Arabic Diacritics Based Steganography

Mohammed A. Aabed, Sameh M. Awaideh, Abdul-Rahman M. Elshafei and Adnan A. Gutub

Computer Engineering Department
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
{maabed, sameho, shafei, gutub}@kfupm.edu.sa

Abstract

New steganography methods are being proposed to embed secret information into text cover media in order to search for new possibilities employing languages other than English. This paper utilizes the advantages of diacritics in Arabic to implement text steganography. Diacritics - or Harakat - in Arabic are used to represent vowel sounds and can be found in many formal and religious documents. The proposed approach uses 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.

Index Terms - Arabic Text, Data Hiding, Diacritic Marks, Text Steganography, Text Watermarking

1. Introduction

The evolution experienced nowadays in computer technologies is employing a rapid growth in the number of users and diversity of applications. One of the main concerns in this field is the ability to privately exchange information and hide the data of interest throughout the transmission process. A variety of solutions have been proposed for this problem in the literature, some are cryptographic, steganographic or even pure coding. As a way to hide the exchange of data, steganography has gained a wide interest among researchers and security specialists.

Steganography is the science of forming hidden messages such that the intended recipient is the only party aware of the existence of the message [1]. This is usually done by embedding the private data in a cover media without destroying the meaningfulness of this media. The fact that people are not aware of the presence of the message distinguishes between steganography and other forms of information security. For instance, in cryptographic systems the existence of the information itself is not disguised although the comprehension of the message is not possible [2].

In steganography for digital systems, the cover media used to hide the message can be text, image, video or audio files [3]. However, using text media for this purpose is considered the hardest among the other kinds. Unlike video and voice media, text data does not have much needless information within the essential data [4,5].

Different methods and approaches have been attempted to implement text steganography [4]. Most of these known approaches hide data by making minimal modifications to the painting of the characters or spaces. The authors in [4] name feature coding, open spaces, word shifting and line shifting as examples for the these approaches. It should be noted that these approaches are usually hard to implement

and are font or computer dependent. Other approaches use syntactic and semantic characteristics in the language to embed data. Additional implemented methods for text steganography using Arabic language are further discussed in the following sections.

This paper proposes a new steganography scheme to hide binary data in Arabic text media. The rest of this paper is organized as follows; section 2 has the background information about Standard Arabic language and related work. In section 3, the new approach is introduced in details. The analysis and discussion are reported in section 4. Finally, the conclusion statement is in section 5.

2. Background

The Arabic language, written from right to left, is based on an alphabetical system that uses 28 basic letters. Unlike English, Arabic does not differentiate between upper and lower case or between written and printed letters. Moreover, Arabic language uses different symbols as diacritical marks, or simply diacritics which are also known as Harakat. The main eight diacritic symbols are shown in Table 1. Other diacritic marks also exist but are outside of the scope of this paper.

Table 1. The eight main Arabic diacritics

Fatha	ـَ	Kasrah	ـِ
Dhammah	ـُ	Sukkon	ـْ
Shaddah	ـّ	Tanween Fath	ـً
Tanween Kasr	ـٍ	Tanween Dham	ـٌ

Just like most of the diacritics based written languages, the main purpose of using diacritics in Arabic languages is to alter the pronunciation of a phoneme or to distinguish between words of similar spelling. Nonetheless, the use of diacritics in the text is optional in written Standard Arabic. In this work we utilize this characteristic to define a steganographic scheme for hiding binary data within Arabic text, as it will be shown in details in section 3.

Arabic language is poorly used in the steganography field. To the best of the authors' knowledge, there is only one algorithm in the literature that uses Arabic text as a cover media to hide binary data [4]. The authors in [4] proposed a feature coding method for hiding binary values into Persian/Arabic phrases. This method takes advantage of the frequent presence of point in these kinds of phrases. Simply, the approach uses vertical repositioning of these points to represent 0's and 1's. The paper reports some of the advantages and disadvantages of the proposed algorithm and discusses difficulties facing text steganography as well.

Another proposed - yet unpublished - approach exploits the existence of the redundant Arabic extension character, i.e. Kashida [6]. The author proposes to use pointed letters in Arabic with a Kashida to hold the secret bit 'one' and the un-pointed letters with a Kashida to hold 'zero'. This approach as well as other approaches will be used for analysis and comparison of the results. Before embarking upon our proposed approach, it should be noted that this paper is not restricted only for the Arabic language. Most of the Semitic languages use diacritics in one form or another, and the proposed approach can be slightly modified to suit other languages

requirements.

