Journal of Theoretical and Applied Information Technology
30th November 2017. Vol.95. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW TEXT STEGANOGRAPHY TECHNIQUE BASED ON A SET OF TWO-LETTER WORDS

SALWA SHAKIR BAAWI, MOHD ROSMADI MOKHTAR, ROSSILAWATI SULAIMAN
Research Center for Software Technology and Management (SOFTAM), Faculty of Information Science and Technology, National University of Malaysia, Pekan Bangi, 43600 Bangi, Selangor, Malaysia.
E-mail: salwa_sh2001@yahoo.com, mrm@ukm.edu.m, rossilawati@ukm.edu.my

ABSTRACT

Steganography is a secret writing wherein one person communicates with another without drawing suspicion to the secret communication through the medium. Text steganography is regarded the most difficult carrier to conceal secret data with because of its insufficient redundant information compared to image, audio, or video files. In this paper, we propose a new method for concealing information in English writing using non-printing characters, such as zero width non-joiner (ZWNJ) and zero width joiner (ZWJ). This approach uses to text steganography on text files. Secret information is embedded inside the English script using two-letter words based on their locations, hence achieving steganography. Results show that the technique satisfies perceptual transparency and information hiding capacity in the cover file by comparing with two previous developed existing methods. However, the size of the cover and stego document increases by approximately (22.61%) from the original size.

Keywords: *Data Security, Information Hiding, Text Steganography, Carrier File, Unicode*

1. INTRODUCTION

Information is easily available to anyone because of the development of computers and networks. Consequently, data security has become a particularly important issue [1]. Information hiding is the process of secretly embedding a secret message into a cover media. Information hiding generally involves several sub-disciplines in the information security field. Cryptography, steganography, and watermarking are the three primary methods of information hiding [2], [3]. Steganographic systems allow hiding secret messages inside another message without arousing the suspicions of others; these messages can only be understood by the intended recipient [4].

Steganography is regarded as the most common sub-discipline for hiding information. The term steganography originated from the Greek language (στεγανογραφία) was recorded in 1462 to 1516 as steganographia, which literally means "covered writing". Secure communication ensures that unauthorized users cannot obtain secret messages for their private use [5], [6].

Steganography is commonly employed in three ways, namely, through an embedding algorithm, a carrier file, and a key used for embedding and extracting hidden [7]. As shown in Figure 1 illustrates the classification of steganography.

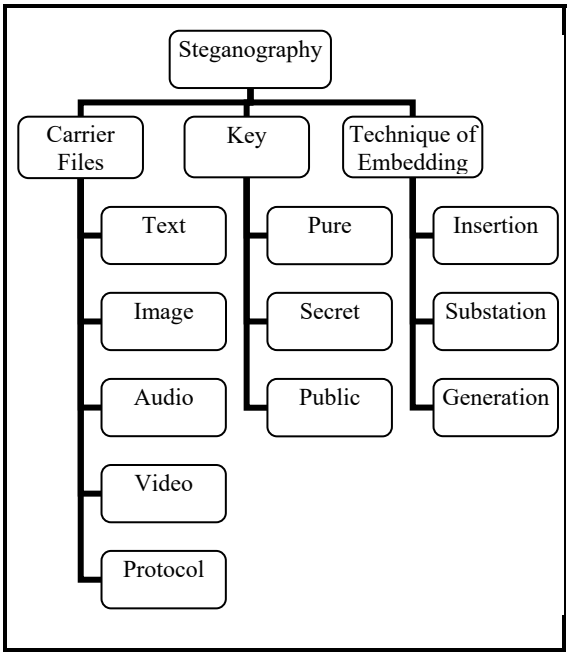

Figure 1: Classification of Steganography

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

According to [8], steganography can be classified based on the types of carrier files used. The design of steganography algorithms is influenced by carrier files, such as image, audio, video, and text files, which are frequently used over the Internet and have different characteristics that enable users to insert secret information. The most commonly used carrier files are images, which contain a high ratio of data frequency [9]. However, using the same image to hide different messages enables attackers to decode the hidden message by comparing similar images. By contrast, audio files can embed messages in .wav, .au, and even .mp3 sound files [10].

