# Steganography in Text by Merge ZWC and Space Character

Ammar Odeh
Computer Science & Engineering,
University of Bridgeport,
Bridgeport, CT06604, USA
Aodeh@bridgeport.edu

Khaled Elleithy
Computer Science & Engineering,
University of Bridgeport,
Bridgeport, CT06604, USA
elleithy@bridgeport.edu

**Abstract**

Secure communication is essential for data confidentiality and integrity especially with the massive growth of the internet and mobile communication. Steganography is an art for data hiding by embedding the data to different objects such as text, images, audio and video objects. In this paper we propose a new algorithm for data hiding using Text Steganography in Arabic language. Our algorithm uses the Zero Width Character from Unicode (U+200B) and space character to pass bits before and after space. Main advantage of our algorithm file format will not be change and this will decrease the ability of Stegoanalysis to observe hidden data. Moreover ZWC algorithm can be applied to any language (ASCII, Unicode).

Keywords: Carrier file, Zero width chracter, Information Hiding, Diacritics.

## 1. INTRODUCTION

### 1.1. Background

The word Steganography is constructed from two Greek words. First word is "Stegano" which means hidden and second word is "Graptops" which means writing. In Steganography the secret data is hidden in different objects, so attackers will find difficulty to recognize it and obtain it [1]. One of the Steganography examples is the invisible ink, where the readable message can't be read without using a proper way to read it. An intruder intercepting the message will not be able to read it. However, the authorized person will be able to read that message after identifying the substances features used in writing the message [2][3].

There is a method an ancient Greece used before which was shaving the messenger head and wait until it grows again, after that the message can be send to the destination[1]. Performing this method gives two possibilities. First, on the message arrival the receiver can read the message and determine if the message has changed or not. Second, message not arrival will mean that the attacker intercepted the message.

### 1.2. Motivation

In Steganography, there are three techniques the algorithms depends on to hide the data in the files.

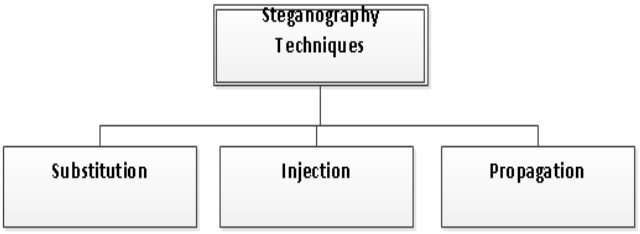


Figure 1. Steganography techniques

The first one is Substitution, which is a technique to substitute or exchange small parts of the carrier file with secret date. The idea is that if there is any attacker in the middle of the channel will find it hard to observe the changes in the carrier file. However, it is very important to carefully choose replacement process to avoid the carrier file to be suspicious. This could be done by changing insignificant part of the carrier file. For example, considering the carrier file to be and image (RGB) then the least signification bit (LSB) will be used as the exchange bit [4].

In Injection technique, hidden data will be added to the carrier file, this will result increasing the carrier file size and also increasing the probability of the being discovered. The goal from this technique is how to add hidden data in the carrier file and make it not suspicious to the attacker [4].
Third technique called Propagation, This technique does not depend on an object for cover instead, it depends on generation engine. The data is fed into the generation engine and then create a mimic file which could be a graphic, audio or text.

The main components of Steganography are cover media, hidden data and stego-medium as in Figure 1.

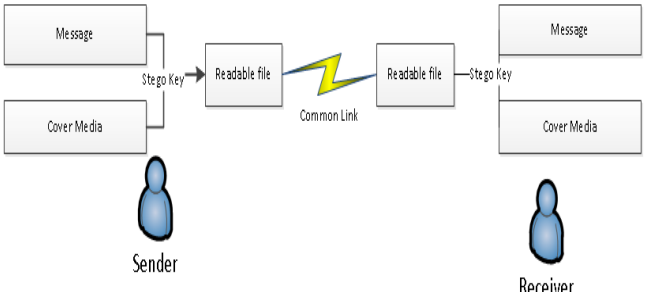


Figure 2. General components of Steganography

In Steganography there are different covering media can be used such as images, audio, video and text. It is very important to carefully choosing the carrier file type to protect the embedded message. For successfully implementing Steganography, the carrier files should not be suspicious. Attacking on Steganography is done by Steganalysis which is done by analyzing the transmitted files to determine if it has any indication that it has hidden data which is defeat the principles of the Steganography [3][4].

