Hybrid Arabic Text Steganography

Haya Mesfer Alshahrani
Princess Nourah bint Abdulrahman University
Riyadh, Saudi Arabia
Email: hmalshahrani [AT] *pnu.edu.sa*

George R S Weir
Strathclyde University
Glasgow, United Kingdom

Abstract—**An improved method for Arabic text steganography is introduced in this paper. This method hides an Arabic text inside another based on a hybrid approach. Both Kashida and Arabic Diacritics are used to hide the Arabic text inside another text. In this improved method, the secret message is divided into two parts, the first part is to be hidden by the Kashida method, and the second is to be hidden by the Diacritics or Harakat method. For security purposes, we benefitted from the natural existence of Diacritics as a characteristic of Arabic written language, as used to represent vowel sounds. The paper exploits the possibility of hiding data in Fathah diacritic and Kashida punctuation marks, adjusting previously presented schemes that are based on a single method only. Here, the secret message is divided into two parts, the cover text is prepared, and then we apply the Harakat method on the first part. The Kashida method is applied on the second part, and then the two parts are combined. When the hidden 'StegoText' is received, a split mechanism is used to recover the original message. The described hybrid Arabic StegoText showed higher capacity and security with promising results compared to other methods.**

Keywords- Steganography, Arabic Text Steganography, Information Hiding, Diacritics Steganography, Kashida Steganography, Cover Text, StegoText.

I. INTRODUCTION

The wide spread reliance on networks and information interchange, requires that information and data be exchanged securely and received reliably. Encryption, Watermarking, and Steganography are security methods for preserving information through communication [1].

Steganography is a security technique wherein data is embedded within other data. In principle, Steganography features confidentiality, integrity, and availability, which makes it especially capable among methods of secure data exchange. The word 'steganography' comes from the Greek words "Stegano", which means hidden, and "Graphos" , which means writing [2].

In steganography, cover data is needed to hide information. Usually the cover or stego-cover is an image, sound or text [1] [2]. It is proven that steganography shows immunity against man-in-the-middle attack and eavesdropper [3]. A categorization of Steganography techniques has been proposed [2] and is illustrated in Figure 1.

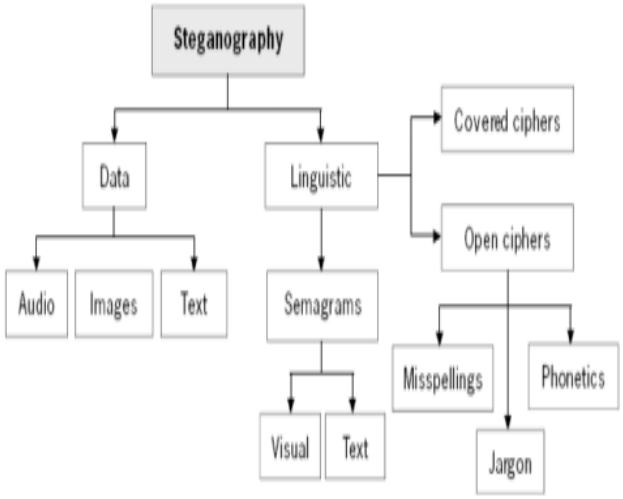

Figure 1: Steganography Categorization [2]

In Figure 1, steganography is divided or categorized based on the data used for hiding information and the linguistic techniques used to hide information. In our work, we are focusing on steganography by text data.

When using steganography we have to consider three issues: the maximum amount of information that can be hidden (Capacity), confidentiality and immunity to eavesdroppers (Security), and how much modification the carrier can stand before hidden data is destroyed (Robustness) [4].

The Arabic language has 28 different characters. Every character has its own shape depending on its location in the word. Arabic characters are distinguished by their ability to connect to each other in shape to formul the word. In some cases, and for aesthetics and decoration, we can extend the connection between characters using "Kashida". For example,

in the word "الرجال" which consists of the characters: "الـ", "ر", "جـ", "ا", and "ل", we can extend this word by inserting "Kashida" between "الـ" and "ر" without affecting the meaning. Moreover, "Kashida" could also be inserted between "جـ", "ا", so that the original word becomes "الـرجـال" and still has the same meaning [5]. A list of Arabic characters and their possible shapes is shown in Figure 2.

