is indistinguishable by the human visual system. The pixel classification assures that the embedding capacity offered by a cover image is bounded by the embedding ca.

Step 1: The above paragraph contains 900 characters with spaces. We firstly divided the given paragraph into 3 parts each of which contains 300 characters. Thus, we obtain 3 secret messages (S_1, S_2 and S_3). The parts have been indicated in underlined, bold and italic styles.

Step 2: For an unbiased and a detailed investigation, we divided the length (n) into 16 intervals. For every secret message, the first 12 intervals have been obtained by incrementing the length 10 by 10. The 13th interval has been obtained by adding 30 to the length of the 12th one and the last 3 intervals have been obtained by incrementing the length 50 by 50. Thus the length (n) of each secret message changes between 10 and 300, as seen in Table 3.

Step 3: The following operations have been performed in order to support an unbiased and a detailed investigation for each secret message:

3.1 16 sub parts of S_1 have been constructed from beginning to the end, sequentially by conforming the given intervals intervals in Table 3.

3.2 16 sub parts of S_2 have been constructed from end to the beginning, sequentially by conforming the given intervals in Table 3.

Table 3
Implementation results.

n	Capacity (%)			
	C_1	C_2	C_3	Average capacity (%)
10	1.165	0.667	1.137	0.9897
20	2.257	1.546	1.446	1.7497
30	3	2.338	2.754	2.6973
40	3.565	2.672	1.894	2.7103
50	4.382	3.170	3.586	3.7127
60	4.788	3.858	3.934	4.1933
70	5.131	3.875	3.655	4.2203
80	5.590	3.835	5.698	5.041
90	5.821	4.083	5.107	5.0037
100	5.945	4.623	5.387	5.3183
110	6.141	4.462	5.475	5.3593
120	6.359	5.349	5.165	5.6243
150	6.931	5.740	7.146	6.6057
200	7.150	6.159	6.009	6.4393
250	7.365	6.500	7.278	7.0477
300	6.860	6.793	7.473	7.042

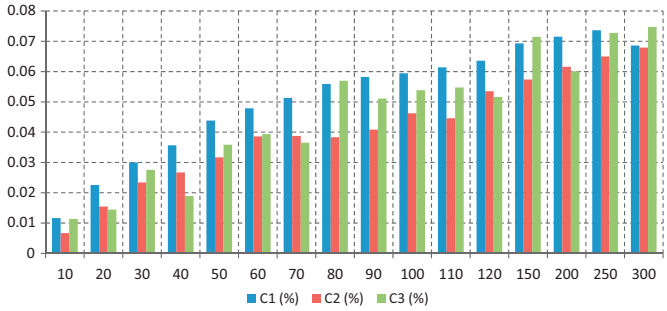


Fig. 4. Capacities (%) of S_1, S_2 and S_3.

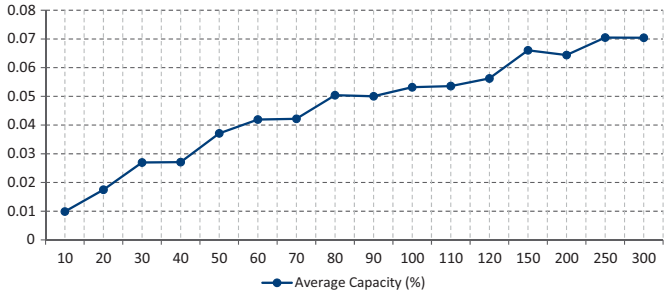


Fig. 5. Average capacity (%) distribution.

3.3 16 sub parts of S_3 have been constructed by concatenating randomly chosen pieces and also by conforming the given intervals in Table 3.

Thus, we have a total of 48 sub parts.

Step 4: We employed the proposed method and we calculated capacity values via Eq. (9) for each sub part of each secret message.

Table 3 contains the information regarding with the explained experimental steps. Length of each sub part has been provided in n column. For each sub part of each secret message, the calculated bit rates have been provided in the capacity (%) column. Finally, average bit rate for each sub part has been provided in average capacity (%) column.

Based on Table 3, capacity (%) of each sub part has been indicated in Fig. 4 and average capacity (%) distribution of the proposed method has been demonstrated in Fig. 5. In Fig. 4, horizontal axis indicates the length (n) while the vertical axis indicates capacity value in terms of percent. Similarly, in Fig. 5, horizontal axis of the graph indicates length (n) of the secret message while the vertical axis indicates the capacity values. As it can be seen in the provided graph in Fig. 5, generally, capacity increases as the character length increases. Thus, the disadvantage of character length on capacity can be an advantage by means of the proposed LZW based text steganography method.

