ARTICLE TEMPLATE

An Email-based High Capacity Text Steganography using Repeating Characters

Mansoor Fateh[a] and Mohsen Rezvani[a]

[a]Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran,
{mansoor_fateh,mrezvani}@shahroodut.ac.ir

ARTICLE HISTORY
Compiled August 18, 2018

ABSTRACT
Email steganography is one of the subcategories of text steganography. This subcategory is noted due to the widespread use of emails for communication. In this paper, we propose a new method of email steganography with high capacity and high security. Furthermore, since no surplus information will be added to the cover text, we can use any kinds of email bodies as our cover text. In this way, Email body wont be suspicious. For having high security, we offer several stego keys. In the first step of this work, the secret message will be compressed by using one of the lossless compression algorithms named LZW to bit streams. Then in every step, due to number of email body' characters, part of bit stream will be selected and by using the email addresses will be created. The capacity of a purposed method for the common example used by other studies has been computed as 10.6, which proved a significant increment in term of capacity.

KEYWORDS
text steganography, email steganography, hiding rate.

1. Introduction

Nowadays the growth of information and communication technology allows various digital contents, such as multimedia files, to be transmitted through a public network such as the Internet. However, such transmission raises some security and privacy issues, as the published contents are available for every user over the public network. Moreover, establishing a private network is an extensive and time-consuming task. Data hiding is an alternative solution for data transmission over a public network such as the Internet [1].

Information hiding techniques can be divided into three categories: cryptography, watermarking and steganography [2,3]. Hiding information through cryptography techniques has two major shortcomings: 1) transferring the encrypted data is prohibited in some governments, such as dictatorial regimes; 2) encrypted data attracts the attention of interceptors which leads to denying any confidential communication [4]. The watermarking techniques can be used for copyright protection. For example, people can conceal information into their products visibly or imperceptibly. Furthermore, watermarking can be used as a broadcast monitoring and deal tracking. Watermarking

CONTACT Mansoor Fateh. Email: mansoor_fateh@shahroodut.ac.ir

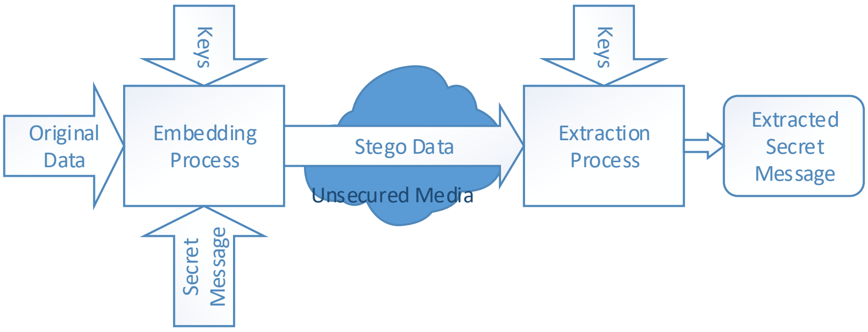


Figure 1. Process of steganography.

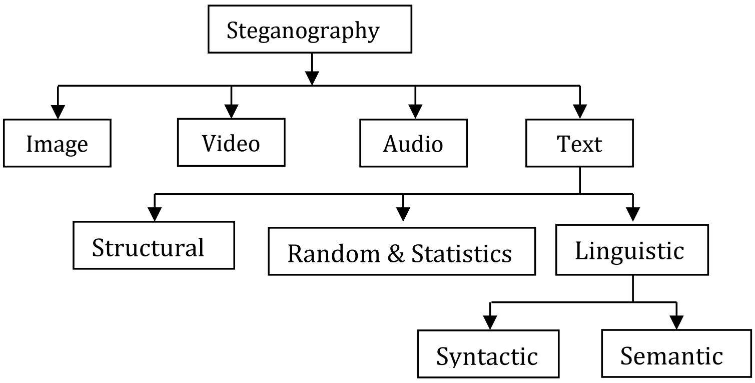


Figure 2. Different categories of steganography approaches [5].

may not have security property. Steganography is the secret hiding data (hidden data) within plain objects (cover data) to produce stego data and transmits the stego data over a public network. The guarantees that only sender and receiver can inform about the existence of the secret message. Cryptography unlike stenography does not require medium to hide the secret data. The process of the steganography shown in Figure 1. Before steganography is often done cryptography. That is, the secret data is encrypted before embedding [5]. Steganography also requires a key to embed and to extract the secret data.