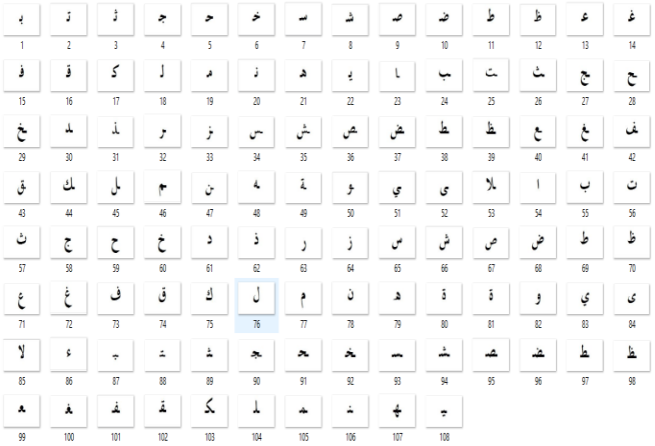

Figure 2: All Arabic Letters Variations

Most steganography approaches hide data by making minimal modifications to the forming of the characters or spaces. Some use "Kashida" after a pointed letter to hide a secret bit of '1' and un-pointed letter to hide a secret bit of '0'. However, this has drawbacks in terms of capacity and security [6].

Others, depend upon a single diacritic technique to hide secret messages in Arabic text. The diacritic is chosen based on a study of the percentage diacritic use across all Arabic diacritics. Usually, "Fatha ◌َ" is the most used diacritic in Arabic text while the second most used diacritic is "Kasrah ◌ِ" [2]. Table 1 shows the 8 basic Arabic diacritics.

Table 1: 8 Arabic Main Diacritics

English Name	Arabic Names	Shape
Fatha	فتحة	◌َ
Dahmmah	ضمة	◌ُ
Kasrah	كسرة	◌ِ
Tanween Fath	تنوين فتح	◌ً
Tanween Kasr	تنوين كسر	◌ٍ
Tanween Dahm	تنوين ضم	◌ٌ
Sukkon	سكون	◌ْ
Shaddah	شدة	◌ّ

Our method for steganography in Arabic text aims to hide the secret message in Arabic text using both "Kashida" and Diacritic methods, based on single diacritic "Fatah". The rest of this paper will explain how to use both methods in a hybrid manner. Section 2 will review Arabic and other text steganography. Section 3 details our methodology and Section 4 details our results along with their analysis. Section 5, concludes our paper.

II. Related Work

Several studies are interested in hiding one text inside another, using a variety of different linguistic method. These studies characterise linguistic steganography into two different classes. The first class is build upon the syntax and the other class is built upon semantic approaches [16][17]. For example, [16] and [18] introduced an algorithm based on synonyms, which aimed at concealing data, and the algorithm was applied in two phases: phase one aims to convert the hidden message into binary codes (using ASCII). After that, by relying on a synonyms file, the sender and recipient should have the same word list to achieve the objective of encryption and decryption of the message in phase two. In the case where the sender inserts a Zero, then it is not necessary to replace a word, otherwise, the synonym file is used as a basis for word replacement. This approach is repeated until reaching the end of the secret message, and the receiver is able to decryp the message based on an inverse strategy.

Another study [19] introduced a similar mechanism, whereby the proposed algorithm includes three input sources: (natural language, secret message, and the key), and there is one output which is called Stego-object. The system recognizes every word and in which group it belongs, by generating lexical replacements group and variant forms of the similar word. To embed the correct word in the carrier file and taking the context into account, the researchers used a lexical analyzer for Chinese language in the proposed system. The steganography algorithm was based on three inputs which show the source natural language text, the information to hide, and the key, and one output which shows the stego-text with embedded data. The steps of the steganography algorithm are as follows:

Embedding data, encrypting the information into the binary bit sequence with the key, which is the embedded data.

Text preprocessing, where the English character, Chinese and English punctuation, blank space and the new line character are the inter-sentence symbols, and every sentence is segmented by using the Chinese lexical analyzer ICTCLAS.

Substitution, where every sentence from left to right is scanned, word by word, and look up the word to determine whether it belongs to the lexical substitution set.

Another study [20] introduced a method based on Line Shifting by shifting lines vertically. This method is employed in many systems, so that it is possible to pass data via the carrier. The main weakness of this approach is the possibility of detecting line shifting when using character recognition programs. Also, when retyping the carrier file, the hidden data will be destroyed.

Researchers in [21], adopted a syntactic method based on punctuation. The secret message is conveyed by adding punctuation in suitable locations to hide the data. One of the strengths of this method is that it does not impact on the meaning of the carrier message nor affect data embedded within it. The study showed that automatic detection of geometric and non-geometric modifications applied to the host signal after data hiding is a key data-hiding technology. The best trade-offs between bit rates, robustness, and perceive-ability need to be defined experimentally.

