Journal of Information Hiding and Multimedia Signal Processing ©2021 ISSN 2073-4212
Ubiquitous International Volume **12**, Number **3**, Septemberl **2021**

Text Steganography based on Noorani and Darkness

Farah R. Shareef Taka
Directorate of Institutional Development and Government Coordination
Baghdad, Iraq
sabahaliraq2014@gmail.com

Received December 2020; revised April 2021

Abstract. *The modern text hiding is a technique that conceals a secret message within a cover text message using a hidden method to secure confidential information. This paper proposed a new approach to conceal secret bits within Arabic text cover media via using Kashida and Unicode space characters (UniSpaCh), which are used in Microsoft word documents. The selected Unicode space characters have been put into inter-word, inter-sentence, inter-paragraph, and end-of-line spacing to encode secret information while enhancing the efficiency for embedding and imperceptibility of the embedded data. The embedding process is dependent upon the Arabic letter's features, which are called (Noorani and Darkness) letters, where Noorani letters are found in (Al-Fatiha) surah and the beginning of all surahs of Al-Quran Al-Kareem. In contrast, Darkness letters are not found in (Al- Fatiha) surah and the beginning of all surahs. Both the Noorani letters group and the Darkness letters group contain 14 letters. The key strength of this method by using five different cases for hiding secret bits with four scenarios for each case. This new method provides a high capacity for hiding big secret messages in Arabic text cover and secret ratio when compared with the other methods.*
Keywords: Arabic text cover, Noorani letter, Darkness letter, Kashida, UniSpaCh

1. Introduction. Steganography is a method of concealing confidential or very sensitive information and transferring it through a public channel, to make sure that appears to become nothing out of the natural, and their existence is not exposed by a third party [1]. Steganography is usually categorized into four types: text, image, audio and video steganography, according to cover media that are employed to embed data [2]. Text steganography is an approach of concealing a secret text message (or even data) within another text (a covering message) or making a cover message automatically relevant to the original secret message. However, text steganography isn't mostly preferred mainly because of the difficulty in getting redundant bits in the text document compared with other cover media such as images, video and audio files [3]. The text body depositions are related to the content noticeable, whilst in other types such as pictures; the structure of deposition is differing from what was noticeable. Thus, in such depositions, the data might be hidden via generating alterability within the body of the document with no need of performing sensible iterability in the outputs. Unrecognized iterability is usually made to an image file or an audio file, however, in text files, also further character or punctuation may be noted by a random reader. In general, storing text files doesn't require a large memory, as well as being quick and effortless in comparison to some other types of steganography techniques. Text steganography is commonly categorized to three types [4]: Linguistic methods, Statistical and Random generation, and Format based. The strategies utilized to embedded data inside a text, its qualities have to be altered. These features may be one of different strategies such as font resizing, insertion of non-displayed characters or spaces, deliberate misspellings distributed over the text, and so on.

On the other hand, because of a little change in the document, it could be noted by the third party or by the attacker. In order to address this issue, a modification to the document is required in such a way that it is not noticeable to the human eyes, however it is possible to

decode it with a computer [5]. In-text steganography, a message is an unobserved data in the form of plain text, cipher-text, images, or anything which can be encoded into a stream of bits. This message is inserted in a cover of the carrier to create a stego message [6]. The method is probably described as in follows [7]:

$$stego - carrier = Stegokey + Embeddedmessage + CoverMessage \tag{1}$$

There are three main parameters when using steganography: the maximum amount of hidden information (Capacity), immunity and confidentiality to eavesdroppers (Security), and the amount of modification on the carrier can stand before destroying the confidential data (Robustness) [8] The criteria that should be taken in consideration for designing any steganography systems can be described as in follows [9]:

1. Hiding capacity (embedded rate): a cover media can store secret information, which can be measured by the amount of secret data (bits) which are hidden inside the cover media.
2. Security: it is the ability of an eavesdropper or observer to detect or suspect the unusual modification and retrieve hidden information. However, in order to perform high security, it should be reducing the embedding Impact (distortion). This can be achieved by minimizing the distortion between the stego text message and cover text message and by limiting the distortions to the portions for the cover text which are difficult to model.
3. Robustness: the ability to protect the invisible data from destroying, especially when transferred through the internet.

Steganography, similar to a secret communication strategy, gets these types of considerations because no method can address all the protection's concepts. Thus, the critical ideas of security that utilized for steganography to consider is essential concepts of information security needed are as follows [10]:

1. Confidentiality: unauthorized people do not have any idea that there is secret data there, when using steganography.
2. Survivability: Implies that most of the processing data is happening in between the receiver side, and the sender-side does not damage the unseen information. Also, this received data should be readable and extractable.
3. No Detection: Steganography fails when anybody may easily detect where you hide your data and locate your message. Hence, whether or not someone knows the way the insert process of steganography works, it cannot discover there is data in a given file.
4. Visibility: The file of stego should be undiscovered and must not to have any noticeable modifications to the stego file. The aim of steganography is mainly precisely the confidentiality of concealed data. In contrast to cryptography that converts the meaning or content of the secret data, steganography is aimed to hide the existence of this data. Thus, unauthorized people do not have any idea there is secret information there [11].

