Improvement of "Text Steganography Based on Unicode of Characters in Multilingual" by Custom Font with Special Properties

Salwa Shakir Baawi¹, Dhamyaa A. Nasrawi*², Lina Talib abdulameer²
¹ University of Al-Qadisiyah- College of Computer Science and Information Technology, Department of Multimedia.
²Computer Science Department, College of Science, University of Kerbala, Iraq

E-mail: dh.alnasrawy@uokerbala.edu.iq

Abstract. Since computer utilization is expanding, for both social and trade ranges, secure communications through channels got to be an exceptionally critical issue. Information hiding away could be a strategy to get a secure communication medium and securing the data amid transmission. Text documents have very less redundant information as compared to the images and audio, therefore, text steganography is most challenging. This paper aims to improve "text steganography based on Unicode of characters in multilingual" by design new font with special properties for purposes of hiding data. Furthermore, this method based on making the same glyphs for the multiple codes, the Set of High-Frequency Letters called SHFL in the English language was chosen for the embedding process. The hiding method replaces the code of English symbol with other code that has the same glyph exactly. Two bits are hidden at once, utilizing glyph1 for hiding 00 and utilizing glyph2, glyph3, and glyph4 for hiding 01, 10, and 11. The improvement increases the steganography capacity, transparency and improves the security and robustness of the text stego file.

Keywords: Capacity, Font design, Steganography, Text hiding, Unicode standard.

1. Introduction
Information security has found particular importance nowadays, because the new development of communication and computers, information is became easily available for anyone over the world. The require for security has ended up exceptionally fundamental. Data hiding away may be a common condition including different sub-disciplines within the data security field. Cryptography, steganography and watermarking are three main methods of information hiding[1].

Steganography is the art of hiding communication; a steganographic system thus hides a secret message within another message without getting any doubt of others so that its educated beneficiary can
as it identified the interior data. Though steganography respects the foremost well-known sub-disciplines of information hiding [2,3].

The steganography process involve of Carrier (cover medium), Secret data, stego-key, and stego medium. In the cover medium embeds the secret information to form the stego medium.

Stego medium = carrier + secret data + stego-key

In this context, the carrier (cover medium) is the file in which we will embed the embedded data, which may also be encrypted using the stego_key. The resultant file is the stego_medium (which will, of course, be the same type of file as the cover medium). Steganography not to change the structure of the secret message just embeds it inside a carrier.

Steganography can be arranged in many ways. Such as, according to carrier file type can be classified as text, image, audio, movie, video, or protocol file used to embed secret data [4,5,6]. According to the kind of the key that used to hide information: pure Steganography, secret key steganography, and public-key steganography. Also, based on embedding method, there are three techniques used to protect information in a cover object: insertion-based, substitution-based, and generation-based techniques [7,5].

Text Steganography is believed to be the complicated due to the lack of redundant data in a text file as compared with an image or an audio file [8].

Capacity, transparency, security and robustness are the four key aspects affecting steganography and its usefulness. The four important aspects are not independent, but should instead be seen as competing destinations, which can be balanced when designing a scheme. The capacity denotes the amount of data that can be hidden in the carrier file. Transparency refers to the measure of distortion in the carrier. Security related to the ability of an eavesdropper to figure the hidden information easily. Whereas, the robustness is the measure of the capability of an algorithm to retain the data embedded in the host file.

The aims of this paper are improve of text steganography based on Unicode of characters in multilingual by design new font with special properties for purposes hiding data. Furthermore, this method based on making same glyph for the multiple codes. The improvement increases the steganographic capacity, transparency and improve the security and robustness of the text stego file which will be explained in details. The contributions of this paper are:

1. Design new font with special properties for hiding purposes.
2. Increase capacity and transparency of hidden data by making same glyph for the multiple codes.
3. Design an algorithm to hide data in English document text based on custom font
4. Evaluate the resistance of the proposed method against stegonalysis attacks(visual, structural, and statistical attacks).

The organization of the paper as follows: literature survey in section 2, proposed method is presented in section 3,While section 4 presents results and discussion. Section 5 demonstrates the conclusion and the future work.

2. Related works
In this paper, we shed light on the various methods used in text steganography were published between the year 2013 to 2019. A steganography system should fulfil diverse necessities such as transparency, capacity, security, and robustness. However, Many methods are available in the following section for expanding payload capacity, enhancing security, maintaining transparency and robustness.

