A Novel Approach for Hiding Information in Text Steganography

Wid Akeel Awadh
Collage of Computer Science and Information Technology
Computer Information-System Department

Abstract— In Information Security (IS), obscuring facts and communications from a 3rd party is the most crucial factor in IT (Information Technology). This can be achieved by hiding the message using a technique called steganography, which hides the presence of messages from the public. Several digital mediums, such as texts, images, audio and video can be used as cover media for digital steganography, while text steganography can also use text files as its cover. However, utilizing text as a target medium is a complex affair, due to the lack of available redundant information in a text file. However, a text file is less resource heavy and is accessible at a low bandwidth. Research is ongoing to enhance the performance metric to allow for large amounts of data to be hidden at lower memory and time consumption. This paper proposes a new text steganography approach to hide secret English text files in a cover of an English text file. To hide secret data bits, the proposed method generated a matrix of location from matching of bits of cover text file with bits of secret text file. The proposed algorithm improves data hiding capacity and time as opposed to other reported text steganography techniques.

Index Terms — Secret text file, cover text file, matrix of location, capacity, security.

1 Introduction

Sending messages to multiple parties via a public discourse that eschew the enemy from discerning its actual content is known as steganography [1].

Linguistically, steganography is defined as a form of secret writing, since the phrase "Steganography" is made up of two Greek words: steganos (secret), and graphy (writing). In practical terms, it means the art and science of obscuring secret data in innocent looking dummy container in such a way that the existence of the embedded data is imperceptible and undetectable [2].

In steganography, information is encompassed using a cover media to prevent anyone from noticing the presence of actual useful information. Certain parts to cover files will be altered for this purpose. There is a collection of files that can be used as cover files in stenography, such as executable files (i.e. exe files), HTML files, XML files, and TCP headers. It should also be pointed out that digital medias such as image, audio, text, and video files are also usable as cover files [3], [4].

Information hiding systems are made up of three aspects: capacity, security, and robustness. Capacity is the amount of information that can be hidden within a medium, security is crucial when a communication needs to be kept secret, and robustness is the amount of punishment a stego-medium can endure before the information contained within will be destroyed [5].

Text steganography is the most complex stenography: this is mostly due to the lack of a redundant information in a text file, which is not a problem in a picture of sound or image file. The structure of a text document is quite close to what is visually observed, while other media cover types (audio, picture, video) differ from what we actually observe, which makes hiding information in them a lot easier than hiding it in a text cover type. The advantage of text steganography is its requirement for smaller memory and simpler communication, enabling it to send more information and decrease printing costs [6]. It is also highly reliant on language, as each possess unique characteristics. For example, the alphabets in English is not reliant on its relative position in a word, while Persian/Arabic letters exhibit different forms based on the position of its alphabets [7].

2 Related work

The terms that will be used to discuss this work needs to be defined. The secret text file that is to be hidden will be referred to as the embedded data, while the text file used for embedding will be referred to as the cover text file. There are previous studies pertaining to hiding information in text. The following is a list of multiple reported works. Generally, text steganography methods can be classified into:

1. Change the format of the text.
2. Change the meaning of the text.

2.1 Acronym Method

Mohammad Sirali-Shahreza and M.Hassan Shirali-Shahreza from Iran proposed a method that substitutes words with acronym. A small amount of data can be hidden using this method [8].

Table1
Acronym Method

Acronym	Translation
218	Too late
ASAP	As soon as possible
C	See
CM	Call me
F2F	Face to face

2.2 Change of Spelling Method

Mohammad Shirali-Shahreza proposed a method of exploiting the way words are spelled in British and American English to hide information [9].

Table2
Change of spelling method

American Spelling	British Spelling
Favorite	Favourite
Criticize	Criticise
Fulfill	Fulfil
Center	Centre

2.3 Semantic Method

Mohammad Sirali-Shahreza and M. H. Shirali Shahreza uses a synonym to hide data and substitute target words. However, this might actually alter the meaning of the text file [10].

Table3
Semantic Method

Word	Synonym
Big	Large
Small	Little
Smart	Intelligent

2.4 White Space Method

W. Bender, D. Gruhl, N. Morimoto, and A. Lu utilize one or two spaces after each terminated sentence of the cover file/text. However, its primary disadvantage is the fact that it can only hide a small amount of data [11].