This work aims to propose a new text steganographic method that can achieve two primary purposes, improve the embedding capacity via increases of embedded data.

inside cover message and the second purpose is to reduce the visibility of the hiding effect. For this purpose, we utilized the Arabic language for being distinguished richness characteristics that make it a strong candidate for hiding data.

2. Literature Survey. Various researchers have done research in the field of text- based steganography. Different methods have been used in order to hide messages in text. However, these methods are classified into three main categories to hide text-in-text messages [12], which are format-based, random and statistical generations, and linguistic methods. Most of these methods are being used for English due to its broad usages in science [13]. However, some of these methods can be applied to all languages as they are not attached to a specific family of languages, and the Arabic language is one of them.

Arabic is a Semitic language, and it has its grammar,spelling, and punctuation rules, its slang

and idioms, and its pronunciation..Modern Standard Arabic involves twenty- eight letters representing twenty-eight consonants, 3 vowels. In addition, every Arabic word has a number of letters that can be connected. This feature of connectivity is suitable to insert an extension character between letters which is called "Kashida".Kashida is an exciting tool used for paragraph justification, which is utilized for extending the horizontal links between joined letters, and it is found initially to decorate the Arabic text in such a way that either format it or modify the lines without affecting the text contents. This feature is significantly associated with the cursive characteristics of Arabic script. It's even more flexible in comparison with alternative letter ligatures and shapes since it is not limited to the number of defined widths. In addition, ithas relatively little effect on the text colour than spacing [14].
In recent years, the researchers offered some methods for concealing secret information in Arabic text cover. These previous works depended on the Arabic language features [15]. However, this approach is probably classified in Format based steganography that was mentioned before. We can classify Arabic text steganography methods to five main categories [16] which are:Dotting-based, Diacritics-based, Unicode-based, Sharp edges- based, and Kashida-based methods. In this work, we focused on works that utilized "Kashida" as an Arabic text stego method, as we intend to improve this method in our work.
In the Arabic language, not all letters accept inserting Kashida before or after them. It can exploit it in Arabic Text Steganography to carry the secret bits in several algorithms. A number of these algorithms that related to our work are described as in follows:

2.1 Pointed or un-pointed letter [17] In 2007, the first new method to conceal a secret message within any letters (pointed or unpointed) rather than pointed letters only. They made use of Kashida with pointed and unpointed letters to conceal secret bits. When the secret bit is one, then kashida is added to a pointed letter. In any other case, it is added to an unpointed letter. Just letters that can handle Kashida are utilized here. Therefore, any letter has no Kashida it doesn't signify any secret bit, that means adequate capacity and yet less security.

2.2 Kashida variation [18] This algorithm had been proposed in order to maximize the robustness. A cover message (text) is segmented to a blocks, and after that hiding bits in every block is based upon one of four scenarios which are: Adding Kashida after dotted letters in order to encode one; otherwise it will be zero, Adding Kashida after undotted letters in order to encode one; otherwise it will be zero, Adding Kashida after letters in order to encode one; otherwise it will pass zero, and Adding Kashida after letters in order to encode one, otherwise it will pass one.

2.3 With specific characters [19] In this work, an invisible watermarking approach that is based on Kashida- marks is proposed. The watermarking key is predetermined, where Kashida is inserted for a bit one and removed for a bit zero. The strategy of adding Kashida is based on inserted Kashida before a specific list of characters **(ر، و ىز، ئ، ؤ ـد، ـذ، إ، أ، آ،)** till reaches to the end of the key. This technique presents a high imperceptibility encoding. However, this is a low capacity watermarking method technique that is more appropriate for copyright protection and authenticity of document.

2.4 Using two diacritics [20] This work improves the method of [21] by utilizing two diacritics to hide secret data inside Arabic text rather than making use of one diacritic. They target concealing data in "Fatah" and "Kasrah" together rather than making use of only "Fatah".

2.5 With normal spaces [22] Hiding secret bits within the Kashida along with the spaces in between words it had been achieved as in follows: In case the secret bit is one, then the Kashida is inserted in between the appropriate Arabic letters which can be accepted till all available Kashida insertions are considered. While, in cases where the secret bit is zero, it holds the word as is with no insertion Kashida. In the space:& If the secret bit is one: Two

consecutive normal-spaces between words are added. & If the secret bit is zero: Single white space is kept as is. The advantage of this method is the increase the capacity via utilized words and spaces between them. Moreover,it is not inserting any character in case where the secret bit is zero, which represents the most bits. However, it uses normal-space (NS), which is a visible character. Therefore, it may bring attention through modifications of the distances in between the words.

