Comparison of Eight Proposed Security Methods using Linguistic Steganography Text Hanaa M. Ahmed and Maisa'a A. A. Khodher
Pages 243 – 251
DOI: http://dx.doi.org/10.21700/ijcis.2016.129

Comparison of Eight Proposed Security Methods using Linguistic Steganography Text

Hanaa M. Ahmed and Maisa'a A. A. Khodher
Department of Computer Science, Iraq University of Technology
[110113, Maisaa_ali2013]@uotechnology.edu.iq

Abstract: *This paper compares eight proposed methods using steganography of Arabic language texts for different search algorithms to consider a secret key. All methods use random numbers to generate the secret key. The objectives are to evaluate each method and to select the best method that provides the best solution suitable to hide the Arabic language texts. Secret sharing is the fourth-best method in security, linear regression is the best method for transparency and capacity of secret message hiding, whereas singular value decomposition is the best method in terms of security and robustness, Huffman code provides secret message compression security and transparency, and steganography in Microsoft Word documents uses the protocol in layer one of single–double quote, which is weak in security. Conversely, the random subtraction of two images method is the best algorithm in terms of security, robustness, and capacity, while Kashida and Single–double quote are the best methods for security, transparency, and robustness, steganography of twice secret messages in layer one is the best method for security and robustness. Of all the aforementioned security methods, secret sharing is the best overall security method.*

Keywords: *Linguistic Steganography, Secret Sharing, Linear Regression, Singular Value, Subtraction Image, Huffman Code, Single–double Quote, Kashida.*

Received*: July 30, 2016* | **Revised***: August 10, 2016* | **Accepted***: August 25, 2016*

1. Introduction

The benefit of comparing different methods including previous methods is to identify the best security method from the collection.

The previous method in Arabic language texts used Kashida in 28 characters, 13 characters with un-point and 15 characters with point. The secret message is in a binary bit string, where bit 1 contains the secret message in two Kashidas, placed in un-pointed characters, and bit 0 contains the secret message in one Kashida, placed in point characters. This method is easy to detect by cyber-attacks.

Unlike the eight methods, Arabic text is used to hide the secret message and convert the Arabic text by Fast Fourier Transform (FFT) and hide the bit secret message in the least-significant bit (LSB) and apply Inverse FFT (IFFT). After these steps, layer one is used to put Kashida in place of the LSB and random Kashida in layer two. This method is difficult to decipher by a cyber-attacker. Eight alternative methods are described next.

"**Formula-Based**: is a new method and uses two levels to hide a secret message, the first level is hiding by embedding and addition, but the second level is hiding by injection. The first level embeds a secret message, one bit in the LSB in the FFT and the addition of one Kashida. Dynamic random linear regression (DRLR) uses NRG to find positions that are hidden within the text. The second level is the injection of one or two random Kashidas within the text" [1].

"**Singular Value Decomposition (SVD)**: is a new method, and a linguistic steganography for Arabic language texts, uses Kashida and FFT based on using a new technique entitled 'Random Singular Value Decomposition Image' as a location to hide a secret message. The proposed approach is an attempt to present a transform linguistic steganography using levels for hiding to improve the implementation of Kashida, and improve the security of the secret message" [2].

"**Compression-Based**: is a new method, depending on the addition of different resultant between the original secret message and another message equal to the new message when the addition is inside other texts. After applying this method, two levels are used to hide a new secret message. This method presents a linguistic steganography for Arabic language documents, using Kashida and FFT on the basis of using a new technique, namely SMC, to obtain a new

secret message using DRLR as the location to hide a secret message" [3].

"**Single and Double Quote**: uses a protocol to present a linguistic steganography of Arabic texts, using single quote mark and double quote mark. The reason for using a new technique entitled RSVD Image is as the location to hide a secret message" [4].

