Journal of Computer Sciences

Original Research Paper

An Efficient Approach to Hide Compressed Voice Data in Arabic Text using Kashida and “La”

Sabeeka M. Al-Oun and Jehad Q. Odeh Alnihoud

Department of Compute Science, Al al-Bayt University, Mafraq, Jordan

Article history
Received: 18-12-2016
Revised: 13-02-2017
Accepted: 24-03-2017

Corresponding Author:
Jehad Q. Odeh Alnihoud
Department of Compute
Science, Al al-Bayt University,
Mafraq, Jordan
Email: jehad@aabu.edu.jo

Abstract: In this research, we propose a new approach to increase the capacity and enhance the reliability of hiding voice data in Arabic text. Using Kashida to hide bits in Arabic text is one of the most promising approaches in steganography. Unfortunately, ignoring the original Kashida in the cover text may affect the results significantly and produce inaccurate results in the extraction process. In this study, we propose tremendous improvements to the Kashida method by considering original Kashida(-) in the cover text, error-detection using Cyclic Redundancy Check (CRC) and hiding bit using the “La” word. Moreover, hiding voice files within Arabic text is considered. The proposed approach is compared with the most related approaches in terms of capacity, security and reliability. Not only are the findings of the paper promising, they also overcome the limitations of other approaches.

Keywords: Steganography, Information Security, Cycle Redundancy Check, Voice Data, Loss-Less Compression

Introduction

Steganography is the science of hiding information via embedding hidden content in a remarkable cover media. Three major aspects affecting steganography are security, capacity and robustness. “Capacity refers to the amount of data bits that can be hidden in the cover medium. Security relates to the ability of intruders to figure the hidden information easily. Robustness is concerned about the resist possibility to modify or destroy the unseen data” (Chen and Wornell, 2001).

The current steganography science uses the opportunity of hiding into digital multimedia files such as audio, video, image, text and IP datagram (Chintan and Patel, 2012).

This research specifically deals with text steganography. In which, text is used as a medium to hide information. Krista Bennett’s definition of text steganography remains the most common one to differentiate it from the more specific term “linguistic steganography”. She states that text steganography involves changing words within a text, changing the formatting of an existing text and generating random character sequences (Bennett, 2004).

Azawi and Fadhil (2010) proposed techniques to hide information through implanting extension characters (Kashidas) at suitable word positions to hold secret bit one and leaving an empty position to hold secret bit zero. They used Huffman Compression Algorithm to convert the embedding message into a compressed binary form. This technique showed more capacity and security when it was compared with some existing Arabic text steganography methods and the results were promising. Unfortunately, they ignored original Kashida(-) at Arabic text, since it has a negative effect on the extracting process because the system deals with the original Kashida(-) as one bit hidden at a text. Furthermore, they did not apply any error detection technique in order to verify the reliability of data extraction process.

In this study, we propose tremendous improvements to the Kashida method proposed in (Azawi and Fadhil, 2010). The current study takes original Kashida(-) in the cover text into consideration. Error-detection using Cyclic Redundancy Check (CRC) is also considered. Furthermore, hiding bit using the “La” word as well as hiding voice file within Arabic text are taken into account.

Previous Studies

Shirali-Shahreza and Shirali-Shahreza (2006) proposed a special security method based on the features of characters of Arabic and Persian letters. Their approach depended on the points inherited in the Arabic and Persian letters, which are quite similar. Arabic and English languages both have points in their letters. Small "i" and small "j" are the only two pointed characters in English language, while Arabic has fifteen pointed letters out of its twenty eight alphabet letters. This bulky number of points in Arabic letters makes the points in any given Arabic text notable and practical for steganography and information security.

Gutub and Fattani (2007) presented a new Arabic text steganography approach to hide secret information bits in the letters aiding from their inherited points. They considered the presence of the points in the letters and the redundant Arabic extension character. The pointed letters with extension is used to hold the secret bit 'one', while the un-pointed letters with extension to hold 'zero'.

Ahmed *et al.* (2008) introduced a new approach for steganography in Arabic and Urdu texts. They considered the existence of Diacritics - or Harakat - (i.e., "Fatha", "Kasra" and "Damma"). They used reverse "Fatha", to hide information in the cover text.