In Table 4, the proposed method has been compared to the other contemporary methods in the literature. This comparison has been performed in terms of capacity. Capacities of the accessible methods, like TEXTO that works a simple substitution cipher and like Mimic functions that produces grammatically correct but meaningless text, have been calculated by employing the given sample message in the following paragraphs.

Capacities of Nicetext and Winstein's scheme have been provided by basing on the given samples in the referred articles. Capacities of Murphy's and Nakagava's schemes which are synonyms-based approaches have been reported in the referred articles.

Capacity of Stutsman's scheme that hides a message in naturally encountered errors of a machine translation has been provided by basing on the referred article. Capacity of another translation based approach, Topkara's scheme, has been provided by basing on the samples in the referred article.

Capacities of Sun et al.'s L-R scheme and Wang et al.'s scheme have been calculated by basing on the samples in Wang et al. (2009a) in UNICODE format since they deal with Chinese language. Finally, capacity of Listega that camouflages data by using textual list has been provided by basing on the sample in the referred article.

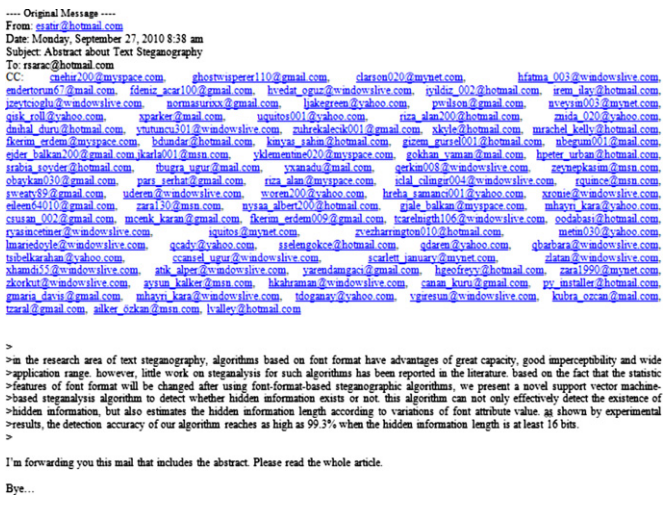
In the proposed LZW based text steganography method, the stego cover consists of naturally generated cover text (T^*) and email addresses (K_2; the set of chosen and modified email addresses as stego keys) in order to show the stego cover as a forward mail as seen in Fig. 6. The given bit rate in Table 4 has been calculated via Eq. (9), by considering both of these stego covers. According to Table 4, the proposed LZW based text steganography method has increased the capacity to 6.925% by considering the given example below. This obtained capacity value provides a significant increment for secret message with the length of 200 characters.

Finally, we provide a secret message and the constructed stego cover in order to illustrate the output of the proposed method (Also, we provide an illustrative example in Annex 2). A sample secret message has been given below:

> "behind using a cover text is to hide the presence of secret messages the presence of embedded messages in the resulting stego text cannot be easily discovered by anyone except the intended recipient"

Table 4
Comparison of bitrates.

Method	Capacity (%)	Explanation
Mimic functions (Wayner, 1992, 2002)	1.27	Calculated by employing the following sample secret message at http://www.spamimc.com
NICETEXT (Chapman and Davida, 1997, 2001, 2002)	0.29	Provided by basing on the samples in the referred articles
Winstein (Winstein, 1999, 2008)	0.5	Provided by basing on the samples in the referred articles
Murphy et al. (Murphy and Vogel, 2007)	0.30	Reported in the referred article
Nakagava et al. (Nakagawa et al., 2001)	0.12	Reported in the referred article
Translation based (Stutsman et al., 2006)	0.33	Noted by the authors in the referred article
Confusing (Topkara et al., 2007)	0.35	Provided by basing on the samples in the referred article
Sun et al.'s L-R scheme (Sun et al., 2004)	2.17	Calculated in UNICODE format by basing on the given sample in Wang et al. (2009a)
Wang et al. (Wang et al., 2009a)	3.53	Calculated in UNICODE format by basing on the given sample in Wang et al. (2009a)
Listega (Desoky, 2009)	3.87	Provided by basing on the example in the referred article
TEXTO (Maher, 1995)	6.91	Calculated by employing the following sample secret message at http://www.eberl.net/cgi-bin/stego.pl
The proposed LZW based text steganography method	6.92	Calculated by employing the following sample secret message