2.6 Hides two bits using Kashida [23]. In this work, each existing Kashida can conceal 2 bits rather than 1 bit only.Their method relies on the characteristics of letters, which are (unpointed or pointed) letters. In this method, the cover text is segmented to two equal blocks. Each block is being dealt with differently. However, in several cases, the secret bits sequences and the best suited appearances of the Kashida can be affected on the capacity ratio in this method, hence, it can be considered as a disadvantage and may lead to unwanted results of such cases.

2.7 Using Sun and Moon letters [24] In this approach, the secret message is concealing the text in Arabic language by utilizing sun and moon letters (sun and moon letters are a type described as coronal consonants, while the moon letters are two, ones-pronounced and the other consonants). Four scenarios are applied in order to achieve this purpose. The 1st scenario, Kashida is inserted just after the sun letter in order to hide the 00 secret bits of. In the 2nd scenario, two Kashida are inserted after a sun letter to hide the 11 secret bits. In the 3rd scenario, a Kashida is inserted after a moon letter to hide the 01secret bits. The last scenario is when two Kashida is inserted to hide the 10 secret bits.

2.8 Using Small Space Characters and Kashida [25] This work proposed a new algorithm for steganography that enhanced the length of secret messages that can be inserted in an Arabic text document through using different characteristics (small space and Kashida characters) and properties of the Arabic language. Each existing Kashida can conceal one-bit and existing space for concealing three-bits.

3. The Proposed Method In this paper, the Arabic text steganography method was depending on the features of Arabic letters that are called (Noorani-letters) and (Darkness-letters) by using Kashida's letter and Unicode-spaces. The proposed approach is utilized Kashida, zero-width joiner (ZWJ), and zero-width non-joiner (ZWNJ) letters with four different types of small spaces: narrow space, hair space, narrow No-Break space, and the last Six-PRE-EM space. The codes for the four-space characters are shown in Table 1 and Fig 1. Adding more small spaces to the text of the Arabic language are not notable in the text when compared to other algorithms that have been used more normal spaces.

Table 1: Unicode space characters [26].

Name	Code
Six-PRE-Em-Space	U+2006
Narrow No-Break Space	U+202F
Hair Space	U+200A
Thin Space	U+2009

3.1 Compress the secret message When the secret message is significant, the gzip algorithm gives a good compression ratio to a secret message that embeds in Arabic text cover (Fig. 2).

Space Character	Windows XP		Windows Vista		Windows 7	
	Hide	Show	Hide	Show	Hide	Show
Space	abc def	abc·def	abc def	abc·def	abc def	abc·def
Six-Per-Em	abcdef	abcdef	abcdef	abcdef	abcdef	abcdef
Thin	abcdef	abcdef	abcdef	abcdef	abcdef	abcdef
Hair	abcdef	abcdef	abcdef	abcdef	abcdef	abcdef

Figure 1: The spacing/width accompanied by the three code space characters

Figure 2: Compressing process for secret message model [27].

3.2 Embedding Process The process of concealing depended on features of Arabic letters that are called (Noorani-letters) and (Darkness-letters); these two classifications of Arabic letters are shown below in table 2:

Table 2: Noorani and Darkness letters

Noorani-letters		Darkness-letters	
أ	ق	ب	ز
ح	ك	ت	ش
ر	ل	ث	ض
س	م	ج	ظ
ص	ن	خ	غ
ط	ه	د	ف
ع	ي	ذ	و

However, in the Arabic language, there are some letters cannot accept Kashida to be after the letter, so these shaded letters (in Table 1) are called (Isolated-Noor) and (Isolated-Dark) with seven letters for each one, as shown in table 3:

The proposed method presents 5 cases for concealing the secret bits in the text cover (Arabic text message) with 4-scenarios in each case

3.2.1 The (Noorani-letter) Case: The first scenario is based on adding kashida1(ŏ640) after the Noorani-letter to represent the secret bits (00), the second scenario is based on adding kashida2 (ŏ200D) after the Noorani-letter to represent the secret bits (11), the third and fourth scenarios are implemented by (No Embedding) any character when the secret bits are (01) or (10), see table 4.

Table 3: Isolated letters (Noorani and Darkness)

Isolated-Noor	Isolated-Dark
ا	د
أ	ذ
إ	ز
آ	و
ء	ؤ
ر	ة
ى	ئ

Table 4: The Embedding Process for case1)

Secret-bits	Action (Embedding)
00	Kashida1-(ـ0640)
11	Kashida2-(ـ200D)
01	No Embedding
10	No Embedding

3.2.2 The (Darkness-letter) Case: The first scenario is based on adding kashida1 after the Darkness-letter to represent the secret bits (01), the second scenario is based on adding kashida2 after the Darkness -letter to represent the secret bits (10), the third and fourth scenarios are implemented by (No Embedding) any character when the secret bits are (00) or (11), see table 5.