3. Proposed Approach

The use of diacritics in written Standard Arabic language is optional. This means that novel Arabic readers can read a text without diacritics correctly by applying the Arabic language grammar to that text. In this work, we use this property to introduce a novel yet simple steganography scheme. As it will be shown in details in section 4, our analysis indicates that in Standard Arabic the frequency of one diacritic, namely Fatha, is almost equal to the occurrence of the other seven diacritics. Thus, the paradigm we introduce in this work assigns a one bit value, namely 1 value, to the diacritic Fatha and the remaining seven diacritics will represent a value of one bit of 0 value.

To implement this approach, we use a fully diacritized Arabic text as our cover media. A computer program reads the first bit of the embedded data and then compares it with the first diacritic in the cover media. If, for example, the first bit to be embedded was a one, and the first diacritic was a Fatha, the diacritic is kept on the cover media and an index for both of the embedded text and the cover media is incremented. If, however, the first diacritic was not a Fatha then it is removed from the cover media and the index for the cover media is incremented to explore the next diacritic. This approach is repeated until a Fatha is found. The same method is used to implement zeros with the only difference that a zero will search for the other seven diacritics instead of the Fatha. The overall process is repeated for as long as there are bits remaining to be hidden.

The following figures illustrate our proposed technique. Fig. 1 shows a Standard Arabic text full of diacritics. We generate pseudo-random sequences to embed into our cover media, the sequence used in this example is: E7 - 30 - E9 - 1C - A4 - FC - B8 - B9 - AF - 1F - 0B - D9 - 22 represented in Hexadecimal format. Finally, Fig. 2 shows how the sequence hidden text can be encoded in the cover media using the presented approach.

حَدَّثَنَا سُفْيَانُ عَنْ يَحْيَى عَنْ مُحَمَّدِ بْنِ إِبْرَاهِيمَ التَّيْمِيِّ عَنْ عَلْقَمَةَ بْنِ وَقَّاصٍ قَالَ سَمِعْتُ عُمَرَ رَضِيَ اللَّهُ عَنْهُ يَقُولُ سَمِعْتُ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ إِنَّمَا الْأَعْمَالُ بِالنِّيَّةِ وَلِكُلِّ امْرِئٍ مَا نَوَى فَمَنْ كَانَتْ هِجْرَتُهُ إِلَى اللَّهِ عَزَّ وَجَلَّ فَهِجْرَتُهُ إِلَى مَا هَاجَرَ إِلَيْهِ وَمَنْ كَانَتْ هِجْرَتُهُ لِدُنْيَا يُصِيبُهَا أَوْ امْرَأَةٍ يَنْكِحُهَا فَهِجْرَتُهُ إِلَى مَا هَاجَرَ إِلَيْهِ

Figure 1. Example of a Standard Arabic text with full diacritics placement.

حَدَّثَنا سُفْيَان عَن يَحْيى عنْ محَمَدِ بْنِ إبرَاهيمَ التَيْمي عَنْ علْقَمة بْنِ وقَّاص قَالَ سمِعْت عمَر رضيَ اللهُ عَنْهُ يَقول سَمعت رَسولَ الله صَلّى اللَّهُ عَليه وَسلّم يقُولُ إنَما الْأعمَال بالنيَة وَلِكُل امرئ مَا نَوى فمنْ كَانتْ هجرَته إلى الله عَزّ وجلّ فهجرَته إلَى مَا هَاجَر إلَيْهِ ومنْ كَانتْ هجرَته لدنيَا يصيبهَا أَوْ امرَأَةٍ ينْكحهَا فهجْرَتُهُ إلَى ما هاجر إليه

Figure 2. Example of using Standard Arabic diacritics to encode a pseudo-random sequence.

It should be noted that the same cover media can be reused more than once if needed. However, unless a method is used to reinsert removed diacritics, capacity will decrease drastically every time we embed a new message into the text. A research group in IBM [7] proposed to use maximum entropy to restore missing Arabic diacritics. Another approach used HMMs to restore vowels [8]. Diacritic restoration can also be done manually if necessary, and in any case, a new cover media that is already diacritized can be used instead.