There are three important and necessary factors in steganography: 1) the cover data and hidden data must be identical against statistical attacks; 2) the information hiding process must be in a way that the cover data and hidden data are human-indistinguishable; 3) considerable capacity of secret data must be embedded into the cover. In other words, the integrity of the communication media must be kept. Steganography methods hide information within some media such as image [6, 7], video [8], audio [9], and text [10]. This categorization is shown in Figure 2.

A very low local change in texts becomes the text steganography a difficult task. Since texts have been widely used in digital media, there is plenty of research proposed text steganography approaches. An email is one of the most popular communication tools. An email can be sent to many recipients. Email-based steganography is one of the most popular techniques for the text steganography. This technique hides the secret message within the email body and email addresses. Email addresses are used to hide the secret data, regardless they are valid addresses or not. However, its relatively high number may attract email recipients attention. There is some research conducted to propose email-based steganography schemes [11–13]. Most of these approaches apply some limitations on the text, and the email body as well. Moreover, they either add some noise or extra information to the text.

In this paper, we proposed a novel Email steganography method in which the secret message is hidden within several email addresses generated through the body of the email. The proposed method employs two security key to improve the security level of the steganography. Moreover, it generates less number of email addresses comparing to the state of the art methods proposed in email steganography that it also improves its security level. The proposed scheme first converts the secret message into a bit-stream using the LZW algorithm [14] and then it embeds the obtained bit-stream into the receivers' addressed using the steganography keys. Such conversion at the beginning of the proposed method helps to have no limitation on the type of cover text. Therefore, the proposed method can be employed over every text without applying any limitations, adding any noise, nor inserting extra information into the email body.

The rest of the paper is structured as follows. Section 2 provides the related work. Section 3 presents our novel steganography scheme. Section 4 describes our experimental result. Finally, the conclusion is drawn in Section 5.

2. Related Work

Three different types of the text steganography are based on structure, random and statistics, and linguistics. The structural steganography is based on changing the structure of a text, for example, hiding a secret message in text by changing the font size of the white spaces in the text, within the word processor [15], manipulating the font and color of characters within cells of a spreadsheet [16], and exchanging words and lines [17]. In [18] each English letter along with the full stop letter is assigned to a string of seven characters. Each character in the secret message is hidden within one the mapped characters by changing the spaces among characters. In this type of text steganography, transforming the text from one space to another diminishes the secret information.

The random and statistics steganography employs statistical features of the secret message to automatically generate the cover text [19]. The proposed approach in [20] uses the Omega network structure and the words in a dictionary to embedding and extraction the secret message. One problem with this approach is that using similar dictionaries several times for replacing the letters reduces the performance of the system and also generates meaningless sentences.

The linguistic steganography modifies the syntax and semantics attributes of text. To change the syntax attributes, one can insert punctuation marks incorrect positions in the text [21]. In contrast, for changing the semantics attributes, one can replace the synonym words in a text to hide a secret message [22]. It is noted that replacing a synonym word can cause unexpected results in the text. Thus, finding a suitable replacement mechanism with high reliability is a challenging problem in the linguistic steganography. Chang et al. [23] proposed a linguistic steganography method that employs a Combinatory Categorical Grammars Parser.

In addition to above categories, there are several different approaches used for information hiding in the literature. Nagarhalli in [24] employs short messages for the cover text and hides information within emoticons. The main shortcoming of this method is its low capacity. Garg in [25] proposed a text steganography method that uses HTML documents as the cover medium to hide secret messages. This method provides integrity between cryptography and steganography that improve its security. Majumder et al. in [26] use a summary of the text to propose a text steganography approach. This method suffers from low security due to the lack of the integrity of the generated summaries. Por et al. in [27] introduce the UniSpaCh system which hides the information over the various white spaces within a Microsoft Word document.