---- Original Message ----
From: esatir@hotmail.com
Date: Monday, September 27, 2010 8:38 am
Subject: Abstract about Text Steganography
To: rsarac@hotmail.com
CC: cnehir200@myspace.com, ghostwusperer110@gmail.com, clarson020@mynet.com, hfatma_003@windowslive.com, endertorun67@mail.com, fdeniz_acar100@gmail.com, kvedat_oguz@windowslive.com, tvildiz_002@hotmail.com, irem_day@hotmail.com, jzeyticioglu@windowslive.com, normasurixx@gmail.com, ljakegreen@yahoo.com, pwilson@gmail.com, nveysm003@mynet.com, qisk_roll@yahoo.com, xparker@mail.com, uquotes001@yahoo.com, riza_alan200@hotmail.com, zmda_020@yahoo.com, dnihal_duru@hotmail.com, ynutuncu301@windowslive.com, zuhrekalecik001@gmail.com, xkyle@hotmail.com, mrachel_kelly@hotmail.com, fkerim_erdem@myspace.com, bdundar@hotmail.com, kmyas_sahin@hotmail.com, gizem_gursel001@hotmail.com, nbegum001@mail.com, egder_balkan200@gmail.com jkarla001@msn.com, yklementine020@myspace.com, gokhan_yaman@mail.com, hpeter_urban@hotmail.com, trabia_snyder@hotmail.com, ttugra_ugur@mail.com, yxanadu@mail.com, qerkin008@windowslive.com, zeynepkasim@msn.com, obaykan030@gmail.com, pars_serhat@gmail.com, riza_alan@myspace.com, silal_cilingir004@windowslive.com, rquince@msn.com, sweaty89@gmail.com, uderen@windowslive.com, woren200@yahoo.com, hreha_ramanct001@yahoo.com, xronie@windowslive.com, eileen64010@gmail.com, zara130@msn.com, nysaa_albert200@hotmail.com, gtale_balkan@myspace.com, mhayri_kara@yahoo.com, crsusan_002@gmail.com, mcenk_karan@gmail.com, fkerim_erdem009@gmail.com, tcarelnigth106@windowslive.com, oodabasi@hotmail.com, ryasincetiner@windowslive.com, squtos@mynet.com, zvesharrington010@hotmail.com, metin030@yahoo.com, lmariedoyle@windowslive.com, qcady@yahoo.com, sselengokce@hotmail.com, qdaren@yahoo.com, qbarbara@windowslive.com, tubelkarahan@yahoo.com, ccansel_ugur@windowslive.com, scarlett_january@mynet.com, zlatan@windowslive.com, xhamdi55@windowslive.com, atik_alper@windowslive.com, yarendamgaci@gmail.com, hgeofrey@hotmail.com, zara1990@mynet.com, zkorkut@windowslive.com, ayyun_kalker@msn.com, hkaharman@windowslive.com, canan_kuru@gmail.com, py_installer@hotmail.com, gmaria_davis@gmail.com, mhayri_kara@windowslive.com, tdoganay@yahoo.com, vguresun@windowslive.com, kubra_ozcan@mail.com, tzaral@gmail.com, aalker_ozkan@msn.com, lvalley@hotmail.com

>
>in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide >application range. however, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic >features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine->based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of >hidden information, but also estimates the hidden information length according to variations of font attribute value. as shown by experimental >results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.
>

I'm forwarding you this mail that includes the abstract. Please read the whole article.

Bye...

Fig. 6. The constructed stego cover.

This message has 200 characters with spaces and without quotation marks. According to the embedding phase (Step 4), the chosen cover text (T^*) has been given below:

> "in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. however, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits."

The constructed stego cover has been demonstrated in Fig. 6. The stego cover has been arranged as a forward mail platform not to raise suspicion. As seen in Fig. 6, stego cover consists of the chosen cover text given above and the chosen and modified email addresses (K_2). Besides, the employed Latin square has been given in Annex 1. According to Eq. (9), capacity has been computed as 6.92% for this example.

