ARABIC/PERSIAN TEXT STEGANOGRAPHY UTILIZING SIMILAR LETTERS WITH DIFFERENT CODES

M. Hassan Shirali-Shahreza*1
Virtual Education Graduate College
Amirkabir University of Technology
Tehran, Iran

and Mohammad Shirali-Shahreza2
Computer Science Department
Sharif University of Technology
Tehran, Iran

*Corresponding Authors:
E-mails: hshirali@aut.ac.ir1; shirali@cs.sharif.edu2

Paper Received June 30, 2007; Paper Rewritten April 7, 2009; Paper Accepted December 9, 2009
ABSTRACT

Sending information secretly and communicating covertly have been of great interest for ages. For this purpose, text documents have been widely used and, consequently, various methods for hiding information in texts (text steganography) have been developed so far. This paper deals with a new method to hide information in Persian (Farsi) and Arabic texts.

In the Unicode Standard, there are two different characters for “Ya” «ي» and “Kaf” «ك». The two characters of «ي» and «ك» have the same shape but different codes if they are used at the beginning or in the middle of words. As a result, we can hide information in texts, considering the specified data to be hidden and using one of these two characters (for the positions in which «ي» and «ك» are at the beginning or in the middle of the words). The above-mentioned method has been implemented by Java Programming Language.

Key words: information hiding, Persian/Arabic text, text steganography, Unicode
ARABIC/PERSIAN TEXT STEGANOGRAPHY UTILIZING SIMILAR LETTERS WITH DIFFERENT CODES

1. INTRODUCTION

With the development of the computer and the expansion of its use in different areas of life and work, the issue of security of information has gained special significance. One of the concerns of information security is the concept of hidden exchange of information. For this purpose, various methods including cryptography, steganography, coding, and so on have been used. Steganography is one method which has attracted more attention in recent years [1].

There are three important parameters in designing steganography methods: perceptual transparency, robustness, and hiding capacity. These requirements are known as "the magic triangle" and are contradictory [2].

One of the main goals of steganography is perceptual transparency, which means that nobody will notice the existence of hidden information. Therefore, in steganography methods, if someone knows that there is a piece of hidden information, usually he/she can extract it easily.

This is the major distinction between steganography and other methods of hidden exchange of information. For example, in cryptography, people become aware of the existence of information by observing coded information, although they will be unable to comprehend the information. However, in steganography, nobody will notice the existence of information in the resources [3].

Most steganography methods make use of pictures [4], video clips [5], and music and sounds [6]. Text is the most difficult media for steganography; because in contrast to other media such as pictures or sounds, the structure of text documents is identical with what we observe and there is not any redundant information in a text file to use to hide data [7].

Of course today, the security of information hiding has been considerably improved by combining steganography with other methods mentioned above. In addition to hidden exchange of information, steganography is used in other areas, such as copyright protection, preventing e-document forging, and other applications [8].

Contrary to other media such as pictures, sounds, and video clips, using text documents has been common for centuries. Even after the invention of the printing press, most books and documents contained only text. This practice continues today. Using text is preferred over other media because text takes less memory, communicates more information, and is less costly to print. As the use of text and hidden communication goes back to antiquity, we have long witnessed the use of steganography, for example, in order to prevent disclosure of government documents by the press. Margaret Thatcher, former British Prime Minister, used to place a certain number of white spaces in documents related to each cabinet minister so that she could identify the owner of the document [8].

Today, computer systems facilitate hiding information in texts. The range of techniques for hiding information in text has also expanded. Different methods are used for hiding information in text, which will be dealt with in Section 2.

The present paper offers a new method for hiding information in Persian and Arabic texts by using Unicode. It can do this due to the existence of two different codes for “Ya” ﯾ and “Kaf” ﯷ characters in Persian and Arabic languages.

In Unicode, there are two different codes for “Ya”, which have the same shape at the beginning and middle of words. The “Kaf” character also has the same feature. We use this feature in our proposed method to hide information in Persian and Arabic texts.

In Section 3, we will explain our algorithm. Section 4 discusses the implementation of this project. Section 5 surveys the advantages of the method. The final section concludes the paper.

2. RELATED WORKS

In this section, at first the usual methods for text steganography are reviewed. Then some text steganography methods that are especially designed for Arabic and Persian texts are surveyed.

2.1. Usual Text Steganography Methods

In general, text steganography methods can be divided into two groups. The first group changes the format or presence of the text. The second group tries to change the text according to the meaning of the text. The methods which are based on the meaning are limited. Some examples of these methods are as follows:

In the semantic method [9], the synonyms of words are used for certain words, thereby hiding information in the text. Another method for hiding information is the use of abbreviations [7]. In another method, by placing some
punctuation marks such as a full stop (.) or comma (,) in proper places, the concerned information can be hidden in a text file [9]. This method requires identification of proper places for putting punctuation marks.

In these methods, very little information can be hidden in the text. For example, only a few bits can be hidden in a file of several kilobytes. A major advantage of the above methods is the protection of information in case of retyping or using OCR (Optical Character Recognition) programs (contrary to methods which change the format or presence of the text). However, these methods may alter the meaning of the text.

As a final example, there is a method in which some specific characters from certain words are selected as hiding places for information [10]. In the simplest form, for example, the first words of each paragraph are selected in such a way that by placing the first characters of these words side by side, the hidden information is extracted. This method has been done by classic poets of Iran. This method requires strong mental power and takes a lot of time. It also requires special text and not all types of texts can be used in this method.