"**Image-Based**: is a new method, using two levels to hide a secret message, the first level is hiding by embedding and addition, but the second level is hiding by injection. The first level embeds a secret message, one bit in the LSB in the FFT and the other is the addition of one Kashida. The subtraction of two random images (STRI) uses NRG to find positions that are hidden within the text. The second level is the injection of one or two random Kashidas within the text" [5].

"**Kashida and Single–double Quote:** is a new method, using two levels to hide a secret message, the first level is hiding by embedding and addition, but the second level is hiding by injection. The first level embeds a secret message one bit in the LSB in the FFT and the other is the addition of one Kashida. DRLR uses NRG to find the positions that are hidden within the text. The second level is the injection of a random single or double quote within the text" [6].

"**Twice Hiding Secret Message**: is a new method, using one level to hide a secret message. This level is hiding by embedding and addition. The single level embeds a secret message twice, one bit in the LSB in the FFT and the other is the addition of one Kashida and Single–double Quote is added in the same secret message. RSVD uses the RNG to find the positions that are hidden within the text" [7].

"**Block-Based**: is a new method, which is a secret sharing in cover and uses two levels to hide a secret message, the first level is hiding by embedding and addition, but the second level is hiding by injection. The first level embeds a secret message one bit in secret sharing in the LSB in the FFT, and the other is the addition of one Kashida. The STRI uses NRG to find the positions for hiding part of the secret sharing within the text. The second level is the injection of one or two random Kashidas within the text" [8].

2. Steganography types

It is common knowledge that many forms of communications exist in the world, such as the telephone, the fax, computer communications, and wireless. Naturally, these communication vehicles use security, of which, there are three principal steganography kinds [9]:

1. Pure steganography,
2. Secret key steganography,
3. Public key steganography.

1- Pure steganography

Pure steganography is a cryptography system that does not request priority exchange of several secure data before transmitting the message; there is no necessity for a data request to begin the connection operation; thus, the securing of the system relies totally on its security, as shown in Figure 1 [9]. Pure steganography can be defined as the quadruple (C, M, D, and E) where

C: the set of possible covers.

M: the set of secret messages with $|C| \geq |M|$.

E: $C \times M \rightarrow C$ the embedding function.

D: $C \rightarrow M$ of the extraction function with the property that

$D(E(c, m)) = m$ for all ϵ M, and c ϵ C [9].

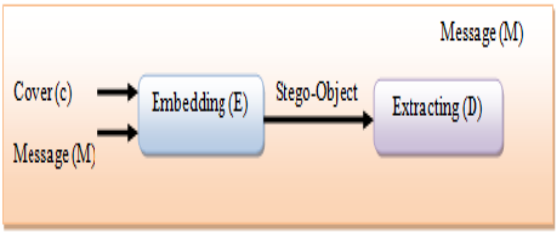

Figure 1. Schematic of pure steganography [9].

2- Secret key steganography

A secret key steganography system is analogous to a symmetric cipher, where the sender chooses a cover and embeds the secure message into a cover using a secret key. If the secret key used in the embedding process is defined to the receiver, he can reverse the operation and extract the secure letter, shown in Figure 2 [9].

Anyone who does not know the secret key cannot obtain evidence of the encoded data. Secret key steganography can be defined as the quintuple (C, M, K, DK, EK) where:

C: the set of possible covers.
M: the set of secret messages.
K: the set of secret keys.
EK: $C \times M \times K \rightarrow C$.

The property that is DK (EK(c, m, k), k) = m for all m ϵ M, c ϵC, and k ϵ K [9].

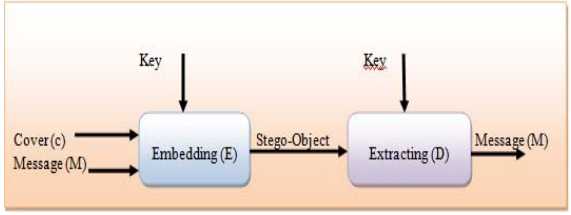

Figure 2: Schematic of secret key steganography [9]

3- Public key steganography

Public key steganography does not rely on the interchange of a secret key. It demands two keys, one is special (secret) and the other is general. The general key is stored in a general database and is used in the embed operation. The secret key is utilized to rebuild the secret message. One path to construct a general key steganography system is to utilize a general key cryptography system. The transmitter and the receiver can interchange general keys of several general key cryptography algorithms before imprison. Public key steganography utilizes the fact that the decoded function in a steganography system can be applied to any cover, whether or not it already contains a secret message [9].