The steganography that hides secret data in text files (carrier) is called text steganography, which is one of the most challenging forms of steganography. The challenge is primarily attributed to the insufficient redundant information of texts[6], unlike image files that have substantial redundant information capacity, and in another aspect, the image process is more visceral [11]. Redundancy can help increase the capacity of the hidden data. Another drawback is that any changes to the text file with regard to size, font type, spaces between words, color, and format (e.g., from .txt to .pdf) can be easily detected.

The structure of text files corresponds to what we see, whereas other files, such as images, have different structures. The difference of each file structure allows messages to be hidden within the structure without creating a notable change in the corresponding output. Changes in audio or image files are nearly invisible, whereas the slightest addition of a letter or punctuation in a text file is noticeable to a reader. However, the minimum storage space and the ease of use for communication make text preferable than other steganography types [12], [13].

In this study, a new idea for concealing secret data in English writing is proposed. In English writing, each character shape is independent of its position in the word, unlike the Arabic/Persian characters that have different forms based on the position of the character. Therefore, we propose a new method for concealing information in English writing using non-printing characters, such as zero width non-joiner (ZWNJ) and zero width joiner (ZWJ). The method uses a set of words that contains two letters based on the Oxford dictionary. On the basis of the location of these words in the text, we randomly apply four location scenarios to hide the secret data.

The rest of this paper is organized as follows. Section 2 presents the text steganography categories. Section 3 demonstrates the proposed steganography method. Section 4 offers an example of the proposed method. Section 5 discusses the results and Section 6 discusses the conclusions and directions of future works.

2. TEXT STEGANOGRAPHY CATEGORIES

Text steganography is the art or process of hiding a text into another to secure communication. It ensures that unauthorized users cannot obtain the secret message for their private use [14]. As previously mentioned, text steganography is regarded as the most difficult technique compared with audio and image steganography because of insufficient redundant data, which is common in other carriers, rendering most of its techniques with insufficient capacity and security even though these techniques are generally transparent (secret) [12], [15]. However, the ability of cover files (such as text files) to embed secret data depends on the availability of redundant or insignificant within them. The characteristics of a cover file that is changed, manipulated, or modified during the embedding process should remain invisible to unauthorized users.

Despite the challenges of text steganography, numerous efforts have been made to design approaches in different languages, such as Arabic [16]–[18], English [19], Indian [12], [20], Farsi (Persian) [21] , Uyghur [22], Czech [23]and Telugu [24]. These efforts include three basic categories, namely, format-based, random and statistical generation, and linguistic method [13], [15], which are shown in Figure 2.

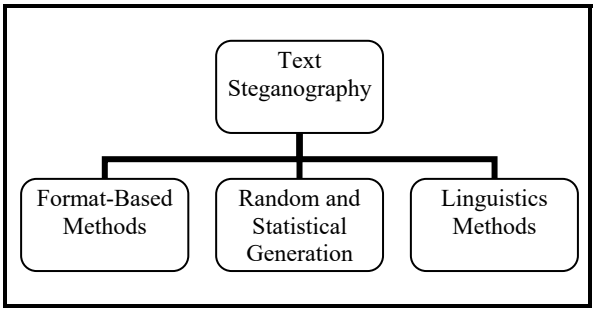

Figure 2: Categories Of Text Steganography

Physical text formatting is used to hide data through the format-based method. The term random and statistical generation refers to the statistical properties of the document, which are based on its character and word sequences. The linguistic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

method depends on the linguistic structure of the text or document to hide secret data.

2.1 Format-Based Methods

Format-based methods require physically changing the format of a text or document to hide information, which results in particular flaws. Any manipulations made can be discovered if the suspected steganographic text is compared to its original document. Even minor changes, such as different font sizes, extra white spaces, and misspellings, can be detected by a reader through a word processor [25]. Different approaches to hide information in textual files exist, as presented below.

2.1.1 Line shifting coding

This method is achieved by vertically shifting to a few degrees (for instance, each line is shifted 1/300 inch up or down), thus making it suitable for printed texts. The shifting process can represent the binary "1" or "0" The marked line assists in finding the direction of movement [26]. However, the secret information would be destroyed if character recognition (OCR) programs are used or if the text is retyped.