The most difficult type of Steganography is Text Steganography because of the data redundancy with the comparison with the other carrier files [5], and that reduce the capacity of the hidden data. However, Text Steganography has some dependency on the language that being used, the language characteristics are different among the other languages for instance, considering the letters shapes in the English language and in Persian/Arabic, we find that in English letters shapes doesn't depend on their position in the word where in Persian/Arabic depends on the letters position in the word and also have different forms in different positions in the word [6].

In this research we aim to propose an algorithm to hide text inside text using Arabic language. Random algorithm will be employ for hidden bits distribution inside the message. Choosing the Arabic language was because of four reasons. First, multi dotted pointers letters were used in the algorithm we propose. Therefore, we need to employ it on a language that offer as much as possible of the dotted latters. For instance, in Arabic language there are 5 multi dotted points letters and there are 7 multi dotted points letter [7], where the English language has doesn't have such latters. Second, because the availability of the Arabic electronic textual information. Third, the research of Text Steganography in other languages is less compared to English language. Fourth, Arabic could be extended to other languages that use similar letters such as Urdu, Farsi and Kurdish.

### 1.3. Main Contributions and Paper Organization

In this paper we propose an algorithm for Steganography using Arabic Text. The main goal is to use Kashida, which is a character in Arabic with zero width that allows us use two bits for each letter. However, the previous algorithm concentrated only on hiding one bit for each letter. Parallel connection and randomization will be used to avoid the hidden data to be suspicious.

The organization of the paper is as the following. Discussing some of Text Steganography is section II. In section III we present and discuss Kashida and Zero width algorithms for data hiding. In section VI the conclusion and the remarks.

## 2. PRIOR WORK

There are two main categories for Text Steganography. First semantic based. Second, formatting based as in Table I. We present some of the example about these two categories in the section. Table I has simple comparison between semantic and format methods.

**Table I. Comparing between texts Steganography**

| | Semantic Method | Format Method |
|---|---|---|
| Amount of hidden data | Small amount | More than semantic |
| Flaws | Sentence meaning | notice from OCR or retyping |

Evaluation of Steganography is based on the size of the data that could be hidden and the challenges in these methods. We make a comparison in this section between ten algorithms that are used to hide data in side text documents. The last two of these algorithms are employed on Arabic and Persian languages.

Word Synonym [7][10][11], classified as one of the semantic method. This method focuses on replacing some of the words by their synonym. In this technique the hidden data will be transmitted without being suspicions to the attackers. However, in this method the data is considered small comparing to the other methods but it could change the sentence meaning.

**Table II. Using Word Synonym**

| Word | Synonym |
|---|---|
| Big | Large |
| Find | Observe |
| Familiar | Popular |
| Dissertation | Thesis |
| Chilly | Cool |

In [9] present Punctuation method uses punctuation such as (.) and also (;) to form hidden data. For instance, "NY, CT, and NJ" is similar to "NY, CT and NJ" the extra comma could represent 1 or 0. Taking the amount of data in consideration, this method produces smaller amount comparing to the media cover. Careful should taking when using this method as inconsistence of using commas or other punctuations could reveal the methods and make it suspicions [9].

Line Shifting method make shifting of the vertical line which make a space for data hiding along that line using unique shape of text. However, line shifting could be detected by some programs such optical character recognition. Moreover, in case of retyping the document the hidden data will be lost. An example of vertical line

shifting is in Figure 3 where there is a small vertical line shifting (1/300 inch). It is hard to notice the vertical line in normal situations.

.