Email-based steganography is one of the most popular techniques for the text steganography. The main idea in [1, 5, 11, 12, 28] is to send an email to a huge number of email addresses. In [1, 5, 11, 12] the email addresses are used to generate the steganography key. Tohari et al. [13] randomly generate the email addresses to improve the performance of the proposed steganography method. They also showed that such random method improves the security level of the steganography. Satir and Isik leverage the LZW compression along with a set of 48 sample texts to propose an Email-based steganography [1]. The main problem in the proposed method is that it is limited to the sent message due to its low capacity. In the other hand, the method uses several steganography keys which improve its security level. Kumar et al. in [11] employ a combination of compression algorithms as BWT+MTF+LZW to increase the capacity of the steganography. Later on, they improved the capacity of their steganography scheme from 7.03% to 7.21%using the Huffman compression [12]. Aruna et al. in [28] proposed different Email steganography in which the secret message is hidden by colouring the characters in the cover text. The proposed method has a high level of capacity because of the huge number of available colours. However, it suffers from a low security because the method changes the original text.

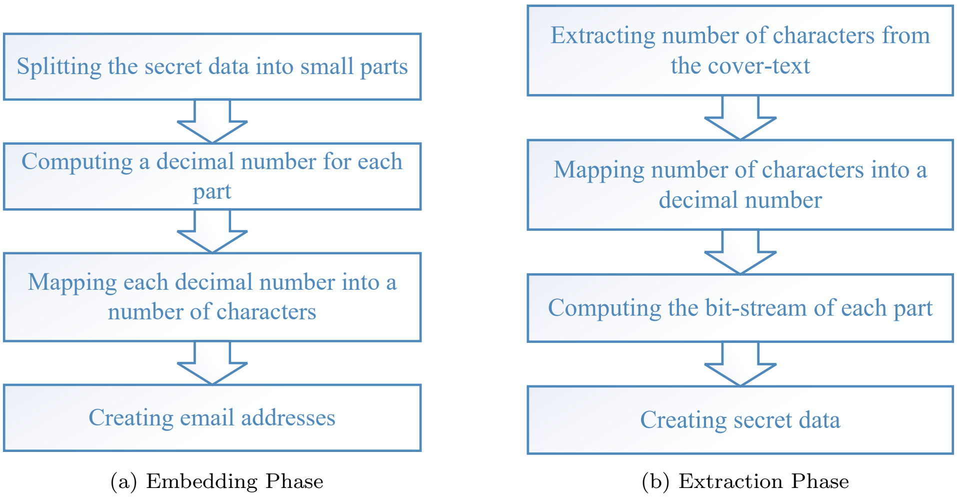
In this paper, we proposed a novel Email steganography method in which the secret message is hidden within a number email addresses generated through the body of the Email. The message body is simultaneously sent to the generated email addresses as well as the receiver address. This helps us to protect the identity of the original receiver. Moreover, the original receiver is the only person can extract the secret message from other email addresses in the receipt part of the email. Also, our method neither has any limitations on the body of the email nor inserts any noise or extra information into the body.

3. The Proposed Scheme

In this section, we describe the details of our Email-based text steganography. The proposed scheme hides the secret data within a number email addresses. The steganography scheme includes two phases: embedding and extraction explained in the rest of this section. Figure 3 shows the main steps of these two phases in the proposed scheme.

The main idea of the proposed approach is mapping the secret data to several characters. In the embedding phase, the secret data is first divided into small parts. Then, a decimal number for each part is computed. After that, each decimal number is mapped to some characters in the cover text. Based on the generated decimal number, the cover text may be traversed several times. Finally, an email address is created using the number of such traverses, the number of traversed sentences and the number of characters within the last sentence.

In the first step of the extraction phase of our scheme, the number of characters is obtained from each email address. The number of characters is then mapped into a decimal number. After that, the decimal number is converted into a bit-stream. Finally, we achieve the secret data from contamination of the bit-streams obtained for each decimal number.


Splitting the secret data into small parts
Computing a decimal number for each part
Mapping each decimal number into a number of characters
Creating email addresses
Extracting number of characters from the cover-text
Mapping number of characters into a decimal number
Computing the bit-stream of each part
Creating secret data

(a) Embedding Phase (b) Extraction Phase

Figure 3. Main steps in the proposed scheme.