In [22], the researchers used emoticon-based steganography in order to facilitate secret chatting. They depended on feelings from the emoticons to pass the secret message among the parties. They rely on the fact that most people nowadays use emoticons in their chats. The proposed algorithm classified symbols semantically and then controlled the symbol order by using a secret key. Four groups of emoticons were created for hiding data. For example, when passing the symbol in the beginning of a sentence, a 0 value bit would be passed, otherwise, passing a 1 value bit. There are other approaches based on symbol order, and extracting data from the symbol based on the symbol order in its group. This method is considered robust and beneficial, so it is used in chat systems. Also, their other chat systems allow users to generate their own customized symbols.

Another study [23] introduced an approach called Harakat, based on Arabic language attributes known as Diacritics (Harakat), predicated upon the fact that diacritic use in Standard Arabic language is optional. The introduced paradigm aimed at assigning one bit value to the diacritic Fatha and the remaining seven diacritics represent a bit value of 0. When passing 1, they keep that Harakat, otherwise remove it to pass zero. The study generated pseudo-random sequences to embed into the cover media, the sequence used in this example is: E7 - 30 - E9 - IC - A4 - FC - B8 - B9 - AF - IF - OB - D9 - 22 represented in Hexadecimal format.

Also, researchers in [24] introduced a new approach for Arabic Language text steganography based on diacritics or Harakat, which gives vowel sounds. This feature of Arabic test is rarely used, and is not suitable for religious and official texts. There are 8 diacritics used in the Arabic language and the most used is Fatha. This steganography method aims at using fatha to show 1 and 0 for any one of the remaining seven diacritics. Moreover, a value of 1 is the hidden bit, and the first fatha can be found and any other Harakat before it can be removed to hide 1. This method is beneficial in terms of the reusability since the same cover text can be used for more than one hidden message. Furthermore, this method does need sophisticated software and this method is widely used.

The research reported in [25] presented an algorithm based on pointed letters which are the followings: (ن, ف, غ, خ, ظ, ت, ث, ض) by vertical shifting of the point(s). No vertical shift will happen in the case of passing a 0, otherwise passing 1 by vertical shifting. Figure 3 shows an example to illustrate this steganography process.

Secret bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Steganographic text	من حسن اسلام المرء تركه مالا يعنيه 1 1 0 0 1 0

Figure 3: Steganography example adding extensions after letters [25]

The cover text is scanned from right to left – the regular Arabic text direction. The first un-pointed letter in the cover-text is found to be the first, known as 'meem'. This 'meem' should hold the first secret bit '0' indicated by adding an extension character after it. The second secret bit is '1' and the second letter of the cover-text (known as 'noon') is pointed. However, this letter position prevented extension, which forced the researchers to ignore it, and the next possible pointed letter to be extended is 'ta'. The same steganography example of securing: 110010 in the Arabic text, illustrated earlier, is readjusted assuming the extensions added are before the letters, as shown in Figure 4.

Secret bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Steganographic Text	من حسن اسلام المرء تركه مالا يعنيه 1 1 0 0 1 0

Figure 4: Steganography example adding extensions before letters [25]

In contrast, a different approach [26] developed a method based on vertical point shifting. This method depends on using multipoint characters only, and also takes into consideration the shifting and distance among points for transmitting two bits in every multipoint letter.

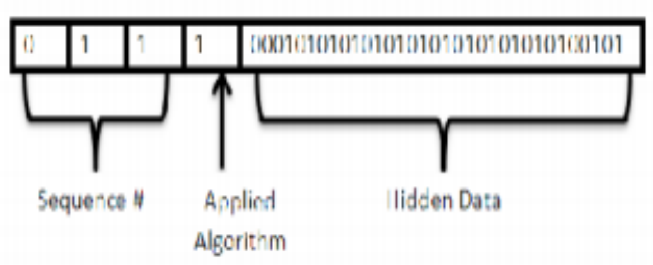

Figure 5: State Diagram for ZKS [26]

The proposed method includes two main phases. The first phase sends embedded 3 bits to recognize the sequence number of that message. Depending upon the message to be hidden, it is possible to add a bit to increase the scope for parallel messages. In the second phase, algorithm permutated-fragmented messages and a randomization function choose which application to use. Therefore, the first four most significant bits determine the sequence message and the last bit selects the applied algorithm, as shown in Figure 5. Each message has a different Stego key regardless of routing path, which increases the confusion scope against steganalysis.