6. Conclusion

In this section firstly, we aim to explain the advantages and disadvantages of the proposed method. An advantage of the proposed method is not being language specific. The method can be applied to any language by reconstituting the text database and adapting the Latin Square to the concerning language, if necessary (for e.g. Chinese and Arabic languages). Another advantage of the proposed method is protecting the originality of the cover media while communicating. The method does not produce noise in order to hide secret information. It changes neither meaning nor format of the cover text. In the proposed method, the stego cover is a forward mail platform that contains two cover medium. One of them is the naturally generated cover text. So the text is meaningful, syntactically and grammatically correct and legitimate. Another is the chosen email addresses in order to show the mail as a forward mail platform. There is not any format or constraint on generating email addresses (numbers, repeating characters can be used) and it is not necessary for them to be meaningful. So they do not raise suspicion. Because of these specifications, the proposed method is strong against OCR programs and retyping. Additionally, security of the proposed method has been supported by means of the employed stego keys. Besides, Combinatorics-based coding and LZW compression have also been employed for this purpose.

As future work, we aim to investigate the effects of other lossless data compression algorithms like Huffman Coding and Arithmetic Coding, firstly on capacity. For a more significant capacity increment, we aim to use shorter naturally generated texts in text base. Finally by increasing the variety of text base with these shorter texts, we aim to obtain the desired randomness in case of hiding similar patterns.

Annex 1. Arranged Latin Square

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
3	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
4	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
5	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
6	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
7	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
8	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
9	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
10	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
11	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
12	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
13	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
14	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
15	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
16	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
17	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
18	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
19	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
21	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
22	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
23	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
24	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
25	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
26	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Annex 2. An illustrative example

Secret message: behind using a cover

$S = \{b, e, h, i, n, d, , u, s, i, n, g, , a, c, o, v, e, r\}$

In Step 1 and Step 2, we estimate $\overrightarrow{\Delta D}$ for every *Text* in *T*.

In Step 3, we obtain *E* and *R* by calculating the exceeding number and reconstructing $\overrightarrow{\Delta D}$ for every *Text* in *T*.

In Step 4, we find the maximum dual pattern repetition number (*p*) as 8. Accordingly,

$\overrightarrow{\Delta D} = (55, 2, 6, 1, 1, 1, 2, 28, 11, 21, 1, 1, 1, 7, 5, 3, 39, 80, 1, 1)$

$\overrightarrow{R} = (3, 2, 6, 1, 1, 1, 2, 2, 11, 21, 1, 1, 1, 7, 5, 3, 13, 2, 1, 1)$

$\overrightarrow{E} = (2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0)$

Chosen cover text (T^*) from *T*:this paper presents a novel steganography scheme suitable for hindi text. it can be classified under text steganography. conveying information secretly and establishing a hidden relationship between the message and its counterpart has been of great interest since very long time ago. methods of steganography are mostly applied on images, audio, video and text files. during the process characteristics of these methods are to change in the structure and features so as not to be identifiable by human eye. text documents are the best examples for this .this paper presents a novel hindi text steganography, which uses hindi letters and its diacritics and numerical code. this method is not only useful to hindi text but also to all other similar indian languages.

In Step 5, LZW dictionary:

(1) 1	(15) 15	(29) 6, 1
(2) 2	(16) 16	(30) 1, 1
(3) 3	(17) 17	(31) 1, 1, 2
(4) 4	(18) 18	(32) 2, 2
(5) 5	(19) 19	(33) 2, 11
(6) 6	(20) 20	(34) 11, 21
(7) 7	(21) 21	(35) 21, 1
(8) 8	(22) 22	(36) 1, 1, 1
(9) 9	(23) 23	(37) 1, 7
(10) 10	(24) 24	(38) 7, 5
(11) 11	(25) 25	(39) 5, 3
(12) 12	(26) 26	(40) 3, 13
(13) 13	(27) 3, 2	(41) 13, 2
(14) 14	(28) 2, 6	(42) 2, 1

Bit Steram is obtained by $\left(\overrightarrow{R'}\right)_2$:

0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, **0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0**, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, **0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0**, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, **0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1**, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0

If the number of bits is not a multiple of 12, then the bit steram is completed the nearest multiple of 12 by adding 0 in this case. Here since the bit stream contains 96 bits, no completion is necessary.