3.1. Embedding Phase

We define several constants and keys shared between the sender and receiver as follows:

- $\hat{x}$: The maximum number of times a text can be used.
- $\hat{y}$: The maximum number of sentences in a text can be used in our computations.
- $\hat{z}$: The maximum number of characters can be used in our computations.
- A: A set of eight email prefixes. We assign a three-bit digit to each prefix. Here is the list of prefixes and their assigned digits:
  A=”gmail.com” (000),”hotmail.com” (001),”yahoo.com” (010),”rediffmail.com” (011), ”btinternet.com” (100),”aol.com” (101),”msn.com”(110),”verizon.net” (111)
- S: A set of sentences which are used in the message body (Email body) and denoted as $S = s_1, s_2, \dots, s_{\hat{y}}$, where s_i is a sentence. It is worth noting that a message body consists of several sentences that are grouped together.

We also define some symbols will be used in our scheme in Table 1. The embedding phase in our steganography scheme contains 12 steps, describes as follows:

Step 1 In the first step, the secret message is converted to a stream numbers using the LZW compression [14]. LZW is lossless dictionary based adaptive compression algorithm. LZW compression works by reading a sequence of symbols, grouping the symbols into strings, and converting the strings into integer codes. Decompression converts codes back into strings, obtaining the original symbols.6. After that, these numbers are converted to binary and a bit stream is generated from the secret message.

Step 2 The number of characters in the message body is computed and denoted as *total_char*.

Step 3 Starting from the least significant bit in the bit-stream of the secret message, compute the decimal value of the bit-stream until the computed decimal value is not greater than $\hat{x} \times total\_char$. Also, store the index stopped over the bit-stream in the index.

Step 4 Set $index \leftarrow index - 1$. If the bit on the index is zero, decrement the index value until reaching to a one value. Also, the number of zero values traversed in this step is stored in *count_zero*.

Step 5 Compute the values of x, y, and z as follows:

$$x = \frac{decimal}{total\_char} \tag{1}$$

$$x = \frac{decimal}{\mathrm{mod}(total\_char)} \tag{2}$$

Now, we read the message body from the beginning and sentence by sentence. Compute the sum of the length of traversed sentences until the summation value is not greater than m. Now set the value of y equals to the number of traversed sentences minus 1.

$$\sum_{i=1}^{y} |s_i| \leq m \leq \sum_{i=1}^{y+1} |s_i| \tag{3}$$

Compute the value of z as follows:

$$z = m - \sum_{i=1}^{y} |s_i| \tag{4}$$

Step 6 As mentioned, $\hat{z}$ is the maximum number of characters. Since z might become greater than $\hat{z}$, the value of z is set by Equation (6). The assigned value for each category is obtained from Table 1. If there is no value for a category in an email, its value is set to 1.

$$cate = \frac{z}{\hat{z}} \tag{5}$$

$$z = \frac{z}{\mathrm{mod}\hat{z}} \tag{6}$$

Step 7 Compute the bit-streams for values of x, y, and z. Now concatenate the three bit-streams to make the combined bit-stream.

Table 1. Symbols are used for each category.

Category	Symbol
1	0
2	_
3	20

Step 8 Now extract the three most significant bits in the combined bit-stream and use the three-bit digit to find the email prefix from key A.

Step 9 The remaining bits in the combined bit-stream are divided into some 4-bit segments with assigning a continuous index value to each segment. Now, an English letter is assigned to each segment computed as follow:

$$char = \frac{(i \times 16) + d}{\bmod 26} \tag{7}$$

Where i denotes the assigned index to the segment and d is the equivalent decimal value of four bits of the segment. The char value is in the range [0, 25] and indicates an English letter.

Step 10 Now we use Equation (7) to assign an English letter to $count\_zero$ and concatenate the letter to the end of the message obtained from Step 9. To this end, we set $i = 0$ and $d = count\_zero$ in Equation (7).

Step 11 Create a meaningful email address using the generated characters in Step 10 and the email prefix generated in Step 8.

Step 12 Repeat steps 3 to 11 by incrementing the index value. Above steps generate all email addresses included the secret message. The cover text is simultaneously sent to the generated email addresses as well as the original receiver.

3.2. Extraction Phase

The extraction phase of our Email-based text steganography scheme consists of 7 steps described as follows:

Step 1 In the first step of the extraction phase, we extract the values of x, y, and z for each email address in the received email. It is noted that the receiver knows the number of letters containing the secret message in the each email address. The receiver obtains the decimal value of each letter as follows:

$$\theta = \left\lceil \frac{16i - char}{26} \right\rceil \tag{8}$$

$$d = char - 16i + 26\theta \tag{9}$$

where $char$ is the numerical value of the letter and i indicates the position of the letter in the email address. Now we convert the generated decimal numbers into the bit-streams and then concatenate them to obtain a combined bit-stream.