2.5 Syntactic Method

Uses syntax or text format to hide data. In this method, punctuation is inserted in the cover text for this purpose, examples being full stop (.) and comma (,) etc. at the correct places [11].

2.6 Line Shifting Method

The lines of text are vertically shifted up (for example 1/400 inch up or down), which is especially useful for printed texts [9].

2.7 Word Shifting Method

This method manipulates the horizontal distance between the words using white spaces. It should also be pointed out that this method requires more time, and the results remains discernable to the human eye [12].

2.8 Feature Coding Method

The syntax is modified to create redundancies. An example of this is to stretch or shorten letters vis-à-vis their dimensions. Attributes such as colors are also used to obscure data [13].

2.9 Method Based on Curves

This method manipulates the shape of the letters i.e. two groups based on their corresponding structures, where group A is made up of letters with curves, while group B are letters that lack curves [14].

Table4
Approach Based on Curves

Group	Group Name	Bit	Letters
A	With curves (full / partial)	0	B, C , D, G , J ,O, P , Q , R , S , U
B	Without curves	1	A, E , F , H , I , K , L , M , N, T, V, W, X ,Y , Z

2.10 Approach Based on Vertical Straight Line

This method form groups based on its respective vertical straight lines. Group A is made up of words with vertical lines, while group B are words that lack a vertical straight line [14].

Table5
Approach based on vertical straight line

Group	Group Name	Bit	Letters
A	With curves (full / partial)	0	B, C , D, G , J ,O, P , Q , R , S , U
B	Without curves	1	A, E , F , H, I,K, L, M, N,T, V , W, X ,Y , Z

2.11 Quadruple Categorization Method

This method forms four groups that are based on the presence of curves, middle horizontal straight line, single straight vertical line, or multiple straight vertical lines [15].

Table 6
Quadruple Categorization Method

Group	Group Name	Bit	Letters
A	Curves letters (full / partial)	00	C , D , G , O , Q ,S ,U
B	letters with middle Horizontal straight line	01	A, B , E , F , H , P , R
C	letters with middle	10	I, J , K , L , T, Y

3 PROPOSED METHOD

3.1 Algorithm for embedding secret text file in the cover text file.

Input: A secret text file (Ts), cover text file (Tc).
Output: A matrix of location (Lom).
Step1: Read the secret text file (Ts) and cover text file (Tc).

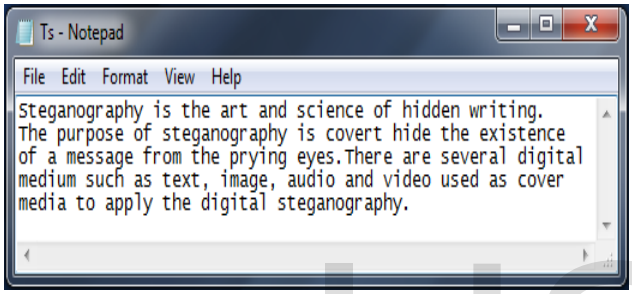

Fig. 1. Secret text file (Ts.txt)

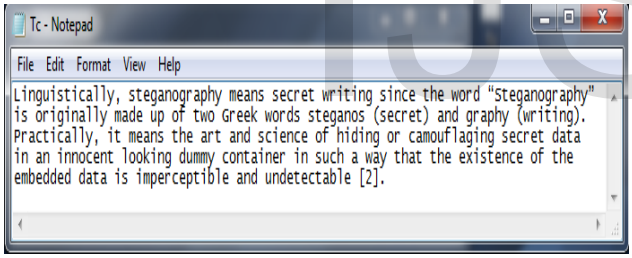

Fig. 2. Cover text file (Tc.txt)

Step2: Calculate the number of characters in the secret text file (Ts) and the cover text file (Tc).

Number of characters in secret text file (Ts) =279
Number of characters in cover text file (Tc) =318

Step3: Check if the number of characters in the cover text file (Tc) greater than the number of characters in the secret text file(Ts), if condition is true continue to step 4, otherwise, go to step11.
Step4: Conversion of the secret text file (Ts) and the cover text file (Tc) into ASCII and then into binary format.