In Step 6 – by basing on Eqs. (3)–(5) – *x*, *y* and *z* are obtained as follows:

Mail address no.	*x* is used for the first letter email address of via K_1	*y* is used for the second letter of email address via K_1	*z* is used for email address extension via *A*
1	2	24	0
2	15	24	7
3	1	17	2
4	4	13	5
5	10	7	1
6	3	5	5
7	1	26	5
8	1	20	6

In Step 7, the chosen email addresses by employing *Latin square*, are provided. Besides by using $\overrightarrow{E}$ and *A* these email addresses are modified and construction of K_2, is completed:

K_2 = {bxenon@hotmail.com, py_installer@mynet.com, csusan_88001@yahoo.com, gpirsu300@mail.com, nkaragul@gmail.com, hjersey@mail.com, gfergie65@mail.com, harun_duru@myspace.com}

In Step 8, stego cover is arranged by combining K_2 and T^*:

----Original Message----
From: esatir@hotmail.com
Date: Monday, September 27, 2010 8: 38 am
Subject: Abstract for Text Steganography
To: rsarac@hotmail.com
CC: bxenon@hotmail.com, py_installer@mynet.com, csusan_88001@yahoo.com, gpirsu300@mail.com, nkaragul@gmail.com, hjersey@mail.com, gfergie65@mail.com, harun_duru@myspace.com
>

>information secretly and establishing a hidden relationship between the message and its counterpart has been of great interest since
>very long time ago. methods of steganography are mostly applied on images, audio, video and text files. during the process
>characteristics of these methods are to change in the structure and features so as not to be identifiable by human eye. text documents
>are the best examples for this. this paper presents a novel hindi text steganography, which uses hindi letters and its diacritics and
>numerical code. this method is not only useful to hindi text but also to all other similar indian languages.
>
I'm forwarding you this abstract. Please read the whole article.

Bye.

For this example, the calculated capacity value is: $C = 1.71\%$.

References

Aabed, M.A., Awaideh, S.M., Abdul-Rahman, M.E., Gutub, A., 2007. Arabic diacritics based steganography. In: IEEE International Conference on Signal Processing and Communications (ICSPC 2007), Dubai, UAE, November 24–27, pp. 756–759.

Al-Bahadili, H., 2008. A novel lossless data compression scheme based on the error correcting Hamming codes. Computers & Mathematics with Applications 56 (1), 143–150.

Al-Haidari, F., Gutub, A., Al-Kahsah, K., Hamodi, J., 2009. Improving security and capacity for arabic text steganography using 'Kashida' extensions. In: The 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA – 2009), Rabat, Morocco, May 10–13, pp. 396–399.

Bailey, K., Curran, K., 2006. An evaluation of image based steganography methods using visual inspection and automated detection techniques. Multimedia Tools and Applications 30 (1), 55–58.

Chang, C., Kieu, T.D., 2010. A reversible data hiding scheme using complementary embedding strategy. Information Sciences 180 (16), 3045–3058.

Chapman, M., Davida, G., 1997. Hiding the hidden: a software system for concealing cipher text as innocuous text. The Proceedings of the International Conference on Information and Communications Security. Lecture Notes in Computer Science, vol. 1334. Springer, Beijing, pp. 335–345.

Chapman, M., Davida, G.I., 2001. A practical and effective approach to largescale automated linguistic steganography. Proceedings of the Information Security Conference (ISC '01), Lecture Notes in Computer Science, vol. 2200. Springer, Malaga, pp. 156–165.