A further study [27] suggested using Unicode methods for steganography in Arabic and Persian texts. Two characters were required in this approach: Zero Width Non Joiner (ZWNJ) and Zero Width Joiner (ZWJ). Using these characters, it simply applies a Unicode method characterized by providing high hiding capacity, since it hides one bit in every letter. Also, this method is beneficial because it does not affect the original text and also provides high transparency. For extracting the information from the text having hidden information (stegotext), the researchers respectively investigated the letters of the text words. If after the letter there is one or three ZWNJ or one ZWJ character, it means that the bit 1 is hidden in that word, but if after the letter there is no ZWNJ and ZWJ or there are two ZWNJ, it means that the bit 0 is hidden in this letter. By putting all the bits of 0 and 1 next to each other it is possible to extract the hidden information from the carrier text.

A different steganography approach, reported in [28], suggested a new technique based on bit optimization using mapping tables. The proposed approach was found attractive and it is possible to modify it to enhance the security and capacity attributes in Arabic and other languages that have similar properties. In the proposed method, the secret object is hidden in the form of zeros and ones that show 16-bit Unicode for every character based on the UTF-8 encoding which depends on the 16 bit (ASCII code). The proposed method aimed at adding one Kashida representing secret bit = 0 and two consecutive Kashidas when bit = 1. The Kashida is placed after any letter that can hold it. The optimization part of the algorithm deals with the message to be hidden. Arabic language has 28 main letters, but there are special forms of a letter, as with the letter "Alef (ا , إ , أ) :"(ا , …), which are used in Arabic writing and each one has a UTF-8 representation. So, the number of letters and forms add up to more than 32 and less than 64. Since each letter and form is represented by 16 bits, the researchers used a mapping table in which each letter was assigned, instead, a 6-bit code to save 10 bits. In the mapping table, they assigned the 6-bit codes starting from 000000 and incremented by one to all letters and forms ordered alphabetically.

An alternative steganography method that depends on changing the pixel values in an image with the message text bits, was introduced in [29]. This method also hides text files of different sizes into image files for authentication of computer login and logout, to create more secure systems. Only authorized users can hide and disclose the message. Text files of different size were used to test the system with the result that the system satisfied all requirements of steganography and proved to be secure. Figure 6 shows the proposed method, in which the last bit is changed.

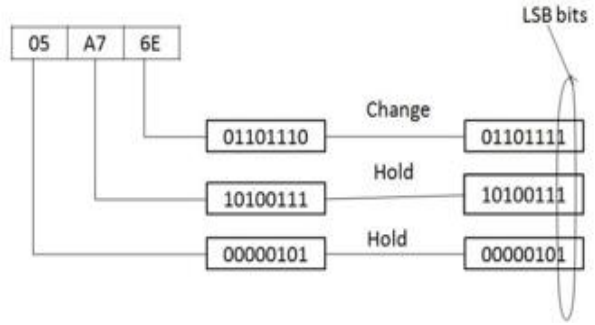

Figure 6: The Proposed Method

Two publications [30] [31] described an experimental system developed as a test bed for textual steganography. The proposed system enabled users to explore the application of part of speech (POS) tagging. The main objective of the system is the ability to hide data in text in such a way that hidden content is undetectable. The researchers applied three experimental systems: system 1 which shifts the source text into replacement words by selecting words of the same POS types from the tagged Brown Corpus. The tagged n-grams generated from the Brown Corpus are used in text shifting. Therefore, the number of replacement words will expand, because a single source word could be changed by 1, 2 or 3 words. System 2 where the text shifting process is similar to System 1 except that only the content words in the source text are POS tagged and shifted. The function words remain unchanged. Then, system 3 shifts the source words into replacement words from the Brown Corpus solely on the basis of n-gram frequency. This study is suitable for English language, and also plain text can be used as a carrier text.

In summary, a comparison is conducted based on the weaknesses and strengthens of each approach, as shown in Table-2.