As an instance, Bhaya et al. [9] have proposed a new technique called it (SEFT). This technique is objective to achieve the security of hidden messages in stego-text base on the similarities of English fonts type. The hiding method implemented by altering the font type to a new. The secret message was encoded and hidden as similar fonts in capital letters of the cover media. The technique provides very high capacity(number of Capital Letters in the cover document). Besides, the perceptual transparency based on the similarity of font types which it has good. However, the increase in size ratio was about 0.766%.
Ivan Stojanov [10], proposed a method for hiding data called as Property coding belongs to format based methods of text steganography in a Ms-Word files. It is that uses some text formatting that are invisible to the human eye. Property coding exploits the properties of the different document object like characters scale or character underline, Paragraphs border and senfources border for embedding data. The approach satisfies higher capacity and transparency. However, it has low robustness, and also it introduces slight overhead on file size about (1%) according to experimental results.

Besides, For having the better visual quality of stego-text, it takes care of the point at the end of the senfources area for embedding data. Ardakani, Latif, and Mirzaie [11] used one bit to hide secret data in each dot by changing the font. The main stages are point recognize and modify its font according to secure information. So in this way, the method offers transparency 100 percent, and it is not increasing in the volume of stego-text. Moreover, in all languages can be used and it can have resistant against of attacks such as resize text, change of text color, font style, insert a word or phrase in the text. The drawback of this method, it has not resistant against attacks by changing the font, copying, and cutting and it has a small amount capacity.

Muhammad Aman et. al. [12] proposed a new hybrid method combining Unispach [13] and Zero-Width Characters [14] approaches based on format-based open-spaces. The proposed method outdo the drawbacks of existing methods, such as high hidden capacity by using lose-less compression algorithm and hiding 4 bits per space by any version of MS Word file as a stego carrier. As well as, improved the robustness by multi-layers of security and SHA-1 algorithm. They ensure in their novel method that it is the best suitable for large messages.

In this research, we attempted to improve the text steganography method that based on Unicode of characters in Multilingual [15]. This method suffer from increasing of stego-file because the embedding of Unicode characters which have 2-bytes, also there are some conditions to replacement the original glyphs with new one.

3. Methodology
The Unicode was used in many application, such as ancient text [16,17], font design [18], cryptography [19,20], and in information hiding [12,13,14,15,21].

The proposed method depends on [15] which is a new English text steganography method presented using the Unicode of multilingual characters. This method depends on a good appearance of specific characters in other languages, with different codes and different glyphs. Because of the glyphs of them dissimilar to original English scripts, just 13 characters were chosen for hiding process based on the following criteria: (Located in Plane 0, similar to the original text, and supported by standard fonts).

In this paper, new improvements were made to the mechanism of proposed text steganography scheme explained in Fig.1 as shown below. Firstly, a secret message converts to binary bit, after that it will be covered up by two carriers cover text and font file by applying the embedding algorithm to produce a stego-text and font file. The stego-text and font file will be transmitted over an insecure communication channel to the receiver. In other side, the extracting algorithm implements by open stego text file and convert the binary bits to get secret message. The methodology phases of the proposed work explained in the followed subsections.
Figure 1. The proposed text steganography

Based on design font (by font creator program) with special properties the hiding purposes overcomes some drawbacks in proposed method of [15] which will be listed and illustrated in details by the subsections below.

Now, how the font file was produced with special properties? what is the hiding algorithm? and What is the extraction algorithm? All answers of these questions will be explained in the following subsections:

3.1. Font file production with special properties
The text displayed on the screen by Fonts. Font production needs to understand the process in that a series of characters typed are transformed into a series of glyphs displayed on the screen. One important issue is that for each character there is own code and glyph.

Many research was explained the steps of font file production[17,18] which are summarized in the following:

Step1: Determine the glyphs of Basic Latin symbols. The term "glyph" can be defined as an element of writing. The glyph of letters can be created using (Font Creator) program, This step (design of font) involves all glyphs of Latin symbols. The Latin symbols located in Plane0 started from U0000-U00FF. There are two blocks of controls and Latin Letters but the first block was chosen (which take one byte only) in hiding process: C0 Controls and Basic Latin, this block contains all symbols which is exactly the ASCII code. To visually read off character codes of characters from the table, first identify the row index (hexadecimal number) in which the character occurs, and then append the column index as the least significant digit to the number. Example: The code for the character ’a’ (LATIN SMALL LETTER A) is 0x61, or Unicode U+0061. See Fig. 2.