Table 5: The Embedding Process for case2)

Secret-bits	Action (Embedding)
01	Kashida1-(ـ0640)
10	Kashida2-(ـ200D)
00	No Embedding
11	No Embedding

3.2.3 The (Isolated-Noor letter) Case: The first scenario is based on adding Unicode space character (Narrow-No-Break space) after the Isolated-Noor letter to represent the secret bits (00), the second scenario is based on adding (Hair-space) after the Isolated-Noor letter to represent (11), the third, and fourth scenarios are implemented by (No Embedding) character when the secret bits are (01) or (10), see table 6.

Table 6: The Embedding Process for case3)

Secret-bits	Action (Embedding)
00	Narrow-No-Break space(ـ202F)
11	Hair space-(ـ200A)
01	No Embedding
10	No Embedding

3.2.4 The (Isolated-Dark letter) case: The first scenario is implemented by adding (Hair-space) after the Isolated-Dark letter to represent (01), the second scenario is implemented by adding Unicode space character (Narrow-No-Break space) after the Isolated-Dark letter to represent the secret bits (10), the third and fourth scenarios are implemented by (No Embedding) character when the secret bits are (00) or (11), see table 7.

Table 7: The Embedding Process for case4)

Secret-bits	Action (Embedding)
01	Hair space-(U+200A)
10	Narrow-No-Break space(U+202F)
00	No Embedding
11	No Embedding

3.2.5 The exist of space letter case: The first scenario is based on replacing the standard space with (ZWNJ) when the secret bits are (00), the second scenario is based on replacing with (PRE-space) when the secret bits are (01), the third scenario is implemented by replacing with (Thin-space) when the secret bits are (10), the fourth scenario is implemented by replacing with (PRE-space+ Thin-space) when the secret bits are (11), as shown in table (8):

Table 8: The Embedding Process for case5)

Secret-bits	Action (Embedding)
00	ZWNJ-(U+200C)
01	PRE-space(U+2006))
10	Thin space(U+2009)
11	PRE-space + Thin space

3.3 The Embedding Process. In the embedding process, the secret message is firstly converted to binary (zeros and ones) representation to be ready for the insertion (2 bits) in the Arabic cover. The checking process is applied for all the letters of Arabic text cover, whether it is acceptable to insert Kashida or not. Some of the Arabic letters are not accepted to insert " Kashida" with it, and we named them (Isolated). They are divided into two parts, one for Noor-letters named as (Isolated-Noor), second for Dark-letters named as (Isolated-Dark). Also, the checking process depends on the existence of space letters, so when the space letter is between two letters, there is a possibility for replacing space letters with ZWNJ or one of Unicode-space (PRE-space, Thin-space, PRE + Thin).

Embedding Algorithm.
Input: secret message(s), Arabic Text Cover(c).
Output: Arabic-Stego (AS).

1. i=counter for cover.
2. Convert secret(s) to bits.
3. Check if:
   (1) C(i)= (Arabic Harkat Group) then, i++.
   (2) C(i)= (Delimiters Group) then, i++.
   (3) C(i)= (CR-courage return) OR (NL-new line) then, i++.
4. If c(i)= (Noorani-letter Group) then:
   (1) If (s(i)=0) and (s(i+1) =0) then put (Kashida1 after character).

(2) If (s(i)=1) and (s(i+1) =1) then put (Kashida2 after character).
(3) ElseIf (s(i)=1) and (s(i+1) =0) OR (s(i)=0) and (s(i+1) =1) then i++.

5. If c(i)= (Darkness-letter Group), then:
   (1) If (s(i)=0) and (s(i+1) =1) then put (Kashida1 after character).
   (2) If (s(i)=1) and (s(i+1) =0) then put (Kashida2 after character).
   (3) ElseIf (s(i)=0) and (s(i+1) =0) OR (s(i)=1) and (s(i+1) =1) then i++.
6. If c(i)= (Isolated-Noor Group), then:
   (1) If (s(i)=0) and (s(i+1) =0) then put (Narrow-No-Break space).
   (2) If (s(i)=1) and (s(i+1) =1) then put (Hair space).
   (3) ElseIf (s(i)=0) and (s(i+1) =1) OR (s(i)=1) and (s(i+1) =0) then i++.
7. If c(i)= (Isolated-Dark Group), then:
   (1) If (s(i)=1) and (s(i+1) =0) then put (Narrow-No-Break space).
   (2) If (s(i)=0) and (s(i+1) =1) then put (Hair space).
   (3) ElseIf (s(i)=0) and (s(i+1) =0) OR (s(i)=1) and (s(i+1) =1) then i++.
8. Case of space exists:
   (1) If (s(i)=0) and(s(i+1) =0) then put (ZWNJ).
   (2) If (s(i)=0) and(s(i+1) =1) then put (PRE-space).
   (3) If (s(i)=1) and (s(i+1) =0) then put (Thin space).
   (4) If (s(i)=1) and (s(i+1) =1) then put (PRE-space +Thin space).
9. End.