Gutub *et al.* (2008) utilized the advantages of diacritics in Arabic to implement text steganography. Diacritic, or Harakat in Arabic, are used to represent vowel sounds and can be found in many formal and religious documents. The proposed approach used eight different diacritical symbols in Arabic to hide binary bits in the original cover media. The embedded data are then extracted by reading the diacritics from the document and translating them back to binary. Nevertheless, according to this proposed approach, it is easy to notice the changes in the cover text and the reusing of the same text cover significantly reduces the capacity every time the text is used.

Roslan *et al.* (2011) highlighted a new approach based on sharp edges method. They utilized the even and odd hiding module. The capacity issue is resolved and as compared with other approaches, the proposed method provides high capacity of hiding secret bits.

Odeh and Elleithy (2012) proposed new Arabic text steganography algorithm. They approach the connected letter through adding kashida and zero width character. The algorithm uses concepts as parallel connection and randomization. The result was promising as compared with the related algorithms.

Mersal *et al.* (2014) presents a new Arabic text steganography algorithm centered on sharp-edges to be used in smart phones. They implement the new algorithm using Java language. As compared to other method the proposed algorithm showed high ability to increase the hiding capacity.

Alginahi *et al.* (2014) uses a predefined key to encode the original text. A kashida is placed after a set of characters considering whether the letter is assumed to have high frequency of recurrence or not..

Al-Nofaie *et al.* (2016) used whitespaces as well as kashidas to hide the secret bits. If the secret bit to be embedded is one, they add a kashida between two suitable letters and no kashida is added in case of embedding zero bit. Moreover, they deploy the whitespaces to hide secret bits. If the end of the word is reached, two consecutive whitespaces are added to hide one bit and the normal space between words is used in the case of hiding zero bit.one whit but if the secret bit to be embedded is zero.

The Proposed Approach

The proposed architecture of hiding voice data in an Arabic text using kashida and "La" with loss-less compression and cyclic redundancy check system (HVDATKLCRCS) depends on two phases; embedding data phase and extracting original data phase.

Embedding Data Phase

Preprocessing

Preprocessing stage includes voice file compression, conversion of a compressed file to a binary file and CRC calculation. In this stage, the noise in the voice file will be eliminated and then the file is compressed using Loss-Less Compression technique. Loss-less compression techniques depend on the interval of a sampled signal. The signal is sampled after a given interval Ts (sampling period), in order to be a Pulse Code Modulation (PCM) signal. Thus, the Loss-Less compression technique presented in (Kabir *et al*., 2010) is deployed in this research.

Throughout the process of data transmutation, errors may occur, so the communication method includes the use of cyclic check redundancy codes for additional error detection and correction competency. This method is for error detection and correction in a received message that comprises n message bits with m cyclic redundancy check bits attached. It is specified if at least one bit error has occurred in the n message bits and m CRC bits of the received message. Then K bits with a lowest quality metric are selected from the n and m bits. The bit error is amended based upon likely bit error patterns and the selected K bits (John, 2009).

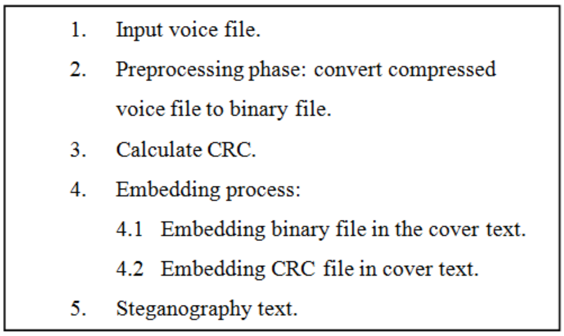
Embedding Binary File and CRC in Arabic Text

In this stage, the positions of the original Kashidas in the text are determined and stored in an array. The proposed approach identifies whether the character accepts a Kashida to hide bit 1 or leaves it without kashida insertion for bit 0. The same steps are utilized to hide the CRC for a later use.

Table 1 shows Arabic letters that accept an extra Kashida based on the position of that letter in a word.

Table 1. Arabic letters that accept Kashida

Position	Arabic letter	Position	Arabic letter
1,2	ب	1,2	ع
1,2	ت	1,2	غ
1,2	ث	1,2	ف
1,2	ج	1,2	ق
1,2	ح	1,2	ك
1,2	خ	1,2	ل
1,2	س	1,2	م
1,2	ش	1,2	ن
1,2	ص	1,2	ه
1,2	ظ	1,2	ض
1,2	ط	1,2	ي



1. Input voice file.
2. Preprocessing phase: convert compressed voice file to binary file.
3. Calculate CRC.
4. Embedding process:
   4.1 Embedding binary file in the cover text.
   4.2 Embedding CRC file in cover text.
5. Steganography text.