Table 2: Comparison Between Methods

Approach	Strengths	Weaknesses	Evaluation
Linguistic syntax-approach [16]	- This approach converted plain text into ASCII Bit stream. It uses sounds. - Also, it uses a 'semantic method' which depends on word synonyms.	- Slow execution - It may change message meaning. - High error percentage, because the method depends on meaning (linguistic) and sound.	- This technique is suitable for Arabic language. - It affords plausible carrier messages.
Line shift-approach [20]	- Marking the text line in two ways: a) vertically by use line shifting, and b) horizontally by use of word shifting. - Easy to execute.	- Errors may occur on horizontal and vertical profiles. - Modifying or rewriting the text electronically may destroy the hidden information. - Using character recognition such as OCR, may damage the visual shapes that are hiding data and they may not be retrieved accurately.	- It does not give 100% secret message relay.
Syntactic-approach [21]	- This method does not affect the message meaning or the data embedded inside it. - Uses punctuation.	- It senses that it has encoded message. - The word processor may inadvertently change the number of spaces, destroying the hidden data.	- It is more suited to English language than Arabic language. - It affords plausible carrier messages.
Semantic Method-approach [21]	- It encodes binary data by exploiting ambiguity of form. - This method can assign two synonyms; primary or secondary value.	- Some problems occur in the case when the nuances of meaning interfere with the desire to encode data. - It may change the meaning of the text.	- It is more suited to English language than Arabic language.
Emotional Icons-Approach [22]	- It embeds secret information into emotional icons. - Using the meaning of emotional icons. - Easy to extract the secret message within emotional icon based on its meaning.	- It is possible to guess the secret message based on the emotional icon meaning based on its type. - It is limited, where it includes limited messages' meaning.	- It depends on images, and it does not use text as a carrier, so it is not suited.
Harakat-Approach [23]	- It is not affected by printing, using OCR methods, font changing, and retyping, as long as the medium can show Arabic. - It has fast execution. - It is simple to implement manually. - This method provides the highest capacity. - It is more secret because it is uncommon nowadays to send diacritized text.	- It needs use a fully diacritized Arabic text as the cover media. - It is not suitable if the cover message uses religious or political documents.	- It is well suited to Arabic language. - It affords plausible carrier messages.
Diacritics-Haraka – Approach [24]	- It has fast execution. - It is simple to implement manually. - No lost data when using character recognition such as OCR.	- It is not suitable if the cover message uses religious or political documents. - It attracts the attention of the reader.	- It is better suited to Arabic language than English language. - It affords plausible carrier messages.

Letter Points and Extensions – Approach [25]	- This approach is suitable for secret storing of large number of hidden bits in any Arabic text. - This method has no effect on the message content. - This method is beneficial in hidden exchange of data via text documents, and in making secret communication. - This approach is beneficial to other languages that have same characters as Arabic language, like: Persian and Urdu texts.	- Slow execution - This is a novel approach, and it is possible there may be errors in implementation.	- It is secure and suited to English and Arabic languages.
Kashida and Zero width character- Approach [26]	- Has no effect on the word meaning in cases where it joins with other words. - Speedy execution.	- It depends on using same fixed font.	- It is suited to Arabic language only. - It affords plausible carrier messages.
Pseudo-Space and Pseudo- Approach [27]	- It achieves both perceptual transparency and hiding capacity needs.	- Slow	- This technique is suited to English only.
Kashida- Approach [28]	- It depends on ASCII code. - More secure.	- Slow - Message content may be changed.	- It is suited to Arabic language only
Encryption and Decryption Algorithms [29]	- It is more secure because it depends on encryption and decryption methods. - It uses LSB to change bits.	- Slow because it executes encryption and decryption methods as well as LSB method.	- It is suitable to Arabic and English languages.
Textual Steganography [30]	- It uses text as a carrier - The main application frame affords the user a number of useful features - Secure	- The used method is not explained in detail.	- It is suited to English more than Arabic. - It is secure. - It affords plausible carrier messages.

III. METHODOLOGY

Textual steganography depends on hiding secret messages in texts. In our study, the secret Arabic message will be hidden in Arabic text. In this context, "Kashida" and "Harakat" or diacritics will afford our hybrid method. The secret message is divided into two parts. The first part will be hidden using the "Kashida" method and the second part will be hidden using the "Harakat" method. Then, the secret parts will be combined. The proposed hybrid method is shown in

Figure 7.

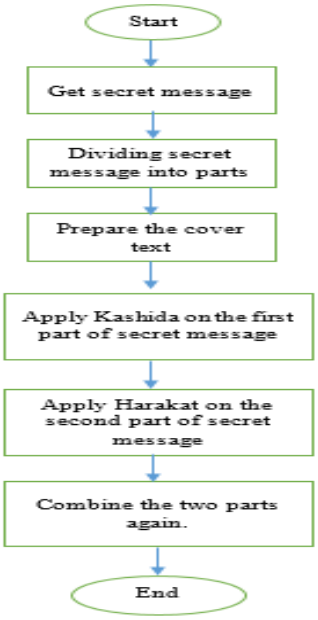

Figure 7: Hybrid Arabic Text Steganography Flow Diagram

In

Figure 7, we start with the secret message, then divide it into two parts. The Harakat method is applied on the first part and Kashida on the second part, after preparing the cover text. After that, we combine the two parts.