Step2: The English letter frequency is the main feature that used for hiding amounts of secret bits inside the cover text, thereby increasing the payload capacity of the stego-text radically. So the Set of High-Frequency Letters called SHFL in the English language was determined to use it in hiding process. Assumed four high-frequency characters in the English language as illustrated in Fig. 3.
Step3: Include four similar glyphs for each SHFL with different codes in custom font. The properties of the custom font are summarized as follows:

- Involves all glyphs of Latin symbols that started from U0000-U00FF but replacing some symbols in the C0 Controls and Basic Latin Block by the letters of SHFL, in another word, change the glyphs of some symbols in the block to glyphs of SHFL, see Fig. 4 and Table 1.
- Each SHFL in custom font has one byte only.

Step4: Test and install the custom font.

Figure 2. The C0 Controls and Basic Latin Block

Figure 3. The English Letters Frequency

Figure 4. The C0 Controls and Basic Latin Block after replacement in new font
As a result, the designed font has four similar glyphs of each letter in the SHFL. The hiding method replaces the code of English symbol with other code that has the same glyph exactly. Two bits are embedded at one time, using glyph1 (ASCII code in range 0x61-0x7A ) for embedding 00 and using glyph2, glyph3, and glyph4 for embedding 01, 10, and 11.

### 3.2. Hiding Algorithm

Hiding (Embedding) algorithm based on the frequency feature of English letters called the Set of High-Frequency Letters (SHFL) which contains four high-frequency letters, SHFL= \{e, t, a, o\}. As the proposed method in [15], this method work by scan cover document to find appearance of SHFL, then check up the capacity of hiding (enough or not). The secret message converted to binary, then can be hidden two bits at one time.

### Table 1. Selected letters in SHFL for hiding process

| Letters | ASCII code | Unicode  |
|---------|------------|----------|
| S=00    | 0065       | 0023     |
| S=01    | 0074       | 003C     |
| S=10    | 0061       | 005B     |
| S=11    | 006F       | 007B     |
| e       | 0065       | 0023     |
| t       | 0074       | 003C     |
| a       | 0061       | 005B     |
| o       | 006F       | 007B     |

As a result, the designed font has four similar glyphs of each letter in the SHFL. The hiding method replaces the code of English symbol with other code that has the same glyph exactly. Two bits are embedded at one time, using glyph1 (ASCII code in range 0x61-0x7A ) for embedding 00 and using glyph2, glyph3, and glyph4 for embedding 01, 10, and 11.

### Example

Let the secret message S = "happy", while the text file T explained in Fig. 5. The bold letters in red color represent the four high-frequency letters, SHFL that embedding message in them.

Communicators and marketers can now adopt a personalized approach to their work, ideally one based on behavioral science. But the execution lags behind the science while the claims of some marketers as to what personality marketing can do far exceed it. Moreover, public controversies like the Facebook and Cambridge Analytica story threaten personality marketing’s potential before it has really matured.

**Figure 5.** Text file T or cover document
For each character, get the binary form of the hexadecimal code, see Table 2:

| Secret Message S | h a p p y |
|------------------|---------|
| Hexadecimal      | 0068 0061 0070 0070 0079 |
| Binary           | 01 10 10 00 01 10 00 01 01 11 00 00 01 11 00 00 01 11 10 01 |

Now, scan cover document T to find appearance of SHFL, note that the first appearance is the letter: o, then, for each two bits of S check up if they 00, 01, 10, or 11 based on Table 1. See the first six symbols from SHFL in the cover document T, see Table 3.

| Cover document T | o a t o a a .... |
|------------------|----------------|
| Code before hiding | 006F 0061 0074 006F 0061 0061 .... |
| S in Binary      | 01 10 10 00 01 10 |
| Code after hiding| 007B 005D 003D 006F 005B 005D .... |

3.3. Extracting Algorithm

The extracting algorithm implements by open stego text file and convert the binary bits to get secret message. Firstly, scan stego file SF to find SHFL characters, then check up the code of current character if it in C0 Controls and Basic Latin range(U+0000- U+007F) then secret message S = 00, else the S= 01, 10, or 11 based on Table 1. Note that the first 2-bytes denoted to length of S. Extracting steps explained in the following algorithm:

```
Extracting algorithm
Input: Stego file SF
Output: Secret message S.
1. Open stego file(SF).
2. Extract the length of S in first 2-bytes.
3. Scan SF file to find SHFL characters,
4. Check up the code of current character:
   - if code in range(0x61-0x7A) then S=00
   - else S=01, or 10, or 11 based on Table 1.
5. Return secret message S
```

To extract the secret message in previous example, see Table 4:

| Stego file SF | o a t o a a .... |
|---------------|----------------|
| Code of SHFL in SF | 007B 005D 003D 006F 005B 005D .... |
| Check up in range(0x61-0x7A ) | No No No Yes No No .... |
| S in binary | 01 10 10 00 01 10 ... |
4. Experimental results and discussion

The improvements were made in the proposed method hiding data in text files and evaluated by determining some important parameters.