Fig. 1. Proposed system architecture for embedding process

Position1 indicates that this letter is at the first position of the word and can accept a Kashida, while position2 indicates that the letter at any position of a word, except at the end of the word and can accept a Kashida. The result of embedding process is displayed as a (TFF) text format.

Figure 1 shows the proposed system architecture of the embedding process.

Algorithm 1 shows the detailed steps of embedding a secret message and CRC in the cover text.

Algorithm 1: Embedding Secret Message and CRC
Input: Compressed binary file (CBF), Cover Text (CT), CRC
Output: Stego Text

```
1.  Let N← length.CBF ;i← 1.
2.  If size.CT< (size.CBF + size.CRC)
         then select new CT
3.    Else
4.      Identify original Kashida(-) in CT and save it
          in Array (A).
5.      if i> N then
6.          Insert two Kashida(-) to indicate end of
              secret message insertion.
7.      Key ← Position of (6)
8.      Let i← key +1
9.      Repeat steps 5-21 with CRC.
10.     Exit.
11. Else If i > CT.length then
12.    "Message the size of text is not enough" and exit
13.   Else X ←getCharacter
14.      If X accept extra Kashida(-) && X has no
            original Kashida(-)
15.        If (Bit == 1) then
16.           insert Kashiad(-)
17.           i← i+1
18.           go to (5)
19.        Else
20.           i←i +1
21.           go to (5)
```

Table 2 shows an example of hiding (0011111111000011) within Arabic text. Notice that at “الناس” the “ن” accepted a kashida, but since the word has an original kashida, it has simply been ignored. Moreover, the “لا” in the “الاسم”accepted a kashida.

Extraction Process

The main objectives of the extraction process are to obtain the original voice file and CRC. The following Fig. 2 shows the extracting process of the original voice file (secret message) and CRC, where key 1 represents the position of the last embedding file and key 2 represents the position of the last embedding CRC.

Post Processing

After the extracting process of binary file and CRC, we must determine whether there is an error or more and try to correct it by using Cyclic Redundancy Check method. Detection of errors occurs through comparing the extracted CRC with the received CRC. Figure 3 shows the flowchart of error detection.

System Design

The proposed system is implemented using C#.net and Rich Text Format (RTF) with two forms: Encoded and Decoded. “RTF is a proprietary document file format with published specification developed by Microsoft Corporation since 1987 for Microsoft products and for cross-platform document interchange” (John, 2009; Chen and Wornell, 2001; Provos and Honeyman, 2003). Majority of word processors are capable to read and write some versions of RTF (Chandramouli and Memon, 2001). There are numerous revisions of RTF requirement and portability of files, depending on what type of RTF is being used (Chen and Wornell, 2001; Doërr and Dugelay, 2003).

Experimental Result and Discussion

The proposed approach is tested with different cover text file sizes in terms of capacity and reliability.