Public key steganography depends on the actuality that encrypted data are randomly suitable to hide in plain sight. The transmitter encrypts the data with the recipient general key to obtain a random-looking message and embeds it in a channel to define the recipient, which may change some of the natural randomness with which every communication process is joined. Presume that both the cryptographic algorithms and the embed functions are generally defined [10].

3. Literature review

Kashida is an Arabic redundant character that is used to justify the text, without affecting the meaning of the words. Researchers suggest using one Kashida as bit zero, and two Kashidas as bit one, or vice versa.

In 2007, Gutub and Fattani introduced a novel Arabic text steganography technique for Arabic script using letter points and Kashida. The technique hides secret information as bits in Arabic letters (cover) by using Kashida and points of letters. The technique considers un-point Arabic letters followed by a Kashida if the secret bit is (0), and point Arabic letters followed by Kashida if the secret bit is (1).

Their technique enhances robustness and security but might have some limitation with a capacity of the cover media if the number of secret bits of the secret information is large. This steganography technique is found to be suitable for other languages having similar scripting to Arabic, for example, Persian and Urdu [11].

In 2009, M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahreza introduced a novel approach to conceal data in Persian (Farsi) and Arabic languages. In the Unicode scale, there are two characters to conceal 'Ya' (ی) and 'Kaf' (ک). The two characters of (ی) and (ک) have themselves shape; however in various codes they are utilized at the start or in the center of words. The major aim in this way is perception translucence. It has stellar perception translucence causes the stego-text that the employee sees to appear alright, like the main text [12].

In 2012, Odeh and Elleithy introduced a new steganography algorithm for Arabic script by applying zero width and Kashida letters. When using the Kashida embedded letters with other letters, the extension letters do not change the meaning of the word. In addition, the zero-width characters (Ctl+shift+1) do not change when hidden in 1-bit using an extension of Kashida letters and when hidden in 2-bits using zero-width letters. This algorithm, ZKS, is better than that of Hassan and … because it applies various concepts similar to parallel communication, and alteration, to very difficult stegoanalysis operations [13].

4. Kashida-based method

The Arabic expansion character "Kashida" is used to extend the space between joint letters. The Kashida refers to a character representing this extension (-) which increases the length of a line of the script. It cannot be added to the start or end of words. It is used to adjust the script without any change to the content of the text [14].

5. Overview of steganography

This overview involves the utilization of steganography, more commonly referred to as data hiding, the method of concealing a secure letter in often overtly ready information. With steganography, the transmission of a letter should conceal it in a host file. The host file, or public letter, is the file that anyone can view. When one uses steganography, this often conceals the correct intent for communicating in a more usual communication scenario [15].

6. Text steganography

This paper addresses text steganography, which uses text as the medium in which to conceal data. It is another type of steganography; that is largely because of the relative lack of excessive data in a text file compared with an image or a voice file. The frame of text documents is analogous to how we look, in other kinds of documents as in an image, the frame of the document is different from what it appears. So, in such documents, it can be concealed by interchanges in the structure of the document without making a change in the concerned output. In contrast to other media, such as images, sounds, and video clips, using text documents has been common for centuries. After the invention of the printing machine, documents have contained only text. It uses text best over other media because text consumes less memory, can communicate more data, and is printed at low cost [16].

Text steganography is broadly classified into two kinds: linguistic steganography, which is further split into semantic and syntactic; and format-based steganography, which is further split into the following, line-shift encode, word-shift encode, open-space encode, and feature encode [16].