This is a method of altering a document by vertically shifting the locations of text lines to uniquely encode the document. This method provides the highest reliability for detection of the embedded code in images degraded by noise. To demonstrate that this technique is not visible to the casual reader, we have applied line-shift encoding to this paragraph.

Figure 3. Line shifting where second line is shifted up 1/300 inch [7]

Word shifting method of the words shifting is to make space between words that make us use it to hide data. The space between words is small enough to be not normally noticeable. However, it could be detected using Optical Character Recognition when detection the sequences between words..

Transmitting SMS recently is done by using abbreviations which could provide simplicity and some security in some applications such as Email, Internet chatting and mobile messages Table III.
Using abbreviation in SMS messages saves time of typing complete, space and to overcome the keyboard limitation characters. Some of the algorithms that use abbreviation also use some numbers to transfer some information. [12].

**Table III. Some SMS Abbreviations**

| Abbreviation | Meaning |
|---|---|
| ADR | Address |
| ABT | About |
| URW | You are welcome |
| ILY | I love you |
| EOL | End of lecture |
| AYS | Are you serious? |

TeX ligatures method there are some special groups of text letters are joined together to form a glyph. The bit could be hidden after the algorithm finds the available ligature, one bit will be hidden for each one [5].
Using the same algorithm, it could be applied to the Arabic character "ﻉ" or " ﻟ". However, there are two problems with this algorithm. First, the increase of the text file when applying the extension in the text. Second, the OCR could recognize the hidden date after noticing the font change [6][5].
Vertical displacement of the points has better performance when applying in the pointed letters. In the English language there only two dotted letters {i, j} comparing with Arabic and Persian languages there are 13 dotted letters in Arabic out of 26 total letters and 22 dotted letters in Persian out of 32 letters. The algorithm used in the method encodes 1 to shift up the point else it encodes 0. A big number of bits could be encoded using this method; to detect the changes a powerful OCR is required, but retyping the message will remove the entire message [7].

Figure 4. Vertical shifting point [7]

In Arabic language there are Diacritics (Harakat) which are used to distinguish between different words that has the same letters and also for pronunciation of the word and letters, thought these diacritics are optional.
When reading the Arabic script, the diacritics are not required for most of the words because it depend on the Arabic grammar. From the Arabic text diacritics, the most occurrence is Fatha " ◌َ " which it has the value of 1 when it used or 0 when it's not used. The algorithm that uses the diacritics to hide data enhances the using of cover media. However, the carrier file size could be reduced depending on the hidden message. Meanwhile, Optical Character Recognition could detect that there are data hidden. The drawback from this method is retyping the document will remove the message [8].
In [11] the author's studies adding extra diacritics to text to increase the robustness of data and also used other scenario to hide the data in image.
In [17] the method was using the diacritics for data hiding and it shows the diacritics if it encoded with 1 or not showing it if it encoded with 0. The disadvantage of this method is that it could be detected when it compared with original text.

By using one of Arabic language characteristics which is called (Kashida) the extension letter, which can only be places between the letters and not at the beginning or at the end of the words. Un-pointed letters with the extension could be used to store data as zero and pointed letters with the extension to be used to store 1. However, new Unicode will be added (0640).

| Watermarking bits | 110010 |
|---|---|
| Cover-text | من حسن اسلام المرء تركه مالا يعنيه |
| Output text | من حسن اسلام المرء تركه مالا يعنيه<br>1 1 0 0 1 0 |

Figure 5. Kashida character after pointed letter [14]

As in Figure 5 not all the letters can hide data so, some letter will be used and some will not. Stegoanalysis could be suspicious about data being hidden between the content.[15]

Using Pseudo-Space and Pseudo Connection characters method is also called zero width non-joins (ZWNJ) and zero width joiner (ZWJ) characters. In this method there is a classification of join and non-join letters. To hide 1 bit zero width is added other width for 0 bit to be hidden [16].