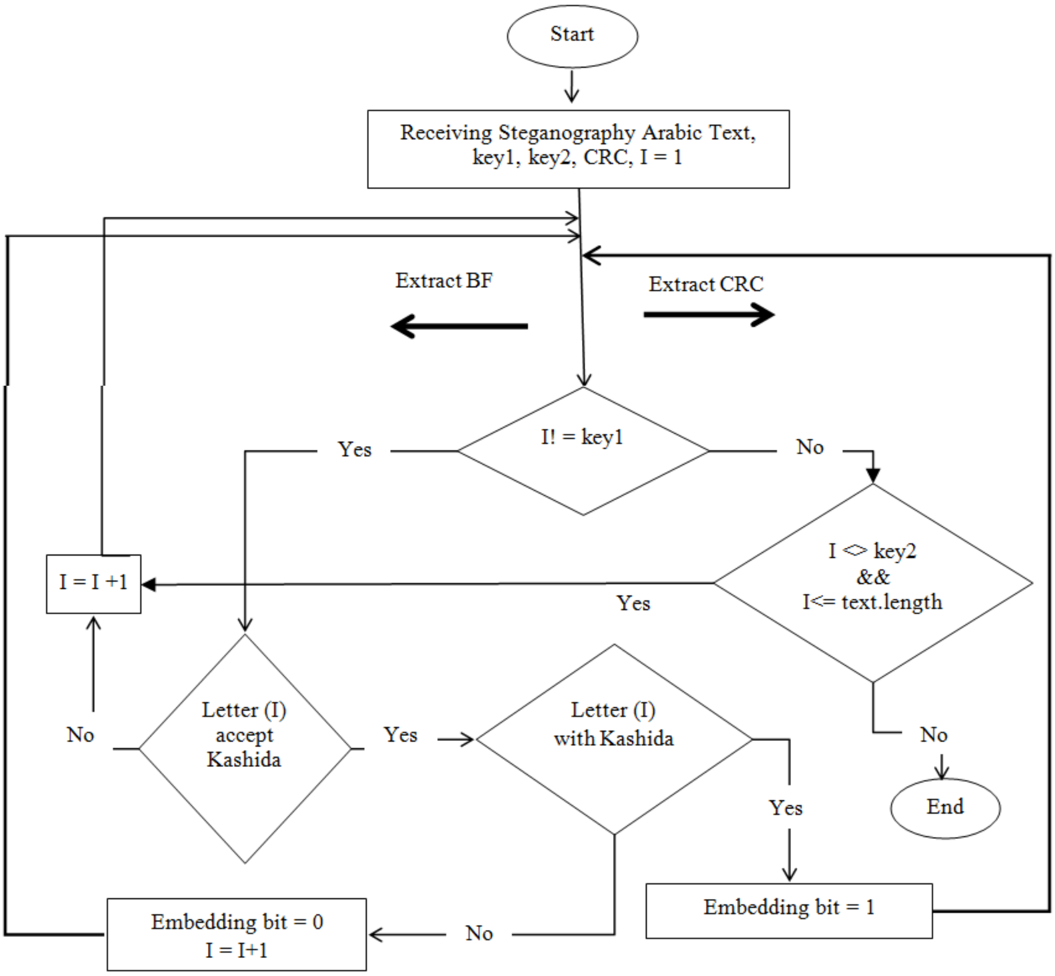

Start
Receiving Steganography Arabic Text, key1, key2, CRC, I = 1
Extract BF
Extract CRC
I! = key1
Yes
No
I <> key2 && I<= text.length
Yes
No
End
I = I +1
Letter (I) accept Kashida
No
Yes
Letter (I) with Kashida
Yes
No
Embedding bit = 0 I = I+1
Embedding bit = 1

Fig. 2. Flowchart of extracting original voice file and CRC

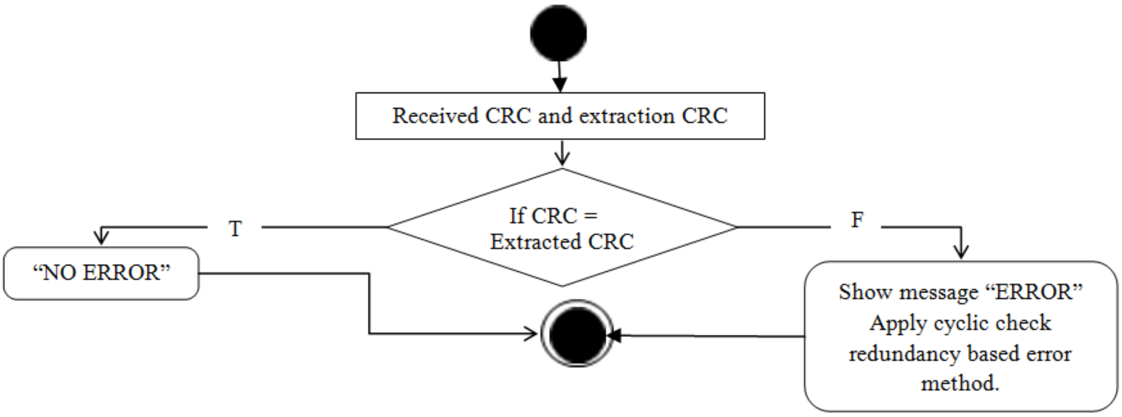

Received CRC and extraction CRC
If CRC = Extracted CRC
T
F
"NO ERROR"
Show message "ERROR" Apply cyclic check redundancy based error method.