7. Method of hiding:

Breaking down steganography techniques, founded on the conceal approach, is the preferred method. There are three basic ways to conceal data: injection, substitution, and generation [16]:

1. Injection finds areas in a file that will be ignored and puts your covert message in those areas.
2. Substitution finds insignificant information in the host file and replaces it with your covert data.
3. Generation creates a new overt file based on the information that is contained in the covert message [15].

8. Technical steganography:

Many different steganography methods have been proposed through the last few years; the extreme ones can be visible as substitution systems. Such methods attempt to replace excessive parts of a signal with secret letters; their main disadvantage is the relative weakness against cover modifications. Recently, the evolution of new strong watermarking methods led to advances in the construction of robust and secure steganography systems. There are several methods in classifying steganography systems. One could classify them according to the kind of covers used for secret communication. A classification according to the cover adjustment applied in the embedding operation is another possibility [17], as shown in Figure 3 [18].

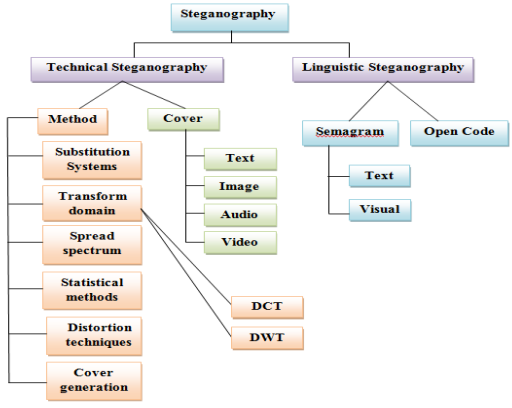

Figure 3. Workflow of technical steganography [17].

9. Fundamental contributions

This paper describes a collection of eight methods that use Kashida in the Arabic language that can hide bits and remove any intruder suspicions. In addition, it offers a comparison of the eight methods in Table 1, including the number of levels, type of embedding, and results.

10. Comparing previous work

A comparison of previous work is shown in Table 2, which includes the collection of methods from 2007 to 2013 used in this research.

11. Results and discussion

Figure 4 indicates the collection of eight methods used in the general framework system.

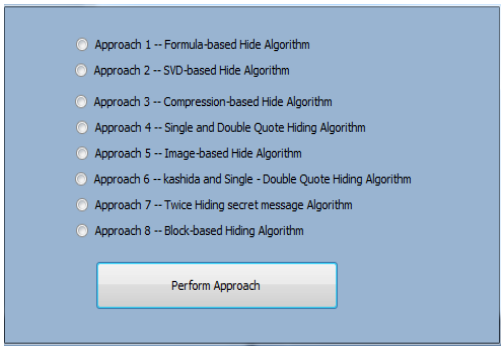

Figure 4. A collection of eight methods in the system.

1. Formula-Based:

This method shows the test for implementation of the formula-based method for security, robustness, transparency, and capacity using DRLR, as depicted in the load process in Figure 5.

Figure 5. Formula-based method: embedding process and retrieved message.

2. SVD:

This method shows the test for implementation of the SVD method for security, robustness, transparency, and capacity using RSVD, as depicted in the load process in Figure 6.

Figure 6. SVD method: embedding process and retrieved message.

3. Compression-Based:

This method shows the test for implementation of the compression-based method for security, robustness, transparency, and capacity using DRLR, as depicted in the load process in Figure 7.

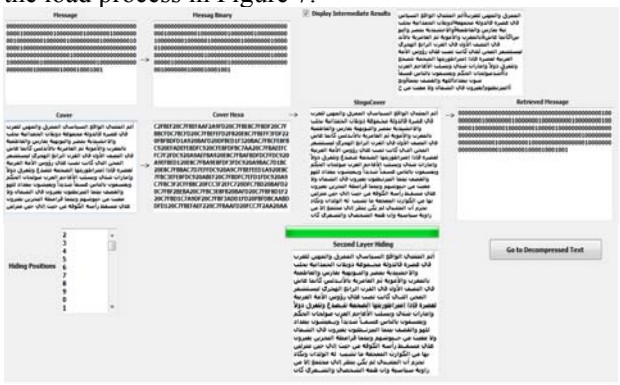

Figure 7. Compression-based method: embedding process and retrieved message.