Chapman, M., Davida, G.I., 2002. Plausible deniability using automated linguistic steganography. In: Davida, G., Frankel, Y. (Eds.), International Conference on Infrastructure Security (InfraSec '02). Lecture Notes in Computer Science, vol. 2437. Springer, Berlin, pp. 276–287.
Colbourn, C., 1984. The complexity of completing partial latin squares. Discrete Applied Mathematics 8, 151–158.
Desoky, A., 2009. Listega: list-based steganography methodology. International Journal of Information Security 8 (4), 247–261.
Easton, T., Gary Parker, R., 2001. On completing latin squares. Discrete Applied Mathematics 113 (2–3), 167–181.
Galambos, G., Bekesi, J., 2002. Data Compression: Theory and Techniques. Department of Informatics, Teacher's Training College, Database and Data Communication Network Systems, vol. 1. Elsevier Science, USA, Copyright 2002.
Gutub, A., Fattani, M., 2007. A novel arabic text steganography method using letter points and extensions. In: WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria, May 25–27, pp. 28–31.
Jun, L., Tong, W., Daxin, L., 2011. Research on ordinal properties in combinatorics coding method. Journal of Computers 6 (1), 51–58.
Liang, J.Y., Chen, C.S., Huang, C.H., Liu, L., 2008. Lossless compression of medical images using Hilbert space-filling curves. Computerized Medical Imaging and Graphics 32 (3), 174–182.
Lou, D., Wu, N., Wang, C., Lin, Z., Tsai, C.S., 2010. A novel adaptive steganography based on local complexity and human vision sensitivity. Journal of Systems and Software 83 (7), 1236–1248.
Maher, K., 1995. TEXTO. ftp://ftp.funet.fi/pub/crypt/steganography/texto.tar.gz.
Murphy, B., Vogel, C., 2007. The syntax of concealment: reliable methods for plain text information hiding. In: Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents.
Nakagawa, H., Sampei, K., Matsumoto, T., Kawaguchi, S., Makino, K., Murase, I., 2001. Text information hiding with preserved meaning—a case for Japanese documents. IPSJ Transactions 42 (9), 2339–2350. Originally published in Japanese. A similar paper by the first author in English. http://www.r.dl.itc.u-tokyo.ac.jp/nakagawa/academic-res/finpri02.pdf (accessed 04.06.08).
Park, J., Lee, S., 2009. Forensic investigation of Microsoft PowerPoint files. Digital Investigation 6 (1–2), 16–24.
Por, L.Y., Wong, K., Chee, K.O., 2012. UniSpaCh: a text based data hiding method using unicode space characters. Journal of Systems and Software, http://dx.doi.org/10.1016/j.jss.2011.12.023.
Sajedi, H., Jamzad, M., 2010. BSS: boosted steganography scheme with cover image preprocessing. Expert Systems with Applications 37 (12), 7703–7710.
Stutsman, R., Atallah, M., Grothoff, C., Grothoff, K., 2006. Lost in just the translation. In: Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France, April 23–27, pp. 338–345.
Sun, X.M., Luo, G., Huang, H.J., 2004. Component-based digital watermarking of Chinese texts. In: Proceedings of the 3rd International Conference on Information Security, Shanghai, China, pp. 76–81.
Topkara, M., Topkara, U., Atallah, M.J., 2007. Information hiding through errors: a confusing approach. In: Proceedings of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, January 29–February 1.
Wang, Z., Chang, C., Lin, C., Li, M., 2009a. A reversible information hiding scheme using left-right and up-down Chinese character representation. Journal of Systems and Software 82, 1362–1369.
Wang, Z.H., Kieu, T.D., Chang, C.C., Li, M.C., 2009b. Emoticon-based text steganography in chat. In: Proceedings of 2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA 2009), vol. 2, Wuhan, China, pp. 457–460.
Wayner, P., 1992. Mimic functions. Cryptologia XVI (3), 193–214, http://dx.doi.org/10.1080/0161-119291866883.
Wayner, P., 2002. Disappearing Cryptography, 2nd ed. Morgan Kaufmann, Menlo Park, pp. 81–128.
Winstein, K., 1999. Lexical steganography through adaptive modulation of the word choice hash, Secondary education at the Illinois Mathematics and Science Academy, January. http://alumni.imsa.edu/~keithw/tlex/lsteg.ps (accessed 15.04.08).
Winstein, K. Lexical steganography. http://alumni.imsa.edu/~keithw/tlex (accessed 03.08.08).

Prof. Dr. Hakan Isik was born in Adana – Turkey on 19th of July 1968. He completed his under graduate degree in Gazi University, Technical Education Faculty, Computer and Electronic Education department, in 1990 and his graduate in Gazi University, Electronic and Computer Education Department, in 1998 and his doctorate in Gazi University, Electronic and Computer Education Department, in 2002. Currently, he is working as head of Electronic Engineering Department in Selcuk University, Technology Faculty. He is interested in medical electronic and instrumentation in medicine.

Research Assistant Esra Satir was born in Konya – Turkey 26th of April 1983. She completed her undergraduate degree in Gazi University, Technical Education Faculty, Computer and Electronic Education Department, in 2005. She completed her graduate degree in Selcuk University, Technical Education Faculty, Computer and Electronic Education Department, in 2009. She has been executing her doctorate education in Selcuk University, Faculty of Engineering, Computer Engineering Department since 2009. Currently, she is working as a research assistant in Selcuk University, Technical Education Faculty, Electronic and Computer Education Department. She is interested in artificial intelligence, data compression, information security and especially steganography.