Fig. 3. Flowchart of error detection and correction

Table 2. Example of using Arabic words with an extension kashida to hide

Before	After
إن هذا الاسم الجديد الذي تسامع الناس به منذ أربعة عشر قرناً عنوان لحقيقة قديمة بدأت مع الخليقة وسايرت حياة البشر	إن هـذا الـاسم الجديد الـذي تـسـامـع النـاس بـه مـنـذ أربـعـة عشر قرناً عـنوان لحقيقة قديمة بدأت مع الخليقة وسايرت حياة البشر

Table 3. Kashida extensions with Huffman code (Azawi and Fadhil, 2010)

Cover size (byte)	No. of hidden bits	Capacity ratio (%)	Average capacity (%)
4050	969	2.991	3.02
15563	3754	3.015	
24957	6040	3.025	
29283	7156	3.054	

Table 4. Dots based approach (Shirali-Shahreza and Shirali-Shahreza, 2006)

Cover size (byte)	Capacity ratio (%)	Average capacity (%)
131619.2	1.172	1.37
6983.68	1.467	
6799.68	1.275	
3604.48	1.529	

Table 5. Capacity ratio and average capacity of the proposed approach

Cover size (byte)	No. of hidden bits	Capacity ratio (%)	Average capacity (%)
4500	1200	3.33	3.30
16300	3805	2.92	
25650	7320	3.57	
30255	8215	3.39	

Capacity

The capacity ratios computed by dividing the amount of hidden bytes over the size of the cover text in bytes (Shirali-Shahreza and Shirali-Shahreza, 2006). The proposed approach is compared with (Azawi and Fadhil, 2010; Shirali-Shahreza and Shirali-Shahreza, 2006). Table 3 shows the capacity ratio and average capacity of the proposed method in (Azawi and Fadhil, 2010), while Table 4 shows the capacity ratio and average capacity of the proposed method in (Shirali-Shahreza and Shirali-Shahreza, 2006).

Table 5 shows the capacity ratio and the average capacity of our proposed approach.

The results show that the capacity ratio of the proposed approach slightly increases as compared to (Azawi and Fadhil, 2010) and it outperforms the capacity ratio achieved by (Shirali-Shahreza and Shirali-Shahreza, 2006).

Reliability

The aim of reliability theory is to estimate errors in measurement and to suggest ways of enhancing tests so that errors are diminished. The proposed system is highly reliable since it considered the original kashida in the cover text and ignored it, while most of the foregoing studies in this field do not pay attention to the original Kashida, which may affect the correctness of extraction process intensely. Throughout an extensive testing, we approach the 100% reliability, because of eliminating the original Kashida from being considered as part of the secret message in the extracting phase and deploying CRC to detect and correct errors.

Time

In this study, the time to extract the secret message is reduced dramatically, as compared with the related approaches in Arabic text steganography since the proposed approach depends on the size of the secret message and CRC rather than the size of cover text as in (Azawi and Fadhil, 2010; Gutub and Fattani, 2007). In this study, extraction process depends on the size of voice file and it is terminated at key1 (the last position to insert Kashida depends on the last bit at voice file, which is performed by inserting two consecutive Kashidas). This takes at worst case O(M) where M is the size of voice file and the extraction process of CRC starts from the position of key1+1 and it is terminated at key2(the last position to insert a Kashida depends on the last bit of CRC, which is performed by inserting two consecutive Kashidas). This takes at worst case O(K), where K is the size of CRC. The overall time complexity of searching process is O(M+K). It is obvious that the size of the secret message in addition to the size of the CRC is still less than the size of the cover text and using key1 and key 2 to indicate the end of embedded bits reduces the time of the extraction process.

Conclusion

A system to hide voice data in an Arabic text using Kashida and "La" with Loss-Less Compression and Cycle Redundancy Check is developed. HVDATKLCS has proposed a structure of hiding data in Arabic text system.

The system is designed and developed using four different stages, preprocessing, hiding (embedding) binary file and CRC into Arabic text, extracting data from steganography text and error detection and correction. This research presents a novel steganography scheme useful for Arabic language electronic writing. We use Arabic letters and "La" which accept extra Kashidas, insert an extra Kashida after the letter to hold secret information bit 'one' and leave the letter without insertion to hold secret bit 'zero'.

The proposed approach was tested and compared with the most related approaches in Arabic text steganography and consequently, it appears that the presented approach enhances the capacity ratio, ensures high level of security and accelerates the process of extraction.

As a future work to enhance the finding in this study, we are planning to deploy a new technique in order to allow the kashida holding more than one secret bit and integrate that with Huffman code as compression technique to increase the hiding capacity.