All experiments were implemented to evaluate the improved method and original method in [15]. We used *(The Lost Girl, by D. H. Lawrence)* with the first chapter for evaluation. The files are downloaded from The University of Adelaide Library [22].

Many important parameters were considered in this paper such as:

- **Capacity**, includes an amount of data that can be hidden in the cover object. Table 2 explains the capacities of improved method and original.

- **Size Increasing Ratio (SIR)** in file size is computed as follows:
  \[ \text{SIR} = \frac{(\text{stego doc. size} – \text{cover doc. size})}{\text{size of cover doc.}} \times 100\% \quad … (1) \]

- **Invisibility**: is depends on the properties of the human visual system. The if human is unable to distinguish between carriers and stego file, then the embedded information is imperceptible.

The results of embedding different size of secret messages are illustrated in Table 5.

### Table 5. Results of embedding different size of secret messages

| Methods   | Size of Messages (Bytes) | # of selected Letters in Cover | Capacity Max.# of bits | Size of Cover (Bytes) | Size of Stego (Bytes) | SIR % |
|-----------|--------------------------|-------------------------------|------------------------|-----------------------|-----------------------|-------|
| Improved  | 235                      | 13087                         | 26174                  | 46872                 | 46872                 | 0     |
|           | 396                      |                               |                       |                       |                       |       |
|           | 658                      |                               |                       |                       |                       |       |
| Original  | 235                      | 3376                          | 6752                   | 46872                 | 623,748               | 30.9  |
|           | 396                      |                               |                       |                       |                       |       |
|           | 658                      |                               |                       |                       | 715,774               | 50    |

Figures 6 (a and b) show the cover document and stego document.

**Figure 6**: a: Cover document for improved method, b: Stego document for improved method

Now, we will discuss the following parameters for improved and original methods based on the above experiments:

1. **Capacity**:
   - Improved method: depends on most frequency characters SHFL in lower case, so the capacity of this method is very high as well as in each character can be hide 2 bits. See Table 5.
   - Original (Multilingual): method depends on on selected characters frequency, also the capacity is very high but it is low compared to improved method, because most selected characters in
2.Size Increasing Ratio (SIR):
- Improved method: no any increasing ratio because the custom font replace characters in one byte with the another in the same size, so the cover document and stego document in the same size. See Figure 4, Table 5.
- Original (Multilingual) method: there is increasing ratio about 11.1 in stego document[15] because the replacement of characters in one byte with the another in 2 bytes.

3. Invisibility:
- Improved method: this method has a good perceptual transparency based on custom font which designed to make multi glyphs with different codes to each letter in SHFL. So the cover document as the same as stego document.
- Original (Multilingual) method: also has a good perceptual transparency because the glyphs of selected characters are more similar and suitable to glyphs of original script. Therefore the stego text is similar to the cover text.

5. Conclusion and future work
This paper introduced some improvements on txt steganography method that based on Unicode of characters in Multilingual[15]. The improvements based on custom font with special properties that focused on the main goals of steganography (capacity, size increasing ratio, perceptual transparency). These improvements are summarized below:

- Improvement in capacity: the new method depends on most frequency characters SHFL in lower case, for each letter, two bits can be hidden which outperform the original method in terms of capacity.
- Improvements in SIR: Improved method: because the custom font replace characters in one byte with the another in the same size (one byte with one byte), that mean making same glyph for the multiple codes so the cover document and stego document is similar in size.
- Improvements in perceptual transparency: because the custom font has same glyph for the multiple codes, then the glyphs in cover and stego documents are the same.
- Because the stego document will not lost during digital copy-past operation, which means that preserve hidden information. Then, the information hidden in script remains intact during these operations.

Acknowledgement
I cannot express enough thanks to my department (Computer Science) in college of Science at University of Kerbala for their continued support and encouragement. I offer my sincere appreciation for the learning opportunities provided by my department.