A. *Harakat Method*

A simple text steganography method based on "Fatha ◌َ" Arabic diacritic. The method assigns a one bit value, namely 1, to the diacritic "Fatha ◌َ" and the remaining seven diacritics will represent a value of one bit of 0. A fully diacretized Arabic text was used as cover media. To implement this stage of the approach we employ the same steps found in [7]. This method features high capacity, Robustness, low computation power, and simplicity. The Figure 8 is an example which illustrates the Harakat method.

Cover Object	حَدَّثَنَا سُفْيَانُ عَنْ يَحْيَى
Secret Object	E7 (= 11100111)
Stego Object	حَدَّثَنَا سُفْيَانَ عَنَ يَحْيَى

Figure 8: Harakat Example [7].

B. *Kashida Method*

In the Kashida method, adding one Kashida represents a secret bit = 0 while two consecutive Kashidas represents bit = 1. The Kashida is placed after or before any letter that can hold it [8]. The description of the algorithm applied in this stage can be found in [8]. Figure 2 shows an example of this method. As can be seen from the example in Figure 2, Kashidas inserted between the cover text characters one or two depend on the secret bits which represent the stenographic text.

C. *Hybrid Method*

Put simply, any communication system consists of sender, receiver, and communication line. Regardless of the communication line type, the need for security nowadays is a must between any two parties. Figure 9 shows our hybrid Arabic text steganography system which is used to hide secret Arabic text inside another Arabic text (cover text) between sender and receiver.

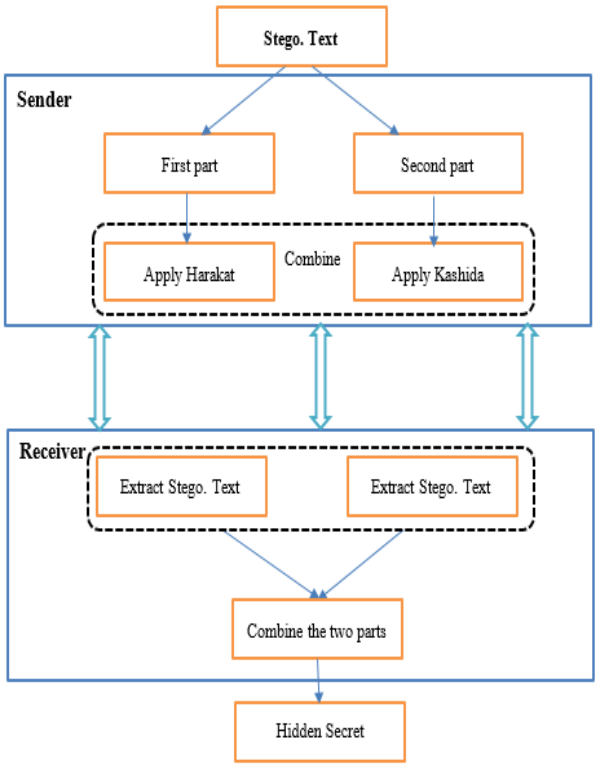

Figure 9: Hybrid Arabic Text Steganography

In Table 3, we introduce an example that shows the original message which will be hidden and sent from the sender to the receiver. Kashida is applied on the first part of the hidden text (which is shown by hidden binary), and this is shown by adding extensions in the Arabic text. Harakat is applied on the second part of hidden text and is shown by diacritics on the Arabic characters. The hidden bits express the hidden text, where each 2 bytes (16 bits) represent one character.

	Stenographic Text
Message Text	لولا ان اشفق على امتي لأمرتهم بالسواك عند كل صلاة
Stego. Technique: **Kashida**	لـولـا أن أشـفـق عـلـى أُمـتـي لأمــرتــهم بـالـسـواك عـنـد كـل صـلـاة
Stego. Technique: **Harakat**	لـولـا أن أُشـفـقَ عَـلـى أُمـتـي لَأمَــرْتُـهم بـالسّـواكِ عِـنـد كُـلِّ صَـلـاةٍ
Hidden Bits	**0110111001100111010101110010110111 1100110111100011110001111101110110 1111011010**
	First part: 0110111001100111010101110010110111 11001 **Second part:** 1011110001111000111110111011011110 11010

Table 3: Converting Text to Steganography Text

To create our approach to hybrid steganography we build a MATLAB program that holds several functions like: the Harakat function (Responsible for applying the Harakat method), the Kashida function (responsible for applying the Kashida operations), a merge function (responsible for merging the two parts after encoding), the Char2bin function (responsible for converting characters into binary). The reason for using MATLAB is to get the benefit of high speed computation and a huge toolbox that facilitates the implementation. Figure 10 is a screen shot of the MATLAB GUI.