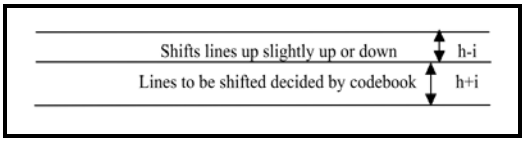

Figure 3: An Example Of Line Shifting Coding[26]

2.1.2 Word shifting coding

This method horizontally shifts words by changing the distance between them to hide information in the text. This method is acceptable for texts that have different distances between words. The shift is less likely to be identified because different distances between words in a line is common [26]. However, this method is vulnerable to attackers who are familiar with the algorithm because they can compare the existing text with the algorithm and extract the differences. This method is also ineffective because the text image can be studied to identify altered distances and word shifting is easily noticed by retyping or using OCR programs.

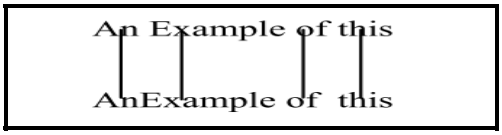

Figure 4:An Example Of Word Shifting Coding[26]

2.1.3 Open space coding

In this method, which is considered very common, white spaces are inserted into the text to hide the coded data [27]. Using white spaces lessens the possibility of identification by unintended parties because the meaning of the text document remains unchanged. White spaces can be inserted to hide secret messages via the inter-sentence space, end-of-line or end-of-file space, and inter-word space methods (see figure 5 [27]).

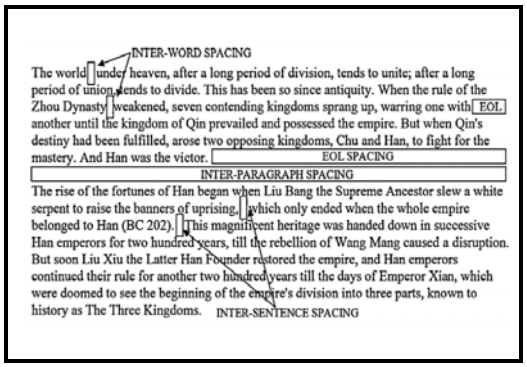

Figure 5:An Example On Open Space Coding[27]

2.1.4 Feature coding

This method modifies the features of texts in assorted ways: (1) changing certain characteristics of the text by changing the font type as proposed by [28]–[30]; or (2) changing the font size or style as introduced by [31], [32], respectively. In another study, the letter cases are changed between capitals and smalls [33]. Other suggested alternatives include changing the height of characters, replacing the points in the letters "i" and "j" [34], replacing multi-points in letters [35], and using separate letters at the beginning and at the end of the word to hide data [8].

2.2 Random and Statistical Generation Method

This method searches for a random sequence of characters. The statistical properties of word length and letter frequencies are used in the document to create words that appear to have the same statistical properties as the actual words in the given language [36].

2.3 Linguistic Methods

According to [6], [24], [37], this method relies on the language properties of the generated and modified text where the coded message is hidden in the language structure. It is the combination of syntactic and semantic methods. Syntactic methods place punctuations in specific places in a document, while semantic methods

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

replace words with its synonyms. A successful text steganography using this method requires great considerations in terms of the capacity factor of the hidden text against its cover text and its embedding and space-saving ratio factors.

3. PROPOSED STEGANOGRAPHY METHOD

Our proposed method embeds data inside a text file without inferring any modification in the file properties, such as format and content. The suggested technique depends on a set of words, which comprises two letters in the Oxford dictionary. Two non-printing characters (ZWJ and ZWNJ) are employed to embed two bits before two-letter words to enhance the embedded capacity ratio compared to other methods. These characters have no effect on the word or text and are only designed for marking the unusual cases in plain text [38]. Hence, the word format or shape in the text would remain unchanged. Furthermore, the suggested method avoids raising suspicions and any eavesdropper notifications, thus improving the robustness of the method. We mainly use ZWJ (U+200C) to mark the places used to embed data and ZWNJ (U+200D) to mark the end of the secret message in the stego-file data.