Step 2 Map the prefix of the email addresses to the 3-bits strings and concatenate them to the combined bit-stream generated in the previous step.

Step 3 Extract the values of x, y, and z from the combined bit-stream. The z value can become greater than $\hat{z}$; thus the number of categories for $\hat{z}$ is computed using the symbols of the email addresses and the value is added to the obtained z value.

Step 4 Convert the letter located after the categories' symbol into the decimal number $count\_zero$ using equations (8) and (9).

Step 5 Compute the number of replicated characters in the body denoted as x, the number of sentences denoted as y, the remaining characters denoted as z, and convert their summation into a bit-stream. After that insert $count\_zero$ zeros to the end of the bit-stream to obtain a new bit-stream.

Step 6 Repeat steps 1 to 5 for all email addresses within the received email and concatenate the obtained bit-streams to generate the final bit-stream.

Step 7 Decompress the final bit-stream generated in the previous step using the LZW algorithm to obtain the secret message. An example of the above computations in both embedding and extraction phases for a simple secret message and cover text are shown in Appendix A.

4. Experiments

The performance of steganography algorithms depends on several parameters including human-indistinguishably, robustness against statistical attacks, and hiding capacity. The most important performance parameter is the hiding capacity. In this section, we compare the hiding capacity of our proposed approach with some of the email-based steganography methods proposed in the literature.

All the experiments have been conducted on an HP PC with 3.30 GHz Intel Core i5-2500 processor with 8 GB RAM running a 64-bit Windows 7 Enterprise. The program code has been written in MATLAB R2015b. The hiding capacity is defined as the ratio of the number of bits in the secret message relative to the number of bits in the cover text. Accordingly, we formulate the hiding capacity as follows:

$$capacity = \frac{\#\ of\ bits\ in\ the\ secret\ message}{\#\ of\ bits\ in\ the\ cover\ text} \tag{10}$$

The secret message and cover text used in our experiments are chosen from [5] and illustrated in figures 4 and 5, respectively. The secret message contains 200 characters including spaces and without quotation marks (see Figure 4). The cover text as shown in Figure 5 contains 847 characters without spaces and quotation marks.

> 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

Figure 4. The secret message used in our experiments.

> In the research area of text steganography, algorithms based on front 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, but also estimate the hidden information length according to variation 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.

Figure 5. The cover text used in our experiments.

Table 2 reports the hiding capacity of various methods proposed in the literature along with our approach. As one can see from these results, our email-based steganography method is superior to other methods by providing a significantly higher capacity. This can be explained by the fact that the number of email addresses generated in our scheme is less than the number of email addresses generated in other approaches. It is to be noted that the number bits in the cover text is obtained by the sum of the number of bits in the generated email ids and the email content. Since our scheme generates fewer email ids comparing to other approaches, this clearly reduces the number of bits in the cover text, and it thus increases the capacity of our scheme.

Table 2. Comparison of hiding capacity of different methods.

Hiding Capacity	Method
7.017	[1]
6.92	[5]
7.03	[16]
7.21	[13]
10.6	**Our Method**

We aimed to generate email addresses which are very similar to the real addresses. Since we send the email to many addresses, generating a large number of invalid addresses can make our email suspicious for a curious receiver. It is to be noted that some of the generated addresses would be real and a receiver can receive the email; however, only one receiver can extract the secret message which is the target receiver. Moreover, according to the theory of Bob and Alice, existing a third receiver who receive whole of the message make no effect on the security of the email steganography.

5. Conclusions

In this paper, we proposed a novel email-based text steganography scheme, which leverages the number of characters in the cover text to hide the secret message. The proposed scheme is neither limited to the language nor any specific type of the cover text. Since we do not change the cover text, the proposed scheme is human indistinguishable. Moreover, the original receiver of the email is kept confidential because that the email is simultaneously sent to many email addresses including the original receiver. The experimental results over a popular secret message and cover text show the superiority of our scheme over several well-known methods proposed in the literature. More specifically, our scheme achieves a hiding capacity of 10.6 which is around 47% higher than the capacity of existing text steganography schemes in the literature.