The message after the decompression load process is shown in Figure 8.

Figure 8. Decompression of the message.

4. Single and Double Quote:

This method shows the test for implementation of the single and double quote method for security, robustness, transparency, and capacity using RSVD, as depicted in the load process in Figure 9.

Figure 9. Single and double quote method: embedding process and retrieved message.

5. Image-Based:

This method shows the test for implementation of the image-based method for security, robustness, transparency, and capacity using STRI, as depicted in the load process in Figure 10.

Figure 10. Image-based method: embedding process and retrieved message.

6. Kashida and Single–double Quote:

This method shows the test for implementation of the Kashida and Single–double quote method for security, robustness, transparency, and capacity using DRLR, as depicted in the load process in Figure 11.

Figure 11. Kashida and Single–double quote method: embedding process and retrieved message.

7. Twice Hiding Secret Message:

This method shows the test for implementation of the twice hiding secret message method for security, robustness, transparency, and capacity using RSVD twice, as depicted in the load process in Figure 12 (a, b).

Figure 12-a. Twice hiding secret message method: embedding process of Kashida one layer and retrieved message.

Figure 12-b. Twice hiding secret message method: embedding process of single–double quote one layer and retrieved message.

8. Block-Based:

This method shows the test for implementation of the block-based method for security, robustness, transparency, and capacity using STRI, as depicted in the load process in Figure 13.

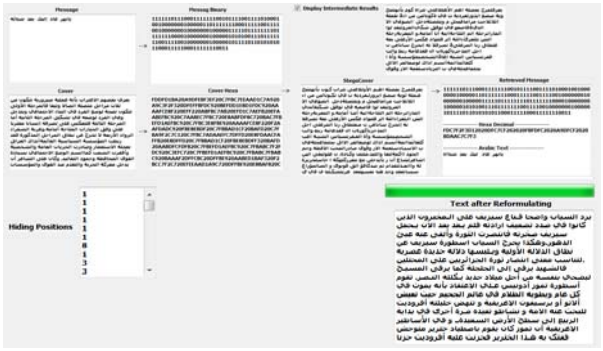

Figure13. Block-based method: embedding process and retrieved message.

Conclusion

In this research, after testing eight methods, powerful security is found in the compression-based and block-based methods in this paper. In previous research in Arabic text using Kashida in the protocol of 28 characters in the Arabic language, using 13 characters in un-pointed hidden of 1-bit in secret message and put two Kashidas, and uses 15 characters in point to hide the 0-bit in secret message put one Kashida. This paper uses two layers to hide the secret message; the first layer uses FFT and hides the bit secret message, and applies IFFT in LSB and places one Kashida in place of the LSB in cover Arabic text. In layer two, it places a random Kashida, without sensitive attacks during transmission across a network. The single and double quote method is weak in security because it uses single–double quote without FFT but uses protocol. The formula-based, SVD, image-based, Kashida and Single–double quote, and twice hiding secret message methods are powerful in transparency and robustness. The block-based method of security is 95%, and the compression-based method of security is 96%.

References

[1] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Arabic Language Script Steganography Based On Dynamic Random Linear Regression," in Journal of College of Education, No. 1, 2016.

[2] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Arabic Language Text Steganography Based On Singular Value Decomposition (SVD)", Engineering & Technology Journal, University of Technology, (in press).

[3] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Arabic Language Documents Steganography Based On Huffman Code Using DRLR As (RNG)," in Journal of Al-Mansour, Accepted on 16 February 2016.

[4] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Arabic Language Text Steganography Based On Microsoft Word Documents", Al-Yarmouk Journal, first issue, 2015.