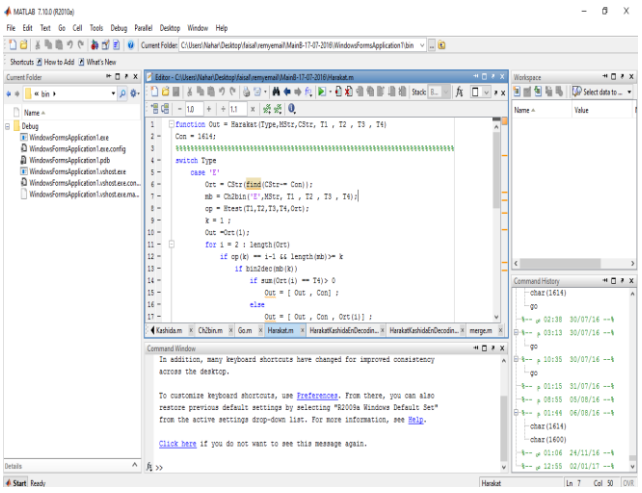

Figure 10: MATLAB GUI and The Code

Another program was written using C#.net to call the MATLAB code because MATLAB does not support Arabic Language. Figure 11 shows the encoding screen which represent the sender.

Figure 11: The Encoding GUI Screen

In the first text box we support the covering text, and in the second box the secret message that has to be hidden in the cover text. After pressing the encoding button, both Kashida and Harakat functions are applied and the result is displayed in the third text box. We reset all boxes and copy the stenographic text by pressing the button "Copy StegoText for Encryption" in order to pass it to the receiver. In Figure 12, we represent the receiver, of the encoded text which has the secret text inside.

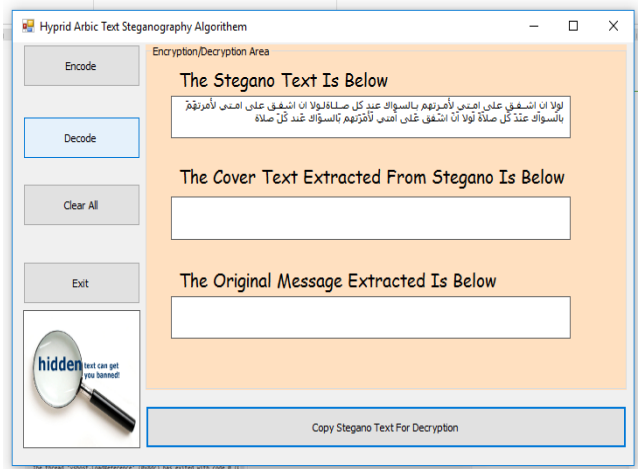

Figure 12: Receiver Receiving the Encoded Text

After pressing the Decode button, a split mechanism is applied to recover the secret from the cover text as shown in Figure 13.

Figure 13: Hidden Text Extraction

IV. Conclusion and Future Work

In our paper, we introduced a new text Steganography in Arabic letters. Our algorithm deals with connected letters by adding Kashida characters and Zero width letters. The ZKS algorithm improves upon the previous version by use of concepts such as parallel connection, permutation, and randomization, to complicate the prospects of steganalysis.

REFERENCES

[1] A. Gutub, Y. Elarian, S. Awaida, and A. Alvi, "Arabic Text Steganography Using Multiple Diacritics," WoSPA 2008 – 5th IEEE Int. Work. Signal Process. its Appl., pp. 18 – 20, 2008.

[2] E. Mohammad Ahmadoh and A. Abdul-Aziz Gutub, "Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance," vol. 3, no. 1, pp. 42–47, 2015.

[3] H. M. Ahmed and M. A. A. Khodher, "Comparison of Eight Proposed Security Methods using Linguistic Steganography Text," Int. J. Comput. Inf. Sci., vol. 12, no. 2, pp. 243–251, 2016.

[4] E. Mohammad Ahmadoh and A. Abdul-Aziz Gutub, "Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance," vol. 3, no. 1, p. 2015, 2015.

[5] A. F. Al-Azawi and M. A. Fadhil, "Arabic text steganography using kashida extensions with Huffman code," J. Appl. Sci., vol. 10, no. 5, pp. 436–439, 2010.

[6] F. Al-Haidari, A. Gutub, K. Al-Kahsah, and J. Hamodi, "Improving security and capacity for arabic text steganography using 'Kashida' extensions," 2009 IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA 2009, pp. 396–399, 2009.