We found a shortcoming in our scheme as it needs to generate many email addresses. Even that the number of addresses generated in our scheme comparing to state of the art in email steganography is promising, we plan to reduce this number in the future work. One idea for reducing the number of email addresses is to consider similar values for $\hat{x}$, $\hat{y}$, and $\hat{z}$ parameters.

References

[1] Satir E, Isik H. A Huffman compression based text steganography method. Multimedia tools and applications. 2014;70(3):2085–2110.

[2] Fridrich J. Methods for data hiding. Center for Intelligent Systems & Department of Systems Science and Industrial Engineering, SUNY Binghamton. 1997;.

[3] Mishra M, Mishra P, Adhikary M. Digital image data hiding techniques: A comparative study. arXiv preprint arXiv:14083564. 2014;.

[4] Chang CC, Kieu TD. A reversible data hiding scheme using complementary embedding strategy. Information Sciences. 2010;180(16):3045–3058.

[5] Satir E, Isik H. A compression-based text steganography method. Journal of Systems and Software. 2012;85(10):2385–2394.

[6] Morkel T, Eloff JH, Olivier MS. An overview of image steganography. In: ISSA; 2005. p. 1–11.

[7] Girdhar A, Kumar V. A Comprehensive Survey of 3D Image Steganography Techniques. IET Image Processing. 2017;p. 1–28.

[8] Saini A, Joshi K, Allawadhi S. A Review On Video Steganography Techniques. International Journal. 2017;8(3):1015–1020.

[9] Zamani M, Manaf AA, Abdullah MS. An overview on audio steganography techniques. International Journal of Digital Content Technology and its Applications(JDCTA). 2012;6(13):107–122.

[10] Rani N, Chaudhary J. Text steganography techniques: A review. International Journal of Engineering Trends and Technology (IJETT). 2013;4(7):3013–3015.

[11] Kumar R, Chand S, Singh S. An Email based high capacity text steganography scheme using combinatorial compression. In: Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference-. IEEE; 2014. p. 336–339.

[12] Kumar R, Malik A, Singh S, Chand S. A high capacity email based text steganography scheme using Huffman compression. In: Signal Processing and Integrated Networks (SPIN), 2016 3rd International Conference on. IEEE; 2016. p. 53–56.

[13] Ahmad T, Marbun MS, Studiawan H, Wibisono W, Ijtihadie RM. A Novel

Random Email-Based Steganography. International Journal of e-Education, e-Business, e-Management and e-Learning. 2014;4(2):129.

[14] Kodituwakku S, Amarasinghe U. Comparison of lossless data compression algorithms for text data. Indian journal of computer science and engineering. 2010;1(4):416–425.

[15] Kumar R, Malik A, Singh S, Kumar B, Chand S. A space based reversible high capacity text steganography scheme using font type and style. In: Computing, Communication and Automation (ICCCA), 2016 International Conference on. IEEE; 2016. p. 1090–1094.

[16] Al-Asadi SA, Bhaya W. Text Steganography in Excel Documents Using Color and Type of Fonts. Research Journal of Applied Sciences. 2016;11(10):1054–1059.

[17] Roy S, Manasmita M. A novel approach to format based text steganography. In: Proceedings of the 2011 International Conference on Communication, Computing & Security. ACM; 2011. p. 511–516.

[18] Ramakrishnan BK, Thandra PK, Srinivasula A. Text steganography: a novel character-level embedding algorithm using font attribute. Security and Communication Networks. 2016;9(18):6066–6079.

[19] Bhattacharyya S, Indu P, Sanyal G. Hiding Data in Text using ASCII Mapping Technology (AMT). International Journal of Computer Applications. 2013;70(18).

[20] Hamdan AM, Hamarsheh A. AH4S: an algorithm of text in text steganography using the structure of omega network. Security and Communication Networks. 2016;9(18):6004–6016.

[21] Shirali-Shahreza M. Text steganography by changing words spelling. In: Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on. vol. 3. IEEE; 2008. p. 1912–1913.

[22] Gardiner J. StegChat: A Synonym-Substitution Based Algorithm for Text Steganography. Master Thesis, School of Computer Science University of Birmingham, 1-64; 2012.

[23] Chang CY, Clark S. Linguistic steganography using automatically generated paraphrases. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics; 2010. p. 591–599.

[24] Nagarhalli TP. A New Approach to SMS Text Steganography using Emoticons. In: International Journal of Computer Applications (0975–8887) National Conference on Role of Engineers in Nation Building (NCRENB-14). Citeseer; 2014. .