279x1 double

	1
1	83
2	116
3	101
4	103
5	97
6	110
7	111
8	103
9	114
10	97

279x7 double

	1	2	3	4	5	6	7
1	1	0	1	0	0	1	1
2	1	1	1	0	1	0	0
3	1	1	0	0	1	0	1
4	1	1	0	0	1	1	1
5	1	1	0	0	0	0	1
6	1	1	0	1	1	1	0
7	1	1	0	1	1	1	1
8	1	1	0	0	1	1	1
9	1	1	1	0	0	1	0
10	1	1	0	0	0	0	1

Fig. 3. ASCII and Binary format of the secret text file (Ts)

318x1 double

	1
1	76
2	105
3	110
4	103
5	117
6	105
7	115
8	116
9	105
10	99

1	0	1	0	0	1	1	0
2	0	1	1	0	1	0	0
3	0	1	1	0	1	1	1
4	0	1	1	0	0	1	1
5	0	1	1	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	0	0	1
8	0	1	1	1	0	1	0
9	0	1	1	0	1	0	0
10	0	1	1	0	0	0	1

Fig. 4. ASCII and Binary format of the cover text file (Tc)

Step5: For all i=1 to 7 repeat steps 5 to 9
Step6: For j=1 to rows_of_cover_text_file
Step7: Matching the bits of the cover text file (Tc) with the bits of the secret text file (Ts) is performed.

- If bit of cover text file (Tc) =0 and bit of secret text file (Ts) =0 then save the number of zero in a matrix of locations (Lom).
- If bit of cover text file (Tc) =0 and bit of secret text file (Ts) =1 then save the number of one in a matrix of locations (Lom).
- If bit of cover text file (Tc) =1 and bit of secret text file (Ts) =0 then save the number of tow in a matrix of locations (Lom).
- If bit of cover text file (Tc) =1 and bit of secret text file (Ts) =1 then save the number of three in a matrix of locations (Lom) of dimensionality n rows and 7 column.

//n is the number of the characters in secret text file (Ts).
// resulting is the locations of matrix (Lom). Save the dimensions of matrix of location (Lom) in variable m x n.

279x7 double

	1	2	3	4	5	6	7
1	1	2	1	0	2	3	1
2	1	3	3	0	3	0	0
3	1	3	2	0	3	2	3
4	1	3	2	0	1	3	3
5	1	3	2	2	0	2	1
6	1	3	2	1	3	1	0
7	1	3	2	3	1	1	3
8	1	3	2	2	1	3	1
9	1	3	3	0	2	1	0
10	1	3	2	0	0	0	3

Fig. 5. The matrix of locations (Lom)

Step8: Increase the value of location and count variable by 1.
//Count variable is used to check whether complete data has been hidden or not.
Step9: If count variable is equal to the number of the characters in secret text file. Then message displays "Secret data file has been embedded successfully", go to step 11.

Step10: Else message displays "Text has not been embedded, Size of the cover text file is small".
Step11: end.

3.2 Algorithm for extracting the secret text file (Ts) from the cover text file (Tc).

Input: Cover text file (Tc), a matrix of location (Lom).
Output: Secret text file (Ts).
Step 1: Read the cover text file (Tc), and matrix of location (Lom).
Step 2: Conversion of cover text file (Tc) into ASCII and then into binary format.
Step 3: Calculate the length of a matrix of location (Lom).
Step4: For all i=1 to 7 repeat steps 5 to 6
Step5: For j=1 to length of a matrix of location (Lom).
Step6: Match the values of matrix of locations (L0S) and the matrix of cover text.

- If bit of cover text file (Tc) =0 and bit of matrix of locations (Lom) =0 then save the number of zero in Extract_matrix (Em).
- If bit of cover text file (Tc) =0 and bit of matrix of locations (Lom) =1 then save the number of one in Extract_matrix (Em).
- If bit of cover text file (Tc) =1 and bit of matrix of locations (Lom) =2 then save the number of zero in Extract_matrix (Em).
- If bit of cover text file (Tc) =1 and bit of matrix of locations (Lom) =3 then save the number of one in Extract_matrix (Em).