[7] M. A. Aabed, S. M. Awaideh, A. R. M. Elshafei, and A. A. Gutub, "Arabic diacritics based steganography," ICSPC 2007 Proc. - 2007 IEEE Int. Conf. Signal Process. Commun., no. February 2014, pp. 756–759, 2007.

[8] A. Gutub, W. Al-Alwani, and A. Mahfoodh, "Improved Method of Arabic Text SteganographyUsing the Extension „Kashida" Character," Bahria Univ. J. Inf. …, vol. 3, no. 1, pp. 68–72, 2010.

[9] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *IEEE computer*,vol. 31, pp. 26-34, 1998.

[10] Mangarae, A. (2006). Steganography faq. *Zone-H. Org March 18th*.

[11] Dickman, S. D. (2007). An Overview of Steganography. *Department of Computer Science, James Madison University Infosec Techreport*.

[12] Changder, S., Ghosh, D., & Debnath, N. C. (2010, November). Linguistic approach for text steganography through Indian text. In *Computer Technology and Development (ICCTD), 2010 2nd International Conference on* (pp. 318-322). IEEE.

[13] Morkel, T., Eloff, J. H., & Olivier, M. S. (2005, June). An overview of image steganography. In ISSA (pp. 1-11).

[14] Al-Husainy, M. A. (2009). Image Steganography by mapping Pixels to letters. *Journal of Computer science*, *5*(1), 33.

[15] Potdar, V., & Chang, E. (2004). Visibly Invisible: Ciphertext as a Steganographic Carrier. In *Proceedings of the 4th International Network Conference (INC2004)* (pp. 385-391).

[16] Bhattacharyya, S., Banerjee, I., & Sanyal, G. (2010). A novel approach of secure text based steganography model using word mapping method (WMM). *International Journal of Computer and Information Engineering*, *4*(2), 96-103.

[17] Prasad, R. S. R., & Alla, K. (2011, September). A new approach to Telugu text steganography. In *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on* (pp. 60-65). IEEE.

[18] Velcheru, N. R., & Shulman, D. (2002). Classical Telugu Poetry: an anthology.

[19] Bennett, K. (2004). Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text.

[20] Yuling, L., Xingming, S., Can, G., & Hong, W. (2007, July). An efficient linguistic steganography for Chinese text.

In *Multimedia and Expo, 2007 IEEE International Conference on* (pp. 2094-2097). IEEE.

[21] Low, S. H., Maxemchuk, N. F., Brassil, J. T., & O'Gorman, L. (1995, April). Document marking and identification using both line and word shifting. In*INFOCOM'95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE* (pp. 853-860). IEEE.

[22] Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM systems journal*, *35*(3.4), 313-336.

[23] Wang, Z. H., Chang, C. C., Kieu, T. D., & Li, M. C. (2009, November). Emoticon-based text steganography in chat. In *Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on* (Vol. 2, pp. 457-460). IEEE.

[24] Aabed, M., Awaideh, S. M., Elshafei, A. R. M., & Gutub, A. (2007, November). Arabic diacritics based steganography. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on* (pp. 756-759). IEEE.

[25] Shakir, A. C., Xuemai, G., & Min, J. (2010). Chinese language steganography using the arabic diacritics as a covered media. *International Journal of Computer Applications IJCA*, *11*(1), 24-28.

[26] Gutub, A., & Fattani, M. (2007, May). A novel Arabic text steganography method using letter points and extensions. In *WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria* (pp. 28-31).

[27] Odeh, A., & Elleithy, K. (2012). Steganography in Arabic Text Using Zero Width and Kashidha Letters. *International Journal of Computer Science & Information Technology (IJCSIT)*, *4*(3), 1-11.

[28] Shirali-Shahreza, M. H., & Shirali-Shahreza, M. (2008). Steganography In Persian And Arabic Unicode Texts Using Pseudo-Space And Pseudo Connection Characters. *Journal of Theoretical & Applied Information Technology*, *4*(8).

[29] Gutub, A. A. A., Al-Alwani, W., & Mahfoodh, A. B. (2010). Improved method of Arabic text steganography using the extension 'Kashida'character. *Bahria University Journal of Information & Communication Technology*, *3*(1), 68-72.

[30] Kumar, S., Singh, G., Kumar, T., & Nehra, M. S. (2013). Hiding the Text Messages of Variable Size using Encryption and Decryption Algorithms in Image Steganography. *International Journal of Computer Applications*, *61*(6), 47-52.

[31] Morran, M., & Weir, G. R. (2010). An Approach to Textual Steganography. In*Global Security, Safety, and Sustainability* (pp. 48-54). Springer Berlin Heidelberg.