[25] Garg M. A novel text steganography technique based on HTML documents. International Journal of Advanced Science and Technology. 2011;35:129–138.

[26] Majumder A, Changder S. A novel approach for text steganography: Generating text summary using Reflection Symmetry. Procedia Technology. 2013;10:112–120.

[27] Por LY, Wong K, Chee KO. UniSpaCh: A text-based data hiding method using Unicode space characters. Journal of Systems and Software. 2012;85(5):1075–1082.

[28] Malik A, Sikka G, Verma HK. A high capacity text steganography scheme based on LZW compression and color coding. Engineering Science and Technology, an International Journal. 2017;20(1):72–79.

6. Appendices

Appendix A. An Illustrative Example

In this appendix, we show the computations in the embedding and extraction phases of our scheme for a simple secret message and cover text, shown in figures A1 and A2, respectively. For this example, we set $\hat{x}$, $\hat{y}$, and $\hat{z}$ to 63, 3, and 127, respectively. The secret message is embedded within the three characters at the beginning and the last character at the end of the email addresses.

Shahrood University of Technology

Figure A1. The simple secret message used in our example.

School of IT & Computer engineering (ITC) was established in 2005 because of the trend of industrial highly demanded experts in ICT general field. Currently the school consists of 8 faculty members & 400 students in IT engineering and software engineering fields. ITC is a developing and growing school and is planning to establish more under graduate and graduate programs in hardware engineering, Machine intelligence and Software engineering.

Figure A2. The cover text used in our example.

The details of each step in the embedding phase of our scheme over the above example is as follows:

Step 1 The following is the bit-stream obtained by applying the LZW compression algorithm over the secret message shown in Figure A1:

1,0,0,1,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,1,0,1,1,1,0,0,0,1,1,1,0,0,0,0,0,1,1,0,
1,1,0,1,0,0,1,0,1,0,0,0,0,1,1,0,1,0,0,1,0,0,0,0,1,0,1,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,1,
0,0,1,0,0,0,1,0,0,0,0,1,0,0,1,1,0,1,1,0,0,0

Step 2 The number of characters in the cover text (message body) is denoted as *total_char* and is equal to 444.

Step 3 Starting from the least significant bit in the above bit-stream, we compute a decimal value of the bit-stream until the computed decimal value is not greater than $63 \times 444 = 27,972$. The obtained bit-stream in this step is "100100011100000001".

Step 4 The value of $\hat{x}$ can not represent the obtained value in the previous step; thus we decrement the index value, traverse backwards over the bit-stream, and set the *count_zero* value to 7 that is the number of zeros until reaching to the first 1. The obtained bit-stream is "1001000111" which is equal to the decimal value of 905. Note that we convert the bit-stream to the decimal from left to right.

Step 5 Using equations (1)-(4), we have $x = 2$, $y = 0$, and $z = 17$. For computing these values, we know that the value for traversing the whole of the text is equal to $950 - 520 = 385 < 520$. The size of the first sentence in the cover text is 86 characters,

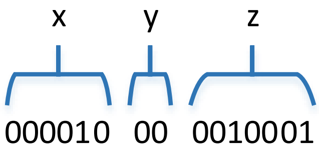


Figure A3. The bit-stream obtained from xyz.

the size of the second sentence is 268 characters, and the size of the third one is 69 characters. Now, we have $86+268+69 > 385$ and we thus obtain $y = 2$. Moreover, the number of remaining characters are $385 - 354 = 31$, and for this case, we have $z = 31$.

Step 6 The value of z is less than $\hat{z}$ and similarly $cate = 0$. Thus, there is no symbol for such value of $cate$ in Table 1.

Step 7 Now, we compute the values of x, y, and z and obtain a new bit-stream, shown in Figure A3, from the concatenation of the equivalent bit-streams of these three values.

Step 8 The "@hotmail.com" postfix is obtained using the most three significant bits in the above bit-stream and key A listed in Section 3.1.

Step 9 The remaining bit-stream is divided into three 4-bits parts: "0000", "1000", and "0010". Since the index value for the first part is $i = 0$, character a is extracted from this part using Equation (7). For the second part, we have $i = 1$ and character y is extracted accordingly. Finally, the index value for the third part is $i = 3$, and we extract character i for this part.