Fig. 3 shows an example of the embedding process. where the shaded letters point to the existence of Kashida and space to conceal secret bits.

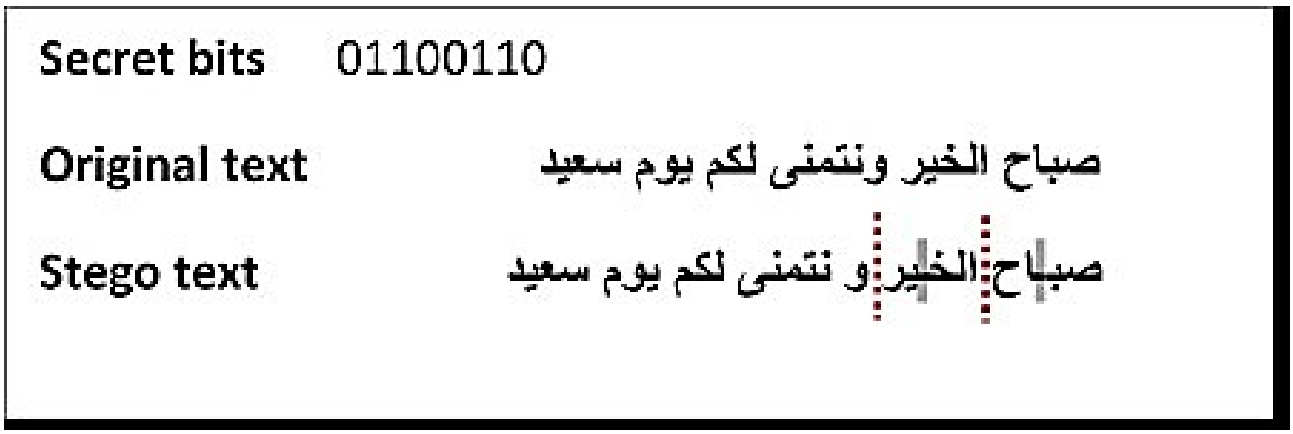

Figure 3: Example of the embedding process.

From Fig.3,If we take the first word (صباح) for embedding the secret bits (01) and the first letter is (ص), but it does not accept Kashida because it is from (Noorani Group), so the scenario is No Embedding (see Table5). The second letter is (ب); it is from (Darkness Group), so the "Kashida" can be inserted after it. Then, the third letter is (ا), which does not accept Kashida because the letter is from (Isolated Noor). The fourth letter is (ح), which does not accept "Kashida" because the letter is from (Noorani Group). The space between two letters (ح) and (ا), so there is an embedding process with (Thin space). The second word is (الخير), No embedding for (01) with the letters (ا) and (ل), but the letter (خ) can accept the Kashida after it. No Embedding for the letters (ر) and (ي). Finally, there is an embedding process with (Thin space) between two letters (ر) and (و) for secret bits (10).

Decoding Algorithm.

Input: Arabic-Stego (AS).
Output: secret message.

1. convert (AS) to array of characters(stego-char/SC).
2. i=counter for cover.
3. check each character of AS with ***Noorani-letter*** Group:
   (1) If (SC(i)= Noorani-letter Group) and (SC(i+1) =Kashida1) then extract (00).
   (2) If (SC(i)= Noorani-letter Group) and (SC(i+1) =Kashida2) then extract (11).
   (3) Else If (SC(i)= Noorani-letter Group) and (SC(i+1) $\neq$ (Kashida1 OR kashida2)) then i++.
4. check each character of AS with ***Darkness-letter*** Group:
   (1) If (SC(i)= Darkness-letter Group) and (SC(i+1) =Kashida1) then extract (01)
   (2) If (SC(i)= Darkness -letter Group) and (SC(i+1) =Kashida2) then extract (10)
   (3) Else If (SC(i)= Darkness -letter Group) and (SC(i+1) $\neq$ (Kashida1 OR kashida2)) then i++.
5. check each character of AS with ***Isolated-Noor*** Group:
   (1) If (SC(i)= Isolated-Noor Group) and (SC(i+1) = Narrow-No-Break space) then extract (00).
   (2) If (SC(i)= Isolated-Noor Group) and (SC(i+1) = Hair space) then extract (11)
   (3) Else If (SC(i)= Isolated-Noor Group) and (SC(i+1) $\neq$ (Narrow-No-Break space OR Hair space)) then i++.
6. check each character of AS with ***Isolated-Dark*** Group:
   (1) If (SC(i)= Isolated- Dark Group) and (SC(i+1) = Narrow-No-Break space) then extract (10).
   (2) If (SC(i)= Isolated- Dark Group) and (SC(i+1) = Hair space) then extract (01).
   (3) Else If (SC(i)= Isolated- Dark Group) and (SC(i+1) $\neq$ (Narrow-No-Break space OR Hair space)) then i++.
7. Check each space exist in AS:
   (1) If SC(i)= ZWNJ) then extract (00).
   (2) If SC(i)= PRE-space) then extract (01).
   (3) If SC(i)= Thin space) then extract (10).
   (4) If (SC(i)= PRE-space) and (SC(i+1) = Thin space), then extract(11).
8. Convert a string of bits to characters.
9. End.