Acknowledgement

The Authors would like to thank Al al-Bayt University for supporting this work.

Funding Information

This work is unfunded.

Author's Contributions

Jehad Q. Odeh Alnihoud: Algorithm design and analysis, running experiments, writing the paper.

Sabeeka M. Al-Oun: Running experiments, data analysis, writing the paper.

Ethics

This article is original and the material is not published previously. The corresponding Author ensure that his colleague have read and approved the manuscript and no ethical issues involved.

References

Ahmed, M.J., K. Khowaja and H. Kazi, 2008. Evaluation of steganography for Urdu/Arabic text. J. Theoretical Applied Inform. Technol., 14: 232-237.

Alginahi, Y., M. Kabir and O. Tayan, 2014. An enhanced Kashida-based watermarking approach for increased protection in Arabic text-documents based on frequency recurrence of character. Int. J. Comput. Electr. Eng., 6: 381-392. DOI: 10.17706/ijcee.2014.v6.857

Al-Nofaie, S.M., M.F. Manal and A.A. Gutub, 2016. Merging two steganography techniques adjusted to improve Arabic text data security. J. Comput. Sci. Comput. Math., 6: 59-65. DOI: 10. 20967/jcscm.2016.03.004

Azawi, A.F. and M.A. Fadhil, 2010. Arabic text steganography using kashida extension with Huffman code. J. Applied Sci., 10: 436-439. DOI: 10.3923/jas.2010.436.439

Bennett, K., 2004. Linguistic steganography: Survey, analysis and robustness concerns for hiding information in text. CERIAS Technology Report, Purdue University, West Lafayette.

Chandramouli, R. and N. Memon, 2001. Analysis of LSB based image steganography techniques. Proceedings of the International Conference on Image Processing, Oct. 7-10, IEEE Xplore Press, pp: 1019-1022. DOI: 10.1109/ICIP.2001.958299

Chen, B. and G.W. Wornell, 2001. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. Inform. Theory, 47: 1423-1443. DOI: 10.1023/A:1008107127819

Chintan, M. and Y. Patel, 2012. Steganography and steganalysis: Different approaches for information hiding. Int. J. Eng. Res. Technol.

Doërr, G. and J.L. Dugelay, 2003. A guide tour of video watermarking. Signal Process.: Image Commun., 18: 263-282. DOI: 10.1016/S0923-5965(02)00144-3

Gutub, A. and M.M. Fattani, 2007. A novel Arabic text steganography method using letter points and extensions. Int. J. Comput. Electr. Automat. Control Inform. Eng.

Gutub, A., Y. Elarian, S. Awaideh and A. Alvi, 2008. Arabic text steganography using multiple diacritics. Proceedings of 5th IEEE International Workshop on Signal Processing and its Applications, (SPA' 08), pp: 1-5.

Shirali-Shahreza, H.M. and M. Shirali-Shahreza, 2006. A new approach to Persian/Arabic text steganography. Proceedings of 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, Jul. 10-12, IEEE Xplore Xpress, pp: 310-315. DOI: 10.1109/ICIS-COMSAR.2006.10

John, W.N., 2009. Cycle Redundancy Check (CRC) based error correction method and device. United States, Patent No. US 7,577,899 B2.

Kabir, H.M., B. Syed, I. Azam, R. Ahmed and R. Islam *et al*., 2010. A theory of loss-less compression of high quality speech signals with comparison. Proceedings of 4th UKSim European Symposium on Computer Modeling and Simulation, Nov. 17-19, IEEE Xplore Press, pp: 136-141. DOI: 10.1109/EMS.2010.33

Mersal, S., S. Alhazmi, R. Alamoudi and N. Almuzaini, 2014. Arabic Text steganography in Smartphone. Int. J. Comput. Inform. Technol., 3: 441-445.

Odeh, A. and K. Elleithy, 2012. Steganography in Arabic text using zero width and kashidha letters. Int. J. Comput. Sci. Inform. Technol., 4: 1-11. DOI: 10.5121/ijcsit.2012.4301

Provos, N. and P. Honeyman, 2003. Hide and seek: An introduction to steganography. IEEE Security Privacy, 99: 32-44. DOI: 10.1109/MSECP.2003.1203220

Roslan, N.A., R. Mahmod and N.I. Udzir, 2011. Sharp-Edges method in Arabic text steganography. J. Theoretical Applied Inform. Technol., 33: 32-41.