**3. PROPOSED ALGORITHM**

In our proposed algorithm we try to hide data inside a word file without any change in the file format. Stegoanalysis will try to analysis file containing and formatting, if there is any change about file format he can catch the hidden data. In our algorithm we will use Zero Width Character (Ctrl+ Shift +I). ZWC it's a Unicode character (U+200B), that does not occupy any space or file formatting. By adding ZWC before and after space letter can hide data.

Microsoft word it's a possible to count number of characters in any file without count space, so also after we add ZWC will not increase number of letters. In our algorithm we will measure file space probability in the file to choose which one is the best file can be used.

$$\text{Space Ratio} = \frac{\text{Number of Spaces}}{Total\ Number\ of\ charcter} \quad (1)$$

In the Figure 6 we represent C++ code to return the best file have success space ratio to insert data on it.

By knowing number of character and number of space we can decide which file can carry our hidden data.

In the figure 6 we represent C++ code to return the best file have success space ratio to insert data on it.

By knowing number of character and number of space we can decide which file can carry our hidden data.

Algorithm I Data Hidden

Input :-File, hidden bit's
Output :- Stego file(embedded ZWC inside file)
Step1:- choose any text file
Step2: Measure Space ratio in selected test file if it success continue otherwise back to step1
Step 3. Repeat while !(EOF)// repeat until finish hidden file
Step4: Embedded Hidden data inside selected file as
Step 4a. select space
Step 4b.pack out first two hidden bit
If 00 then no ZWC before space
Else if 01 then there is no ZWC after space
Else if 10 then there is ZWC before space
Else ZWC after space letter.
Step 5: Go to step 3
Step 6: save file and send it.

As Algorithm I show how to hide data inside it.

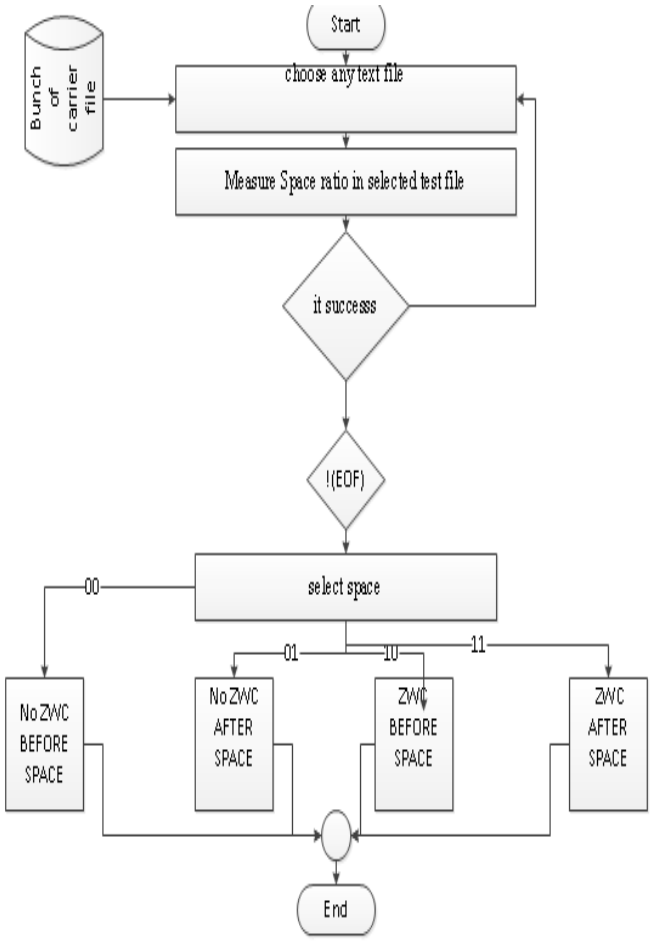


Figure 6. Flow chart for algorithm procedure

Algorithm II Data Extraction

Input:-Stego file
Output: - Secure data
Step1:- open text file
Step 2. Repeat while!(EOF)// repeat until finish hidden file
Step 3b.pack out letter before and after that space
If there is no ZWC before space then hidden data =00
Else if there is no ZWC after space hidden data =00
Else if there is ZWC before space hidden data =01
Else ZWC after space letter hidden data =11.
Step 4: Go to step 3
Step 5 : save file.