4. Evaluation The main aim of this research is to improve the percentage capacity and hiding capacity. There were many equations which had been used in the proposed method [28].

1. ***Percentage Capacity (PC)*:** It is used to give the percentage of cover media.

$$P.C = realusedofcover/lengthofcover * 100 \tag{2}$$

2. ***Hiding Capacity (HC)*:** to give the percentage of the real use of cover (bytes).

$$H.C = secret(bits)/Realused(bytes) \tag{3}$$

3. ***Embedded Ratio equation (ER)***: the embedding ratio determines capacity [24].

$$ER = (covertextletters - embeddedmessageletters)/covertextletters \tag{4}$$

4. ***Ratio(secret/cover)***: It helps to know the ratio between the number of secret bits that need to hide within a number of characters of the cover media that are enough to hide such secret bits.

$$Ratio(secret/cover) = realuseofcover/secretbits. \tag{5}$$

5. RESULTS AND DISCUSSION. In this section, we test the proposed method with six secret messages (3 English,2 Arabic,1 Persian) with different sizes, and we use two Arabic text covers. Table 9 shows the information about secret messages and covers.

Table 9: The information for secret messages and covers)

Secret Message(M) and Covers(C)	Language	Number of characters
(M1)	English	28
(M2)	English	346
(M3)	Persian	1678
(M4)	Arabic	2391
(M5)	English	3212
(M6)	Arabic	3828
(C1)	Arabic	25143
(C2)	Arabic	257895

We tasted three messages (M1, M2, M3) with cover (C1) and three others (M4, M5, M6) with cover (C2) to get the cover percentage, hiding capacity, real use, and the compression result if the message needed, as shown in Fig. 4. and Table 10.

Table 10: The results of Cover percentage, hiding capacity, the real use of characters for five secret messages)

Groups	Secret Message	Cover	The real use of char.	Cover percentage%	Hiding capacity%	Secret ratio	Before compression	After compression
G1	M1	C1	268	1.066	83.582	1.196	28	-
G1	M2	C1	2435	9.685	75.236	1.329	346	229
G1	M3	C1	5872	23.354	77.384	1.292	1678	568
G2	M4	C2	9549	3.703	82.689	1.209	2391	987
G2	M5	C2	15621	6.057	83.273	1.201	3212	1626
G2	M6	C2	13770	5.339	82.382	1.214	3828	1418

As shown from applying six secret messages with different sizes, message (M1) has the best lower value for both (cover percentage and secret ratio) and the best high value for hiding capacity for G1, while the message (M5) has the best value for (hiding capacity and secret ratio). In comparison, (M4) has the best lower value for cover percentage. Also, the proposed method is compared with two other methods [29] and [24]. The first method has been used (Kashida-PS) and (PS-bet words) in [16], as shown in Table 11.

In Table 11, we use the cover text **(لا اله الا الله محمد رسول الله)** with secret bits (16), so the proposed method has the best high value for hiding capacity (94.1%), while the lower value of cover percentage for the proposed method (51.5%) is considered the best because the proposed method is not used a large area from the Arabic cover text, in order not to bring any attention for the secret message existing.

The second method [24] has been used Kashida based moon and sun letters, where the capacity depends on the Embedded Ratio equation (ER). In table 12, the best value for the