//Extract_matrix (Em) containing secret text has been created in binary format.
Step 7: Conversion of Extract_matrix (Em) from binary to ASCII format.
Step 8: Conversion of ASCII format to character format.
Step 9: Display the secret text (Ts).
Step10: End.

4 Expirmental result

Table 7
Cover text size 2640 byte and message text size 800 byte

Text steganography Approach	Message text size (byte)	Cover text size (byte)	No. of charters can hide (byte)	Time overhead (ms)
Proposed Method	800	2640	800	29,565
Method Based on Curve	800	2640	172	37,996
Method Based on Vertical Straight Line	800	2640	161	27,533
Quadruple Categorization Method	800	2640	145	26,562
Inter Word Space Method	800	2640	58	20,825
Feature Coding Method	800	2640	66	18,180
Random Character Method	800	2640	45	31,292

5 Conclusion

This paper discussed a new approach to hide English text files in a cover English text file by creating a matrix of locations. This proposed method comes with several advantages: it improves the data hiding capacity and it hides more data without creating distortions in the cover text file, meaning that the changes that can be seen are actually minimal. The security of the proposed method can be enhanced by encrypting a matrix of location. This method is seen as applicable to any language.

References

[1] K.L. Chiew, "Steganoghraphy of Binary Images", Department of computing faculty of science, Macquarie University, Australia, 2011.
[2] K. Bailey, K. Curran and J. Condell, "Evaluation of Pixel-Based Steganography and Stegodetection Methods". The Imaging Science Journal, 52, PP. 131-150, 2004.
[3] K. F. Rafat, "Survey report- state of the art in digital steganography focusin ASCII text document". International journal of computer science and information security, vol.7, no.2, 2010.
[4] A. Malik, G. Sikka and H. K. Verma,"A high capacity text steganography scheme based on LZW compression and color coding". Engineering Science and Technology, 2016, journal homepage: www.elsevier.com/ locate/jestch.
[5] G. Vennice, R. T. Swapna and K. J. Sasi, "Hiding the Text Information using Stegnography".International Journal of Engineering Research and Applications (IJERA),Vol. 2, Issue 1, pp.126-131, 2012.
[6] A. J. Memon, K. khowaja and K. Hameedullah, "Evaluation of Steganography for URDU /ARABIC", Journal of Theoretical and Applied Information Technology, 2008.
[7] M. S. Shahreza, M. H. Shahreza, "An Improved Version of Persian/Arabic Text Steganography Using "La" Word" Proceedings of IEEE 2008 6th National Conference on Telecommunication Technologies, 2008.
[8] M. S. Shahreza, M. H. SShahreza, "Text Steganography in Chat", IEEE, 1-4244-1007/07 , 2007
[9] M. S. Shahreza, "Text Steganography by Changing Words Spelling", ISBN 978-89-5519-136-3, Feb. 17-20, 2008.
[10] M. H. S. Shahreza, M. S. -Shahreza, "A new approach to persian/arabic text steganography," Proc. 5th Int. Conf. Computer and Information Science, Washington, pp.310-315, 2006.
[11] W. Bender, D. Gruhl, N. Morimoto and etal., "Technique For Data Hiding", Ibm Systems Journal, Vol.35, Issue 3&4, pp.316-336, 1996.
[12] Y. Kim, K.Moon and I.Oh, "A Text Watermarking Algorithm Based on Word Classification and Inter-Word Space Statistiocs", Prceeding of The Seventh International Conference on Document Analysis and Recognition, pp775-779, 2003.
[13] M. S. Shahreza, M. H. S. Shahreza, "Text steganography in SMS," Proc. Int. Conf. Convergence Information Technology, Washington, pp. 2260-2265, 2007.
[14] S. Dulera, D. Jinwala and A.Dasgupta, "Experimenting with The Novel Approaches in Text Steganography", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011.
[15] Shivani ,V. K. Yadav and S. Batham, "A Novel Approach of Bulk Data Hiding using Text Steganography", 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), Available online at www.sciencedirect.com.