The two processes required for the proposed method, namely, the embedding and the extraction of messages, are provided in the next section.

3.1 Embedding Process

The secret message is embedded in a two-letter word in the cover file because two-letter words can have different locations in an English text document. The embedding process comprises three steps:

1) Determining the cover text file.
2) Scanning the cover file to find suitable two-letter words to hide the current two bits of the secret message.
3) Embedding the secret message depending on the location of the selected two letters according to Table 1.

Table 1: Location Of Two-Letter Word.

Embedded Bits	Location of Two-Letter Word
00	At the beginning
01	In the middle
10	At the end
11	Isolated

The cover file must have sufficient capacity to hide data. Therefore, we need to check the capacity of the cover file to hide the secret message.

3.1.1 Checking capacity

Before the embedding process, we have to make sure that the embedding capacity of the cover file is sufficient to hide the secret message. Every cover file has a capacity that depends on the techniques used. In this technique, the secret message is hidden in a set of two-letter words. Therefore, the capacity is determined by computing the number of two-letter words in the cover file. If the number of secret message characters is equal or less than the method capacity, then the embedding process is allowed. Otherwise, a new cover file is chosen. The embedding process is summarized in the following algorithm.

Embedding algorithm of a Set of Two-Letter Words (STLW) Method.
Input: Cover text file and secret message
Output: Stego-text file.
Steps:

1. Open the cover text file and segment it into words: $W = \{w_0, w_1, ..., w_{N-1}\}, 0 \le i \le N-1$.
2. Scan cover file to find two-letter words and determine the location each one.
3. Convert the secret message into binary form and then divided into blocks of two bits each.
4. Compute the number of two-letter-words to check the capacity of embedding.
5. For each block in a secret message, there are four block options (i.e., 00, 01, 10, and 11) are available to insert ZWNJ before a two-letter word that belongs to STLW, according to Table1.
   - 5.1 Block "00" calls initial word checking.
   - 5.2 Block "01" calls middle word checking.
   - 5.3 Block "10" calls end word checking.
   - 5.4 Block "11" calls are isolated word checking.
6. If it is the end of the secret message, insert the ZWJ code (200D) after the current two-letter word.
7. Return Stego-text file

Figure 6 Embedding Algorithm

Journal of Theoretical and Applied Information Technology
30th November 2017. Vol.95. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2 Extracting Process

The extraction algorithm shown in Figure 7 is the opposite of the embedding process. We must check the code of the previous character to find a two-letter word. If the previous character is ZWNJ, then the location of the current two-letter word is checked based on Table 1. If the character is ZWJ, the end of the secret message was reached. The extraction process is summarized in Figure 7.

Extraction algorithm
Input: Stego-text file.
Output: Secret message (M).
Steps:

1. Open the stego-text file and segment it into words: $W = \{w_0, w_1, ..., w_{N-1}\}$.
2. Repeat current char ! (ZWJ) // repeat until the end of the secret message is reached.
3. Select w_i, $0 \le i \le N-1$ and separate each character, then
   - 3.1 Check the Unicode for the invisible character in a current word. If the current character code is (200C), then test the location of the next two letters in the selected word to find identical secret message bits based on the scenarios in the embedding algorithm (Table 1).
   - 3.2 If the code of the invisible Unicode character is (200D), then the end of the secret message was reached.
4. Go to step 2.
5. Return to the secret message (M).

Figure 7: Extracting Algorithm

4. EXAMPLE OF THE PROPOSED METHOD

1. Read the cover file and segment it into words.
2. Scan the cover file to find two-letter words and consider the locations of the two-letter words.
3. The scanned secret message is imported and converted into binary bits and then divided into blocks of two bits each.
4. Check the capacity of the cover file to embed the secret message.
5. Read each word and divide it into blocks. Each block contains two letters (e.g., the words "am", "in", and "is"), which represent an isolated word, whereas the word "his" can be divided into two ("hi" and "is", which means that "his" does not have a middle word). Also, the word "were" can be divided into three two-letter words: an initial word "we", a middle word "er", and an end word "re".
6. Four block options (i.e., 00, 01, 10, and 11) are available.
   - 6.1 Block "00" calls initial word checking.
   - 6.2 Block "01" calls middle word checking.
   - 6.3 Block "10" calls end word checking.
   - 6.4 Block "11" calls isolate word checking
7. If it is the end of the secret message, insert the ZWJ code (200D) after the current two-letter word.
8. Return stego-text file