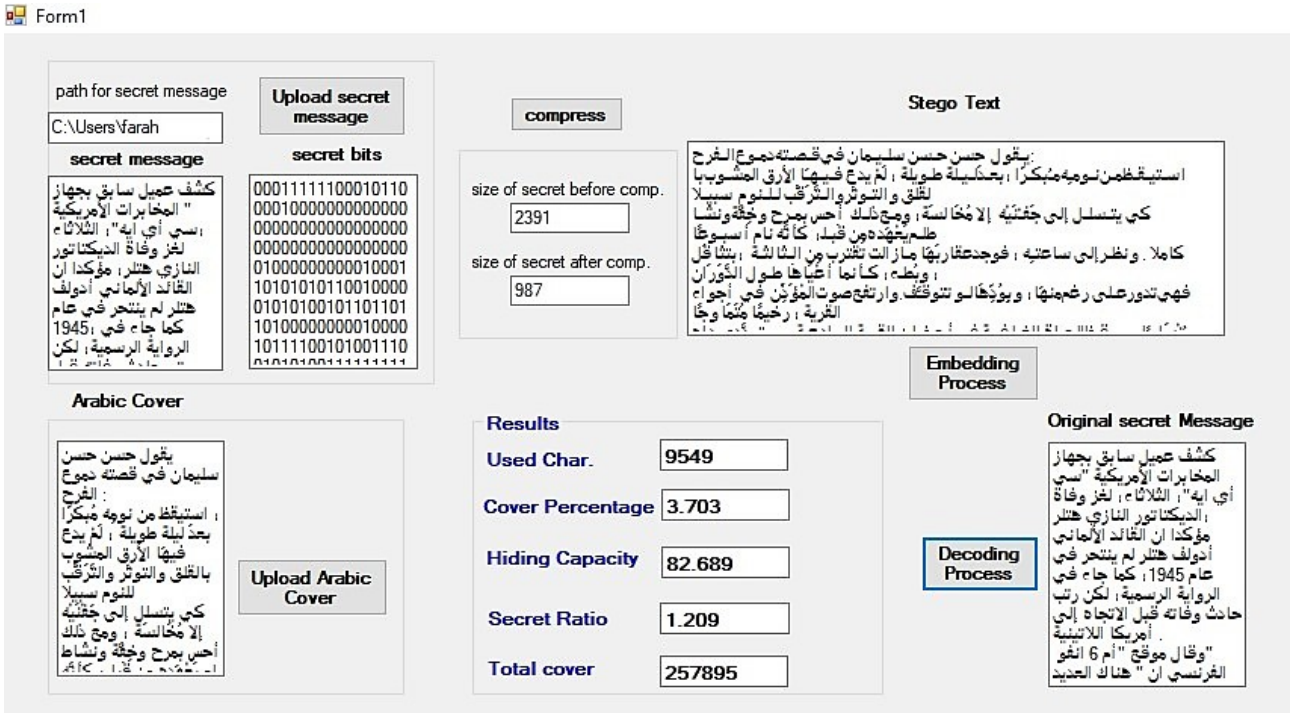

Figure 4: Snapshot of GUI used for applying message 4 with cover 2.

Table 11: Comparison of the proposed method with two methods

Method	Secret message in bits	Cover	The real use of char.	Cover percentage%	Hiding capacity%
Enhancing PS-betWord	16	30	24	80	66.6
Enhancing Kashida_PS	16	30	30	100	53.3
Proposed Method Enhancing	16	30	17	51.5	94.1

embedding ratio is for the proposed method by utilizing a secret message (GOOD) with cover text from newspaper almada-paper.

Table 12: The comparison between method[24] and the proposed method

Methods	Secret word	Average ER
Moon and Sun approach [24]	GOOD	61.16%
Proposed Method	GOOD	96.8%

6. Conclusion. In this work, we have proposed a new text steganography method that uses the features of Arabic language (Noorani and Darkness) letters in order to conceal secret bits by

using the Kashida letter and some Unicode spaces which do not bring attention to the readers. Several conclusions in this work, the main important are:

1. This new method provides high capacity for hiding big secret messages in Arabic text cover.
2. This method used to hide two secret bits instead of one bit.
3. It can be implemented to hide any language.
4. This new method can be run at a low computational requirement, but it will be fast as the computer becomes stronger. In this work, the test is done by a relatively lower level laptop, which has a Core i3 CPU of speed (1.8GH), and it has (4 GB) of RAM.

References

References

[1] M. Douglas, Bailey K., M.Leeney, and K. Curran, An overview of steganography techniques applied to the protection of biometric data, *Multimedia Tools and Applications*, vol. 77, no. 13, pp.17333-17373, 2018..

[2] S. Kaur, S. Bansal, and R. K. Bansal, Steganography and classification of image steganography techniques, *In2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, pp. 870-875, 2014.

[3] V. K. Yadav, and S.Batham, A novel approach of bulk data hiding using text steganography,*Procedia Computer Science*, vol. 57, pp.1401-1410, 2015.

[4] M. Y. Elmahi, and M. H. Sayed, Text steganography using compression and random number generators, *International Journal of Computer Applications Technology and Research*, vol. 6, no. 6 pp. 259-263, 2017.

[5] R. Kumar, and H Singh, Recent Trends in Text Steganography with Experimental Study, *Handbook of Computer Networks and Cyber Security*, Springer, Cham. pp.849-872, 2020.