My completion of this project could not have been accomplished without the support of my colleagues.

References
[1] Saniei, R., & Faez, K. (2013). "The capacity of arithmetic compression based text steganography method". Paper presented at the Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on.
[2] Agarwal, M. (2013). "Text Steganographic Approaches: A Comparison". International Journal of Network Security & Its Applications (IJNSA), 5(1), pp. 91-106.
[3] Vidhya, P. M., & Paul, V. (2015). "A Method for Text Steganography Using Malayalam Text.", Procedia Computer Science, 46, pp. 524-531. doi: 10.1016/j.procs.2015.02.079

[4] Agarwal, M. (2013). "Text steganographic approaches: a comparison". arXiv preprint arXiv:1302.2718.

[5] Ammar Odeh, Khaled Elleithy, Miad Faezipour, a., & Abdelfattah, E. (2015). "Highly Efficient Novel Text Steganography Algorithms".

[6] Subhojit Malik1, W. M. (2015). "Hiding Information- A Survey", Journal of Information Sciences and Computing Technologies(JISCT), 3(3).

[7] Abbas, T. A., & Hamza, H. K. (2014). Steganography Using Fractal Images Technique. IOSR Journal of Engineering, 4, 52-61.

[8] Samanta, S., Dutta, S., & Sanyal, G. (2014). "A Novel Approach of Text Steganography using Nonlinear Character Positions (NCP) ". International Journal of Computer Network and Information Security (IJCNIS), 6(1), 55.

[9] Bhaya, W., Rahma, A. M., & Dhamyaa, A.-N. (2013). "Text steganography based on font type in ms-word documents". Journal of Computer Science, 9(7), 898.

[10] Ivan Stojanov, A. M., Igor Stojanovi'. (2014). "A New Property Coding in Text Steganography of MicrosoftWord Document's. Paper presented at the SECURWARE 2014 : The Eighth International Conference on Emerging Security Information, Systems and Technologies.

[11] Ardkhani, S. B., Latif, A. M., & Mirzaie, K. (2015). "Presentation a new method for Steganography in Persian text of an electronic document". Cumhuriyet Science Journal, 36(4), 700-707.

[12] Aman, M.; Khan, A.; Ahmad, B.; Kouser, S.(2017), "A hybrid text steganography approach utilizing Unicode space characters and zero-width character". Int. J. Inf. Technol. Secur., 9, 85–100

[13] L. Y. Por, K. Wong, and K. O. Chee,(2012), "UniSpaCh: A text-based data hiding method using Unicode space characters", Journal of Systems and Software, vol. 85, pp. 1075-1082.

[14] A. Odeh and K. M. Elleithy,(2012), "Steganography in Text by Merge ZWC and Space Character".

[15] Rahma, A.M.S.; Bhaya, W.S.; Al-Nasrawi,D.A.(2013), "Text steganography based on Unicode of characters in multilingual". Int. J. Eng. Res. Appl. (IJERA) 2013, 3, 1153–1165.

[16] Dhamyaa A. AL-Nasrawi et al,(2014), "Unicode Text Editor for Ancient Egyptian Hieroglyphs Writing System", Egyptian Computer Science Journal , Vol. 38, No.2May. 48-55.

[17] Dhamyaa A. AL-Nasrawi et al, (2016), "Characters Map for Cuneiform Writing System". Journal of Kerbala University, Vol. 7, No.1 , 141-152.

[18] Dhamyaa A. AL-Nasrawi et al, (2016),"Digital Handwriting Imitation of the Fourth Caliph (Imam Ali Bin Abi Taleb)", International Journal of Digital Content Technology and its Applications(JDCTA), Vol. 10, No.3., 42-59.

[19] Dhamyaa A. AL-Nasrawi et al, (2018),"Cryptographic system based on Unicode". Journal of Physics, Conf. Series 1032 012049.

[20] Inas R. Shareef et al., (2019),"New Cryptographic System of Romanized Arabic Text Based on Modified Playfiar". Journal of Engineering and Applied Sciences, Vol. 14, Issue:4: 1331-1338.

[21] Rahma, A.M.S.; Bhaya, W.S.; Al-Nasrawi,D.A.(2013), "Data Hiding Method for English Scripts Using Ligature Characters Unicode". European Journal of Scientific Research, Vol.112, No.4, 452-459.

[22] The University of Adelaide Library, eBooks@Adelaide. Available from: https://ebooks.adelaide.edu.au/l/lawrence/dh/l41lo/chapter1.html