4. Discussion and Analysis

A) Modeling:

This section summarizes the tests ran to evaluate the proposed steganography method, along with some statistics that can enhance this approach. The algorithm was modeled and simulated in C++. As a testbed, or test environment, we used one of the largest online books available in Arabic literature, namely Musnad Al-Emam Ahmed [10]. Because of its large size, only a portion, i.e. half, of the book was used as a testing environment. The testbed contains 7305490 characters and 1037265 words. Initial results showed very promising expectations as for capacity and ambiguity. Diacritics represent almost 50% of the total file size if the cover media is fully diacritized. Table 2 shows some statistical studies done on the document. The proposed testbed is formatted under HTML, which means that using Arabic encoding each character is only 1 byte. The following table represents the hexadecimal representation for each diacritic:

Table 2. Hexadecimal representation for each diacritic (Under HTML Arabic encoding)

Diacritics	Value (HEX)
Fatha	0xF3
Kasra	0xF6
Dhammah	0xF5
Sukkon	0xFA
Shaddah	0xF8
Tanween Fath	0xF0
Tanween Kasr	0xF2
Tanween Dham	0xF1

Having eight different diacritic symbols, we initially assumed that each diacritic symbol (which is a 1 Byte) can carry 3 bits of hidden information at once. This means that we can assume that the usable capacity can be almost 16.7% (50% diacritics multiplied by 3/8 for each Byte). Unfortunately, this is not the case since advanced studies showed huge statistical discrepancies between the different diacritic symbols in the document. It was observed that usually Fatha stands for almost 50% of the presence of all diacritic marks! Whereas, for example, Tanween Fath only makes up for 0.56% of the whole set! The previous results indicate that giving each diacritic a set of 3 bits will decrease capacity by a huge rate. For example, if we map 3 to a Fatha and if we map a 1 to a Tanween Fath, we will encounter more than a 100 Fatha's, which we need then to discard, before encountering a one Tanween Fath.

This prompted for a new technique where we assign a one bit value, i.e. 1, to the diacritic Fatha and another bit value, i.e. 0, to any of the other diacritic symbols (i.e. Dhammah, Kasrah …etc). Mapping each diacritic, which is one byte in size, to a bit of hidden message (1/8) works well, but will reduce the capacity ratio. Theoretically, we can calculate that the best case scenario cannot surpass 6.25% (50% * 1/8). In fact, the average capacity is almost 3.27% per bit which is half the theoretical rate. On the other hand, this technique is much more ambiguous for general users than the previous technique.

As explained before, an encoding/decoding program was developed under C++ that can accepts a HTML Arabic diactrized page for cover media, a file in any format to hide (.txt, .wav, .jpg … etc.), and produces a HTML page that has the previous cover media but containing the hidden file embedded within. The output file

can be processed for the decoding program to retrieve the hidden file and save it in its previous and proper format.

B) Comparison and Analysis:

A question arises, is 3% capacity considered a low ratio in text steganography? Comparing this technique with some other proposed methodologies in the literature, the previous ratio (3%) exceeds all the other proposed techniques. The authors in [4] proposed a new approach in Arabic and Persian steganography that used word shifting which produced on average 1.37% bit per bit. The work in [9] proposed a technique for using typesetting tools and provided a capacity of 0.98% per character or almost 0.12% bit per bit.

Furthermore, the work in [6] was also implemented for comparison reasons. The technique is as follows; for every pointed character we insert a kashida if we need to embed a 1, or don't insert a kashida if we need to embed a 0. The program was tested under various formats and results are reported in Table 4. It produced an average capacity of 1.22%. This concludes that our approach was higher in capacity than any other approach even though technically the work in [6] can be modified to increase capacity but with much greater loss of ambiguity. Our approach seemed to be less ambiguous in the bare eye than the work in [6] even as stated before both can look odd to the experienced Arabic reader. Tables 3 and 4 provide more insight to our approach compared to the approach in [6].

Some properties of the proposed approach and other remarks can be summarized as following:

- If a fully diacritized Arabic text is used as cover media, the scheme provides the highest capacity.
- The proposed method is robust. It can withstand printing, OCR techniques, font changing, retyping as long as the medium can display Arabic.
- It is fast and does not require large computational power.
- Simple and can be implemented manually if needed.
- Might raise suspicions since it is uncommon nowadays to send diacritized text, unless the cover media used is religious or political documents for example.
- After embedding the hidden media, some of the words can contain a lot of diacritics on them, while other redundant diacritics can exist depending on the arrangement of diacritics in that word. This can reduce the ambiguity of the technique.