Step 10 From Step 4 we have $count\_zero = 7$. Thus, the last character for the email address is h obtained from Equation (7).

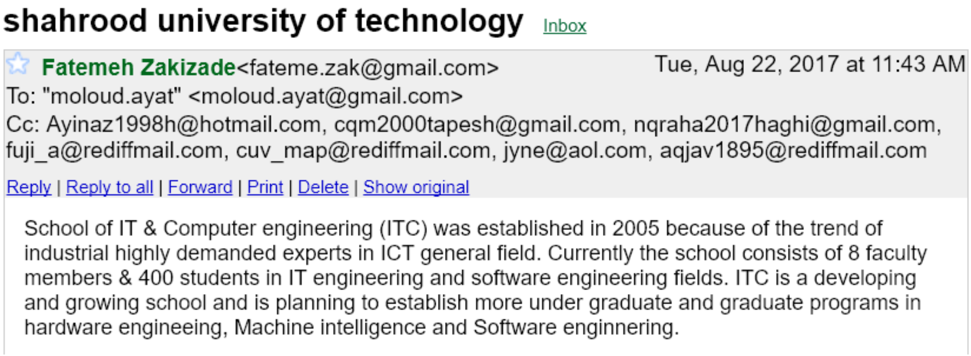
Step 11 Now we generate a meaningful email address using the above-extracted characters, the symbol and the email postfix obtained at Step 8. The generated email address is "Ayinaz1998h@hotmail.com".

Repeat Steps 3 to 11 for the remaining parts of the secret message to generate new email addresses. Figure A4 shows the whole parts of the generated email for our sample secret message and covert-text.

As an example of the extraction phase in our scheme, we extract the first part of the secret message shown in Figure A1 using the first email address and the email body shown in Figure A4.

Step 1 Since the 4-bits parts are hidden within the three first characters of the email id, we first extract the bit-streams corresponding to these parts using equations (8) and (9). Table A1 shows the results of this step.

Step 2 Using key A and postfix "@hotmail.com" in the email address, the last bit-stream "001" is extracted. This bit-stream along with the three bit-streams extracted



shahrood university of technology Inbox

Fatemeh Zakizade<fateme.zak@gmail.com> Tue, Aug 22, 2017 at 11:43 AM
To: "moloud.ayat" <moloud.ayat@gmail.com>
Cc: Ayinaz1998h@hotmail.com, cqm2000tapesh@gmail.com, nqraha2017haghi@gmail.com, fuji_a@rediffmail.com, cuv_map@rediffmail.com, jyne@aol.com, aqjav1895@rediffmail.com

Reply | Reply to all | Forward | Print | Delete | Show original

School of IT & Computer engineering (ITC) was established in 2005 because of the trend of industrial highly demanded experts in ICT general field. Currently the school consists of 8 faculty members & 400 students in IT engineering and software engineering fields. ITC is a developing and growing school and is planning to establish more under graduate and graduate programs in hardware engineeing, Machine intelligence and Software enginnering.

Figure A4. The email generated by our steganography scheme.

Table A1. Extraction of 4-bits parts from the first part of the email id.

char	i	d	4-bits part
a	0	0	0000
y	1	2	1000
i	2	3	0010

in the previous section are concatenated to generate a combined bit-stream.

Step 3 The values of x, y, and z are extracted from the combined bit-stream and consequently we have: $x = 000010$, $y = 10$, and $z = 0010001$. According to Table 1, the number of categories in $\hat{z}$ is equal to zero and thus $z = 17$.

Step 4 The value of *count_zero* is computed using equations (8) and (9) and from the last character in the email id. Thus, we have $count\_zero = 7$.

Step 5 Convert the bit-streams of x, y, and z to decimal, where they denote the number of times we traverse the cover text, the sentences at the beginning of the cover text, and the remaining number of characters, respectively. Now we sum these three values and convert the result to binary to obtain a bit-stream. Since $count\_zero = 7$, we add seven zeros to the front of the bit-stream to obtain the bit-stream "10010001110000000".

Step 6 Repeat steps 1 to 5 for other email addresses within the received email to generate corresponding bit-streams. By concatenating the bit-streams generated for each email address, we have a new bit-stream will be used in next step.

Step 7 Decompress the bit-stream obtained in the previous step using LZW compression algorithm to generate the secret message.