Figure 8 represents cover file and stego file respectively

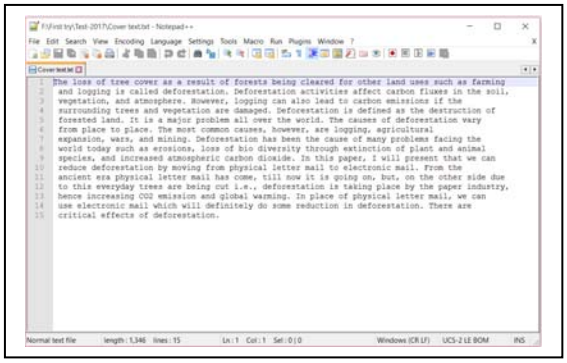

Figure 8.a. Cover File

Figure 8.b. Stego File

Figure 8: a. Cover File and b. Generated Stego File

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. RESULTS OF THE EXPERIMENT

The proposed method focuses on embedding a binary message into a text by placing non-printing character (ZWNJ) before the current two-letter word in the carrier (cover text). This method fully depends on a location of the two-letter words in the English text. In this proposed method of steganography in the English text, is tested by using various files to cover-text with varying sizes and embedding the same secret message in some of them. These resources are chosen for computing the capacity of the methods for data and including eight essay from William Shakespeare novel (eBook@Adelaide) files download from the University of Adelaide library [39] and also some cover file used in previous researches [7], [31], at the corresponding GUI for the suggested method in Figure 9.

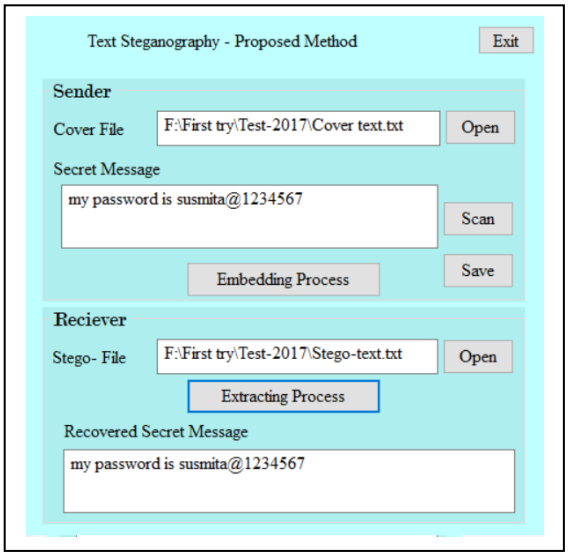

Figure 9: GUI of Proposed Method

The number bits can be hidden in carrier file, computed as follows:

$$No.\ of\ \ hidden\ bits = No.\ of\ \ two\ letter\ \ word \times 2 \quad (1)$$

Therefore, the capacity ratio of our method is

$$Capacity\ Ratio = \frac{(No.\ of\ \ hidden\ \ bits)}{Carrier\ \ file\ size} \times 100\% \quad .(2)$$

Furthermore, the size increasing ratio (SIR) in file size is computed as follows:

$$SIR = \frac{(Size\ \ of\ \ stego\ \ \ file - Size\ \ of\ \ Carrier\ \ \ file)}{Carrier\ \ \ file\ \ size} \times 100\ \% \quad (3)$$

This method does not make any apparent changes in the original text and has a perfect perceptual transparency because zero width non-joiner (200C) and zero width joiner (200D) characters are non-printing characters as shown in Figure 8. On the other side, robustness is achieved in the embedding process because the stego-file text will not change during copy and paste between computer programs, change font or size or case letters and the data hidden in texts remain intact during these operations. The capacity ratio values are presented in Table 2 for each cover file used in proposed method. Besides, Figure 10 illustrates relation between different cover file size and amount of data can be hidden inside it.