**4. DISCUSSION AND ANALYSIS**

By hiding data in file the time complexity will be $= M + (N/2)$ where M is number of carrieries files, and N number of bits wan to embed inside

file. Where best case is $1 + (N/2)$. Table represent the simulation result of file size and number of bits add to carries web pages. ZWC space algorithm had the following advantages

1. File formatted will not be change.
2. Can be applied for any code (Unicode, ASCII). In other word this algorithm represents general form for any language.
3. As figure show file size will not incredible affected.

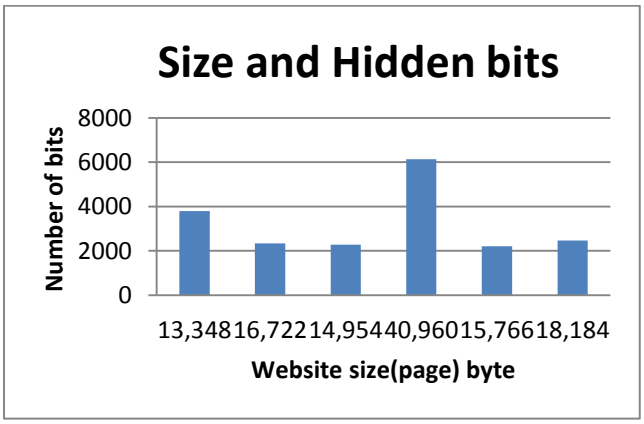


Figure 7 relations between different websites size and amount of data can be hidden inside it.

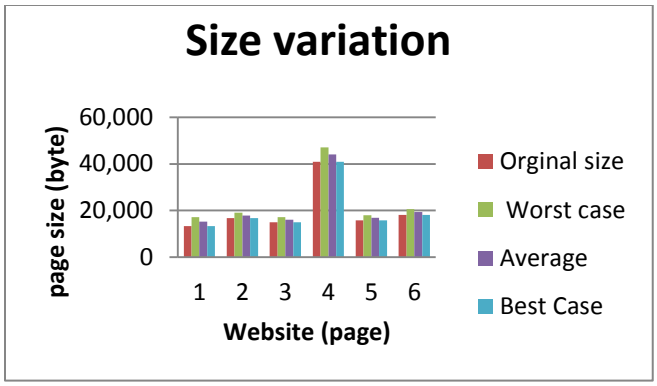


Figure 8 shows the difference between web size in three situation

**5. Conclusion**

Hiding data in different cover media represent one of challenging security issues. One of the difficult media to use for hiding data is a text, where embedding data may affect the text format. The file size and format change will increase the probability of being discovered using Stegoanalysis tools and this will lead to reveal the hidden data. The algorithm presented in this paper use Unicode letter Zero width character and space in text to hide data without any effect in file format. More over file size sometimes not effected depending on hidden data to be insert inside it. Where effect on the file size or text format in any abnormal or suspicious way. Comparison of this algorithm with other techniques in the same categories .our algorithm does not change any format inside files or incredible change in file size. which reduce probability suspicion by Stegoanalysis tools. Furthermore, it can be applied in different languages.

Table III represent the simulation result of file size and number of bits add to carries web pages

| Web | web | Original size | Number of hidden bits | Worst case | Average | Best Case | Space Ratio |
|---|---|---|---|---|---|---|---|
| 1 | www.bbc.co.uk | 13,348 | 3798 | 17146 | 15247 | 13,348 | 28.45 |
| 2 | http://www.cnn.com | 16,722 | 2340 | 19062 | 17892 | 16,722 | 13.99 |
| 3 | http://www.nytimes.com | 14,954 | 2286 | 17240 | 16097 | 14,954 | 15.29 |
| 4 | http://education.astate.edu | 40,960 | 6138 | 47098 | 44029 | 40,960 | 14.99 |
| 5 | http://www.post-gazette.com | 15,766 | 2214 | 17980 | 16873 | 15,766 | 14.04 |
| 6 | http://www.aljazeera.com | 18,184 | 2468 | 20652 | 19418 | 18,184 | 13.57 |

**6. References**


[1] Aelphaeis Mangarae "Steganography FAQ," Zone-H.Org March 18th 2006

[2] S. Dickman, "An Overview of Steganography," July 2007.