[5] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Steganography Based on Arabic Language Texts by Kashida Using STRI As RNG,", Iraq Journal of Science, The University of Baghdad, Accepted on 19 April 2016.

[6] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Steganography Techniques In Arabic Language Texts Utilizing Single–double Quote Using RNG", Journal of Al-Mansour, College of Basic Education, Accepted on 19 April 2016.

[7] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Steganography Arabic Text Based on Natural Language Process Documents," in Journal of College of Education, University of Wasit (in press).

[8] Hanaa M. Ahmed, Maisa'a A. A. Khodher, "Steganography Arabic Language Scripts Based on Secret Share Using RNG," Journal of Imam Ja'afar Al-Sadiq University (in press).

[9] Zaidoon Kh. Al-Ani, A. A. Zaidan, B. B. Zaidan and Hamdan O. Alanazi, "Overview: Main Fundamentals for Steganography," in Journal of Computing, Volume 2, Issue 3, March 2010.

[10] H. A. Jalab, A. A Zaidan and B. B Zaidan, "New Design for Information Hiding within Steganography Using Distortion Techniques," International Journal of Engineering and Technology (IJET), Vol. 2, No. 1, 2010.

[11] Adnan A.-A. Gutub and Manal M. Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions," in International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol. 1, No. 3, 2007.

[12] H. Shirali-Shahreza and Mohammad Shirali-Shahreza, "Arabic/Persian Text Steganography Utilizing Similar Letters with Different Codes," The Arabian Journal for Science and Engineering, Vol. 35, No. 1B, December 9, 2009.

[13] A. Odeh and K. Elleithy, "Steganography In Arabic Text Using Zero Width And Kashidha Letters," in International Journal of Computer Science & Information Technology (IJCSIT), Vol. 4, No. 3, June 2012.

[14] Reem A. Alotaibi, and Lamiaa A. Elrefaeil, "Arabic Text Watermarking: A Review," Int. Journal of Artificial Intelligence and Applications (IJAIA), Vol. 6, No. 4, July 2015.

[15] E. Cole, Ronald D. Krutz, "Hiding in Plain Sight: Steganography and the Art of Covert Communication," Wiley Publishing, IN. 2003.

[16] H. Singh, P. Kumar Singh and K. Saroha, "A Survey on Text Based Steganography", Proceedings of the 3rd National Conference, INDIA Com-2009 Computing for Nation Development, February 26–27, 2009.

[17] Stefan Katzenbeisser, Fabien A. P. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking," Artech House Computer Security Series. Available at: http://www.ifi.unizh.ch/~oppliger/serieseditor.html.

[18] Navneet Kaur and Sunny Behal, "A Survey on Various Types of Steganography and Analysis of Hiding Techniques," in International Journal of Engineering Trends and Technology (IJETT)–Vol. 11 No. 8 – May 2014.

Asst. Prof. Dr. Hanaa Mohsin Ahmed
Assistant Prof. Dr. Hanaa M. A. Salman obtained her M.Sc. and Ph.D. from the University of Technology Iraq in 2002 and 2006, respectively. Currently, she is a lecturer in Computer Science and a member of the Scientific Committee and Promotion Committee in the Department of Computer Science. Dr. Hanaa has more than 23 years of experience and has supervised graduate students. Her research interests include cryptography, computer security, biometrics, image processing, and computer graphics.

Maisa'a Abid Ali Khodher
Ph.D. Student. Maisa'a A. A. Khodher obtained her M.Sc. degree in Computer Science in 2005, and her M.Sc. in Image Processing in 2005 from the University of Technology in Iraq. Currently, she is a lecturer in Computer Science. Maisa'a has more than 20 years of experience and she has supervised B.Sc. final year projects. Her research interests include cryptography, image processing, databases, data security, and linguistic steganography.

Table 1. Collection of eight proposed methods.