Table 1 Experimental Results (Capacity) Of Proposed Method

Experiment #	Cover File (byte)	No. of Hidden (Bits)	Capacity Ratio (%)
1	34537	21440	62.08
2	16786	10326	61.52
3	10682	6922	64.8
4	7897	4976	63.01
5	6182	3882	62.80
6	5059	3262	64.48
7	4390	2782	63.37
8	1737	1200	69.08
9 (Mahato et al. 2014)	1346	942	69.99
10 (Rafat & Sher 2013b)	1208	776	64.24
Average of Capacity Ratio			64.54

The capacity of the proposed algorithm is high compared to the similar method [31], which based on using a slight variation in the font size of invisible character space. As mentioned in [31], the method can hide (208) bits a maximum in the cover file used, Meanwhile, our proposed algorithm enabled us to hide 942 bits with applied the same cover file as shown in Table 2 Experimental (9).

In addition, the abstract from an act of William Shakespeare plays "Hamlet" as shown in Figure 10 in [7] used as a cover file. The maximum

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

number of (blank/space) character is 300 which is used to hide one bit in each space. Meanwhile, the proposed algorithm is used the same cover file enabled us to hide 1200 bits as shown in Table 2 Experimental (10).

Besides, another factor is a major factor to be considered is that the size of the text should not increase or decrease, i.e. the original file text size and the stego-file text should not have a large difference. Table 3 illustrates several attempts that are done in order to hide some characters within a particular text file which size (3476) bytes. It has shown that since the size of secret message is growing, the size of stego-file as well as ration of an extra file size to cover text size will be increased.

Table 2 Experimental Results (Size) Of Proposed Method

Experiment #	Secret Message (byte)	Stego-file (byte)	Size Increasing Ratio (%)
1	10	3556	2.30
2	20	3636	4.60
3	30	3716	6.90
4	40	3848	10.70
5	50	4008	15.30
6	60	4180	20.25
7	70	4420	27.16
8	80	4712	35.56
9	90	5074	45.97
10	100	5470	57.36
Average Size Increasing Ratio			22.61

The average size of the stego-file after embedding increased by approximately (22.61%) from the original size, which is attributed to the use of Unicode instead of the ASCII code in the embedding process and applying utf-16 encodings for each character.

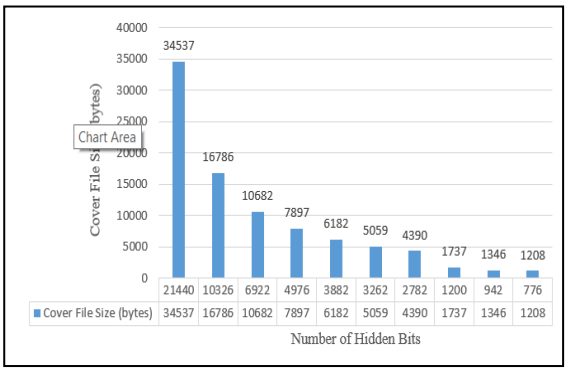

Figure 10: Relation Between Different Cover File Size and Amount of Data Can Be Hidden Inside It.

6. CONCLUSION AND FUTURE WORKS

We presented a new technique that can be used by a set of two-letter words. Our method employs remarks zero width non-joiner (200C) and zero width joiner (200D) symbols to provide data security by hiding secret bit information in the text file. The proposed method satisfies both perceptual transparency, hiding capacity and robustness. Moreover, no apparent changes in the original script are made by hiding data. However, the storage file size increases as a result of the Unicode, which uses two bytes for the representation of each non-printing characters.

Future work should focus on optimizing the highest capacity and security of the algorithm by combining the proposed method with data compression techniques, such as LZW and Huffman coding, to address the increase in the stego-text file size.

REFRENCES:

[1] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, "Comparative study of digital audio steganography techniques," *EURASIP J. Audio, Speech, Music Process.*, vol. 2012, no. 1, pp. 1–16, 2012.

[2] N. Mir and S. A. Hussain, "Secure Web-Based Communication," *Procedia Comput. Sci.*, vol. 3, pp. 556–562, 2011.