[3] V. Potdar, E. Chang. "Visibly Invisible: Ciphertext as a Steganographic Carrier," Proceedings of *the 4th International Network Conference (INC2004), page(s):385–391, Plymouth, U.K., July 6–9, 2004*

[4] M. Al-Husainy "Image Steganography by Mapping Pixels to Letters," *2009 Science Publications*

[5] M. Shahreza, S. Shahreza, "Steganography in TeX Documents," *Proceedings of Intelligent System and Knowledge Engineering, ISKE 2008. 3rd International Conference, Nov. 2008*

[6] 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.*

[7]M. H. Shahreza, M. S. Shahreza, "A New Approach to Persian/Arabic Text Steganography " Proceedings *of 5th IEEE/ACIS International Conference on Computer and Information Science 2006*

[8] M. Aabed, S. Awaideh, A. Elshafei and A. Gutub "ARABIC DIACRITICS BASED STEGANOGRAPHY" Proceedings *of IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*

[9] W. Bender ,D. Gruhl ,N. Morimoto ,A. Lu "Techniques for data Hiding" Proceedings *OF IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996*

[11] M. Nosrati , R. Karimi and, M. Hariri ," An introduction to steganography methods" *World Applied Programming, Vol (1), No (3), August 2011. 191-195.*

[12] M.H. Shirali-Shahreza, M. Shirali-Shahreza, " Text Steganography in chat" Proceedings of *3rd IEEE/IFIP International Conference in Central Asia on Sept. 2007*

[13] Adnan Abdul-Aziz Gutub, Wael Al-Alwani, and Abdulelah Bin Mahfoodh " Improved Method of Arabic Text Steganography Using *the Extension „Kashida" Character" Bahria University Journal of Information & Communication Technology Vol.3, Issue 1, December 2010*

[14] A. Gutub, L. Ghouti, A. Amin, T. Alkharobi, M. Ibrahim. "Utilizing Extension Character Kashida with Pointed Letters for Arabic Text Digital Watermarking". Proceedings *of the International Conference on Security and Cryptography, Barcelona, Spain, July 28-13, 2007, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications. pages 329-332, INSTICC Press, 2007*



[13] Adnan Abdul-Aziz Gutub, Wael Al-Alwani, and Abdulelah Bin Mahfoodh " Improved Method of Arabic Text Steganography Using *the Extension „Kashida" Character" Bahria University Journal of Information & Communication Technology Vol.3, Issue 1, December 2010*

[14] A. Gutub, L. Ghouti, A. Amin, T. Alkharobi, M. Ibrahim. "Utilizing Extension Character Kashida with Pointed Letters for Arabic Text Digital Watermarking". Proceedings *of the International Conference on Security and Cryptography, Barcelona, Spain, July 28-13, 2007, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications. pages 329-332, INSTICC Press, 2007*



[15] Adnan Abdul-Aziz Gutub, and Manal Mohammad Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions " *World Academy of Science, Engineering and Technology 27 200*

[16] H. Shahreza, M. Shahreza "STEGANOGRAPHY IN PERSIAN AND ARABIC UNICODE TEXTS USING PSEUDO-SPACE AND PSEUDO CONNECTION CHARACTERS". *Journal of Theoretical and Applied Information Technology.*

[17] M. Bensaad, M. Yagoubi "High Capacity Diacritics-based Method For Information Hiding in Arabic Text" *2011 International Conference on Innovations in Information Technology.*

[18] A. Azmi and A. Alsaiari "Arabic Typography: A Survey" *International Journal of Electrical & Computer Sciences IJECS Vol: 9 No: 10*