[6] A. Kumar, and K.Pooja, Steganography-A data hiding technique, *International Journal of Computer Applications*, vol. 9, no. 7, pp.19-23, 2010.

[7] S. Bansod and G. Bhure, Data encryption by image steganography, *Int. J. Inform. Comput. Technol. Int. Res.*, Publ. House. vol. 4, pp.453-458, 2014.

[8] H. M. Alshahrani, Hybrid Arabic Text Steganography, *International Journal of Computer and Information Technology*, vol. 6, no. 6, 329-338, 2017.

[9] M. Shirali-Shahreza and M. H. Shirali-Shahreza, an improved version of Persian/Arabic text steganography using "La" word, *In 2008 6th National Conference on Telecommunication Technologies and 2008 2nd Malaysia Conference on Photonics*, IEEE, pp. 372-376, 2008.

[10] C. Eric, Hiding in plain sight, *Stegnography and the art of Covert Communication*, Wiley, Indianapolis, Indiana, ISBN, 10-p0471444499, 2003.

[11] A. Al-Mohammad, Steganography-based secret and reliable communications: Improving steganographic capacity and imperceptibility, *PhD diss., Brunel University, School of Information Systems*, Computing and Mathematics Theses, 2010.

[12] A. M. Hamdan, and A. Hamarsheh, AH4S: an algorithm of text in text steganography using the structure of the omega network, *Security and Communication Networks*, vol. 9, no. 18, ppt.6004-6016, 2016.

[13] S. M. Al-Nofaie, and A. A. Gutub, utilizing pseudo-spaces to improve Arabic text steganography for multimedia data communications, *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 19-67, 2020.

[14] K. Smaïli, Arabic Language Processing: From Theory to Practice, *7th International Conference, ICALP 2019, Nancy, France, October 16–17, Proceedings*. Springer Nature, 2019.

[15] E. Khan, Using Arabic poetry system for steganography, *Asian Journal of Computer Science and Information Technology*, vol.4, no.6, pp.55-61, 2014.

[16] S. Al-Nofaie, A. A. Gutub, and M. Al-Ghamdi, Enhancing Arabic text steganography for personal usage utilizing pseudo-spaces, *Journal of King Saud University-Computer and Information Sciences*, 2019.

[17] A. A. Gutub, and M. M. Fattani, A novel Arabic text steganography method using letter points and extensions, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, World Academy of Science, Engineering and Technology, vol. 1, no. 3, pp. 502-505, 2007.

[18] A. Odeh, and K. Elleithy, Steganography in Arabic text using zero width and kashidha letters,*International Journal of Computer Science & Information Technology*, vol. 4, no. 3, Jun 2012.

[19] Y. M. Alginahi, M. N. Kabir, and O. Tayan, An enhanced Kashida-based watermarking approach for Arabic text-documents, *In 2013 International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, pp. 301-304, Nov 7 2013.

[20] E. M. Ahmadoh, Utilization of two diacritics for Arabic text steganography to enhance performance, *Lecture Notes on Information Theory*, vol. 3, no. 1, pp.42–47, 2015.

[21] G. A. Memon, Evaluation of Steganography for Urdu/Arabic text, *Journal of Theoretical & Applied Information Technology*, vol.4, no.3, 2008.

[22] S. M. Al-Nofaie, M. M. Fattani, and A. A. Gutub, Capacity Improved Arabic Text Steganography Technique Utilizing 'Kashida' with White-spaces, *In The 3rd international conference on mathematical sciences and computer engineering (ICMSCE2016)*, pp. 38-44, 2016.

[23] A. M. Alhusban, A meliorated Kashida-based approach for Arabic text steganography,*Int. J. Comput. Sci. Inf. Technol. (IJCSIT)*, vol. 9, no. 2, 2017.

[24] A. S. Anes, Text steganography using extension Kashida based on the moon and sun letters concept, *International journal of advanced computer science and applications*, vol. 8, no.8, pp.286-290, 2017.

[25] A. Taha, A. S. Hammad, and M. M. Selim, A high-capacity algorithm for information hiding in Arabic text, *Journal of King Saud University-Computer and Information Sciences*, pp.1-8, 2018.

[26] L. Y. Por, K. Wong, and K. O. Chee, UniSpaCh: A text-based data hiding method using Unicode space characters, *Journal of Systems and Software*. Vol. 85, no. 5, pp.1075-82, 2012.

[27] S. Malalla, and F. R. Shareef, Improving Hiding Security of Arabic Text Steganography by Hybrid AES Cryptography and Text Steganography, *Journal of Engineering Research and Application*, vol. 6, no.6, pp.60-69, 2016.

[28] F.R., Shareef, A novel crypto technique based cipher-text shifting, *Egyptian Informatics Journal*, vol. 21, no. 2, pp.83-90, 2020.