[3] A. H. Ali, M. R. Mokhtar, and L. E. George, "Enhancing The Hiding Capacity oF Audio Steganography based on Block Mapping," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 7, pp. 1441–1448, 2017.

[4] Z. Qian, H. Zhou, W. Zhang, and X. Zhang, "Robust Steganography Using Texture Synthesis," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Nov., 21-23, 2016, Kaohsiung, Taiwan*, pp. 25–33.

[5] M. Agarwal, "An Efficient Dual Text Steganographic Approach: Hiding Data in a List of Words," *Computer Networks & Communications (NetCom), Lecture Notes in Electrical Engineering*, vol. 131. Springer, New York, pp. 477–488, 2013.

[6] P. M. Vidhya and V. Paul, "A Method for Text Steganography Using Malayalam Text," *Procedia Comput. Sci.*, vol. 46, pp. 524–531, 2015.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[7] K. F. Rafat and M. Sher, "Secure Digital Steganography for ASCII Text Documents," *Arab. J. Sci. Eng.*, vol. 38, pp. 2079–2094, 2013.

[8] A. A. Mohamed, "An improved algorithm for information hiding based on features of Arabic text: A Unicode approach," *Egypt. Informatics J.*, vol. 15, pp. 79–87, 2014.

[9] S. Udhayavene, A. T. Dev, and K. Chandrasekaran, "New Data Hiding Technique in Encrypted Image: DKL Algorithm (Differing Key Length)," *Elev. Int. Multi-Conference Inf. Process. (IMCIP-2015). Procedia Comput. Sci.*, vol. 54, pp. 790–798, 2015.

[10] M. V.Kale and S. A.Patil, "Text Hiding In Multimedia By Huffman Encoding Algorithm using Steganography," *Int. J. Adv. Res. Sci. Manag. Technol.*, vol. 2, no. 1, pp. 1–5, 2016.

[11] B. Su, G. Liu, D. Xiaoyun, H. Zhang, and J. Xie, "An information hiding method for text by substituted conception," in *Proc. of IEEE ISISE*, 2012, pp. 131–135.

[12] S. Roy and P. Venkateswaran, "A Text based Steganography Technique with Indian Root," *Int. Conf. Comput. Intell. Model. Tech. Appl. 2013, Procedia Technol.*, vol. 10, pp. 167–171, 2013.

[13] M. Agarwal, "Text Steganographic Approaches: A Comparison," *Int. J. Netw. Secur. Its Appl.*, vol. 5, no. 1, pp. 91–106, 2013.

[14] P. Singh, R. Chaudhary, and A. Agarwal, "A Novel Approach of Text Steganography based on null spaces," *IOSR J. Comput. Eng.*, vol. 3, no. 4, pp. 11–17, 2012.

[15] S. Kingslin and N. Kavitha, "Evaluative Approach towards Text Steganographic Techniques," *Indian J. Sci. Technol.*, vol. 8, no. 29, pp. 1–8, 2015.

[16] M. L. Bensaad and M. B. Yagoubi, "High Capacity Diacritics-based Method For Information Hiding in Arabic Text," in *International Conference on Innovations in Information Technology*, 2011, pp. 433–436.

[17] N. A. Roslan, R. Mahmod, and N. I. Udzir, "Sharp-Edges Method in Arabic Text Steganography," *J. Theor. Appl. Inf. Technol.*, vol. 33, no. 1, pp. 32–41, 2011.

[18] N. A. Roslan, R. Mahmod, N. I. U. R. I. Udzir, and Z. A. Zurkarnain, "Primitive Structural Method for High Capacity Text Steganography," *J. Theor. Appl. Inf. Technol.*, vol. 67, no. 2, pp. 373–383, 2014.

[19] A. M. S. Rahma, W. S. Bhaya, and D. A. Alnasrawi, "Data Hiding Method for English Scripts using Ligature Characters Unicode," *Eur. J. Sci. Res.*, vol. 112, no. 4, pp. 452–459, 2013.

[20] R. Shah and Y. S. Chouhan, "Encoding of Hindi Text Using Steganography Technique," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 22–28, 2014.