Table 3. Statistical reports for the cover media used as a testbed.

Diacritics	Frequency	Percentage
Fatha	1679820	49.8%
Kasra	472101	14%
Dhammah	367224	10.89%
Sukkon	459566	13.62%
Shaddah	300906	8.92%
Tanween Fath	18752	0.56%
Tanween Kasr	50820	1.5%
Tanween Dham	24096	0.71%

Table 4. Diacritics Technique

File Type	File Size (Bytes)	Cover Size (Bytes)	Capacity (%)
.txt	10,356	318,632	3.250%
.wav	43,468	1,334,865	3.256%
.jpg	23,796	717,135	3.318%
.cpp	10,356	318,216	3.254%
		Average	3.27%

Table 5. Kashida Technique

File Type	File Size (Bytes)	Cover Size (Bytes)	Capacity (%)
.txt	4439	365181	1.215%
.html	4439	378589	1.172%
.cpp	10127	799577	1.266%
.gif	188	15112	1.244%
		Average	1.22%

5. Conclusion

The paper proposed a novel algorithm for text steganography in Arabic language and other similar Semitic languages. This was achieved by taking advantage of the existence of diacritic symbols in Arabic documents. Diacritics were used to represent binary bits depending on the hidden message. It was shown that in a fully diactritized text, capacity can be very high compared to other methods in the literature. The method was implemented and demonstrated satisfactory results.

This method can be used also in printed documents, and can be used with different fonts as well as different machines as long as it supports Arabic language. Different variations of this technique can be implemented and some have been discussed in section 4.

The presented approach is easy to implement and does not require complex software. A trained human can perform the scheme manually if necessary. In addition, automated tools can be used to reinsert diacritics in the cover media for it to be used again. Further work can be carried out to improve the ambiguity of this method.

References

[1] Donovan Artz, “Digital Steganography: Hiding Data within Data," *IEEE Internet Computing*, vol. 05, no. 3, pp. 75-80, May/Jun, 2001.

[2] Neil F. Johnson and Sushil Jajodia, “Exploring Steganography: Seeing the Unseen," *IEEE Computer*, February 1998, vol. 31, no. 2, pp.26-34.

[3] J.C. Judge, “Steganography: Past, Present, Future”, SANS white paper, November 30, 2001, http://www.sans.org/rr/papers/index.php?id=552, last visited: March 30, 2006.

[4] Shirali-Shahreza, M.H. and Shirali-Shahreza, M., “A New Approach to Persian/Arabic Text Steganography,” in *th IEEE/ACIS International Conference on Computer and Information Science, 2006. ICIS-COMSAR 2006*, Location, 10-12 July 2006, pp. 310–315.

[5] J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O’Gorman, “Electronic Marking and Identification Techniques to Discourage Document Copying”, *IEEE Journal on*

Selected Areas in Communications, vol. 13, Issue. 8, October 1995, pp. 1495-1504.
[6] Adnan Abdul-Aziz Gutub, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", Unpublished manuscript.
[7] I. Zitouni, J. S. Sorensen, and R. Sarikaya, "Maximum entropy based restoration of Arabic diacritics." *In Proceedings of the 21st international Conference on Computational Linguistics and the 44th Annual Meeting of the ACL* Sydney, Australia, pp. 577-584, 2006.
[8] Y. Gal, "An HMM Approach to Vowel Restoration in Arabic and Hebrew." *In ACL-02 Workshop on Computational Approaches to Semitic Languages* 2002.
[9] Chen Chao, Wang Shuozhong, and Zhang Xinpeng, "Information Hiding in Text Using Typesetting Tools with Stego-Encoding", *Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06),*2006.
[10] "Musnad Al-Emam Ahmed", August 27, 2006, http://islamport.com/b/3/alhadeeth/motoon/%df%ca%c8%20%c7%e1%e3%ca%e6%e4/%e3%d3%e4%cf%20%c3%cd%e3%cf%20%c8%e4%20%cd%e4%c8%e1/, last visited: April 18, 2007