No. of Paper	Name of paper	Name of researcher	Tools	No. of levels	Type of embedding	Results
1	Arabic Language Script Steganography Based On Dynamic Random Linear Regression	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	DRLR	Level one is Add One Kashida Level two is Random Kashida	Addition Injection Substitution	Need a suitable cover with secret message. Best for use in high security message
2	Arabic Language Text Steganography Based On Singular Value Decomposition (SVD)	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	RSVD	Level one is Add One Kashida Level two is Random Kashida	Addition Injection Substitution	Need a large cover With small secret message and high security
3	Arabic Language Documents Steganography Based On Huffman Code Using DRLR As (RNG)	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	DRLR	Level one is Add One Kashida Level two is Random Kashida	Compression Addition Injection Substitution	Need a suitable cover with secret message. Best for use in high security
4	Arabic Language Text Steganography Based On Microsoft Word Documents	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	RSVD	Level one is using Protocol Single–double Quote	Addition Injection Substitution	Need a suitable cover With large secret message weak in security
5	Steganography Based On Arabic Language Texts By Kashida Using STRI As RNG	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	STRI	Level one is Add One Kashida Level two is Random Kashida	Subtract Addition Injection Substitution	Need a large cover With small secret message Best for use in high security
6	Steganography Techniques In Arabic Language texts Utilizing Single–double Quote Using RNG	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Students Maisa'a Abid Ali K.	DRLR	Level one is Add One Kashida Level two is Single–double Quote	Addition Injection Substitution	Need a suitable cover with secret message Best for use in high security
7	Steganography Arabic Text Based on Natural Language Process Documents	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	RSVD Twice	Level one is Add One Kashida and Single–double Quote	Addition Injection Substitution	Need a large cover With small secret message Best for use in high security
8	Steganography Arabic Language Scripts Based On Secret Share using RNG	Asst. Prof. Dr. Hanaa M. Ahmed Ph.D. Student Maisa'a Abid Ali K.	STRI	Level one isSecret Share Add One Kashida Level two is Random Kashida	Addition Injection Substitution	Need a large cover With small secret message Best for use in high security

Table 2. Collection of previous methods.

No. of Paper	Name of researcher	Year	Type of embedding	Techniques
1	A. Gutub and M. Fattani	2007	Using Kashida and points of letters	Un-pointed Arabic letters Kashida if the secret bit is (0), and point Arabic letters followed by Kashida if the secret bit is (1).
2	A. H. Fahd et al.	2009	Using Kashida Using three scenarios	The approach discusses the maximum number of Kashida letters added to the Arabic cover word, evaluated the number of hidden bits embedded in the carrier file, and the results are compared with diacritics and Kashida methods.
3	M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahreza	2009	Conceal script of two characters of 'Ya' (ى) and 'Kaf' (ک) used	Center of words. The major aim of this approach is perception translucence. It has stellar perception translucence, which causes the stegano-text which the user sees alright like the main text.
4	Adnan A.-A. Gutub et al.	2010	Using extension character (Kashida)	One Kashida if the secret bit is (0) and two Kashidas if the secret bit is (1).
5	A. Ali and F. Moayad	2010	Using Kashida with Huffman code	Considers absence of Kashida if the secret bit is (0) and one Kashida if the secret bit is (1) after any connected letters.
6	A. Odeh and K. Elleithy	2012	Using the Kashida embedded letters with other letters	Uses the zero-width characters (Ctl+shift+1) not change otherwise. When hiding in 1-bit they use extension Kashida letters and when hiding in 2-bits they use zero-width letters.
7	A. Oden et al.	2013	Using variation in Kashida, selects one of four scenarios randomly	Considers un-pointed Arabic letters followed by a Kashida if the secret bit is (0), and point Arabic letters followed by Kashida if the secret bit is (1) as in the first scenario, and vice versa as second scenario. The third scenario is adding Kashida after Arabic letters if the secret bit is (1) and (0) otherwise, as in the fourth scenario.