[21] S. B. Ardakani, A. M. Latif, and K. Mirzaie, "Presentation a new method for Steganography in Persian text of an electronic document," *Sci. J.*, vol. 36, no. 4, pp. 700–707, 2015.

[22] M. Talip, A. Jamal, and G. Wenqiang, "A proposed steganography method to Uyghur script," in *International conference on cyber-enabled distributed computing and knowledge discover*, 2012, pp. 125–128.

[23] S. Khan, R. Sankineni, P. Balagurunathan, N. S. D. Shree, and A. Balasubramanian, "Czech Text Steganography Method by Selective Hiding Technique," in *Proceedings of the World Congress on Engineering Vol I WCE 2015, July 1 - 3, 2015, London, U.K.*, 2015, p. 4.

[24] R. S. R. Prasad and K. Alla, "A New Approach to Telugu Text Steganography," in *IEEE Symposium on Wireless Technology and Applications (ISWTA), September 25-28, 2011, Langkawi, Malaysia*, 2011, pp. 60–65.

[25] M. Khairullah, "A Novel Text Steganography System in Financial Statements," *Int. J. Database Theory Appl.*, vol. 7, no. 5, pp. 123–132, 2014.

[26] S. Roy and M. Manasmita, "A Novel Approach to Format Based Text Steganography," in *Proc. of Int. Conf. on Communication, Computing & Security (ICCCS).ACM, New York, NY, USA.*, 2011, pp. 511–516.

[27] L. Y. Por, K. Wong, and K. O. Chee, "UniSpaCh: A Text-Based Data Hiding Method Using Unicode Space Characters," *J. Syst. Softw.*, vol. 85, pp. 1075–1082, 2012.

[28] W. S. Bhaya, "Text hiding in mobile phone simple message service using fonts," *J. Comput. Sci.*, vol. 7, no. 11, pp. 1626–1628, 2011.

[29] W. Bhaya, A. M. Rahma, and D. AL-Nasrawi, "Text Steganography Based on Font Type in MS-Word Documents," *J. Comput. Sci.*, vol.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

9, no. 7, pp. 898–904, 2013.

[30] R. Bhuvaneshwari, P. Archana, and S. Mahalakshmi, "Secure Transmission Of Sensitive Data Using System Fonts Technique," *Int. J. Res. Eng. Adv. Technol.*, vol. 1, no. 1, pp. 1–5, 2013.

[31] S. Mahato, D. K. Yadav, and D. A. Khan, "A novel approach to text steganography using font size of invisible space characters in microsoft word document," in *In Intelligent Computing, Networking, and Informatics*, Springer, New Delhi, 2014, pp. 1047–1054.

[32] S. Samanta, S. Dutta, and G. Sanyal, "A Novel Approach of Text Steganography using Nonlinear Character Positions (NCP)," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 1, pp. 55–60, 2013.

[33] A. A. Ali and A. – H. S. Saad, "New Text Steganography Technique by using Mixed-Case Font," *Online J. Comput. Sci. Inf. Tehcnology*, vol. 3, no. 2, pp. 138–141, 2013.

[34] S. Khan, B. Abhijitha, R. Sankineni, and B. Sunil, "Polish text steganography method using letter points and extension," in *Proceedings of IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT*, 2015.

[35] A. Odeh, A. Alzubi, Q. B. Hani, and K. Elleithy, "Steganography by multipoint Arabic letters," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2012, pp. 1–7.

[36] D. Zhang and H. Zhong, "A Text Hiding Method Using Multiple-Base Notational System with High Embedding Capacity," in *Image and Signal Processing (CISP), 2014 7th International Congress on*, 2014, pp. 622–627.

[37] Y. Shu, L. Liu, W. Tian, and X. Miao, "Algorithm for information hiding in optional multi-text," *Procedia Eng.*, vol. 15, pp. 3936–3941, 2011.

[38] Julie D. Allen, D. Anderson, J. Becker, and et.al., Eds., *The Unicode Standard*, Version 8. 2015.

[39] "The University of Adelaide Library, eBooks@Adelaide. Available from: http://ebooks.adelaide.edu.au/," *The University of Adelaide Library, eBooks@Adelaide. Available from: http://ebooks.adelaide.edu.au/*. .