There are a lot of methods belonging to the second group which change the format of the text. These methods usually have a large capacity for hiding information. Some examples of these methods are as follows:

In the line shifting method [11], the lines of the text are vertically shifted to some degree (for example, each line is shifted 1/300 inch up or down) and information is hidden by creating a unique shape of the text. This method is suitable for printed texts.

In the word shifting method [11], by shifting words horizontally and by changing distances between words, information is hidden in the text. This method is acceptable for texts where the distance between words is varied. This method is not easy to identify, because change of distance between words to fill a line is quite common.

In both of these methods, if somebody was aware of the algorithm of distances, he/she could compare a text according to the algorithm and extract the hidden information by using the difference. The text image could also be closely studied to identify the changed distances. Although this method is very time consuming, there is a high probability of finding information hidden in the text. Retyping of the text or using OCR programs destroys the hidden information.

In the open spaces method [7], hiding information is done through adding extra white spaces in the text. These white spaces can be placed at the end of each line, at the end of each paragraph, or between the words. This method can be implemented on any arbitrary text and does not raise the attention of the reader.

In the feature coding method [12], some of the features of the text are altered. For example, the end part of some characters, such as h, d, b, and so on, are elongated or shortened a little, thereby hiding information in the text. In this method, a large volume of information can be hidden in the text without making the reader aware of its existence.

2.2. Arabic/Persian Text Steganography Methods

To the best knowledge of the authors, there are only four Arabic/Persian text steganography methods reported in the literature. The first two methods were developed by the authors of this article.

In the first method [13], data is hidden in Persian and Arabic texts by using a special characteristic of these languages, that is, the existence of so many dots in Persian and Arabic characters. By vertical displacement of the dots (see Figure 1), we hide information in the texts. This method doesn’t attract attention and can hide a large volume of information in the text.

The second method [14] created by the authors uses the special form of the “La” word for hiding the data. This word is created by connecting “Lam” and “Alef” characters. For hiding bit 0, we use the normal form of the word “La” (“لـ”), by inserting an Arabic extension character between “Lam” and “Alef” characters. However, for hiding bit 1, we use the special form of the word “La” (“ﻻ”), which has a unique code in the Unicode Standard. Its code is FEFB in Unicode hex notation. This method is not limited to electronic documents (e-documents); it can also be used on printed documents.

The third method [15] uses the pointed letters with extension (Kashida in Arabic) to hold secret bit ‘one’ and the un-pointed letters with extension to hold secret bit ‘zero’. Note that letter extension does not have any effect on the writing content. It has a standard character hexadecimal code of 0640 in the Unicode system.
The extension is added before (or after) the pointed letters which can be extended with extension character to hide bit 1, and added before (or after) the un-pointed letters to hide bit 0. Figure 2 shows an example of this method.

<table>
<thead>
<tr>
<th>Watermarking bits</th>
<th>110010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover-text</td>
<td>من حسن اسلام المرء تركه مالا يغبنيه</td>
</tr>
<tr>
<td>Output text</td>
<td>من حسن اسلام المرء تركه مالا يغبنيه</td>
</tr>
<tr>
<td></td>
<td>1 1 0 0 1 0</td>
</tr>
</tbody>
</table>

Figure 2. Steganography example: adding extensions after letters [15]

In the fourth method [16], a diacritic Arabic text is used for hidden exchange of information. There are eight diacritics in Arabic text. The most frequent diacritic in Arabic text is “Fatha” and the probability of its occurrence is equal to the probability of occurrence of the other seven diacritics [16].

In this method, at first the cover text is assumed to be a fully diacritical text. To hide a bit “1”, a “Fatha” is kept, and to hide a bit “0”, a non “Fatha” diacritic is kept and other diacritics are removed. Therefore, in the stego text, each “Fatha” represents “1” and each non “Fatha” diacritic represents a “0”.

The main advantage of this method is its high capacity. However, the main disadvantage of this method is that it attracts the attention of the reader. This method needs a fully diacritical text, but most Arabic texts have no diacritic.

3. OUR PROPOSED METHOD

In this paper, we present a new method for text steganography in Persian and Arabic texts.

Before explaining the method, we mention the main characteristics of these two languages. Then we explain the Unicode Standard briefly and, at last, we explain our suggested method in full details.

3.1. The Characteristics of Persian and Arabic Script [17]

The Arabic alphabet has 28 letters. Persian has all the letters of Arabic and four more letters of (پ،چ،ژ،گ). In these two languages, a letter can have four different shapes. The shape of each letter is determined by the position of that letter in a word. For example, the letter «ع» is written as «ﻋـ» at the beginning of a word, like «ـﻌـ» in the middle, as «ـﻊ» at the end, and as «ع» in the separate (isolated) position. We use this characteristic of Arabic and Persian languages in our method and will explain it in this section.

In Persian and Arabic, the letters are connected to each other in writing, while in English the letters are written separately. In English, the letters are written from left to right and in some languages the letters are written from top to bottom, but in Arabic and Persian the letters are written from right to left. In Arabic and Persian texts, dots are very important; 14 of 28 Arabic letters (17 of 32 Persian letters) have one or more dots, while in English only two small letters "i" and "j" have dots.

3.2. The Unicode Standard [18]

The Unicode Standard is the international character-encoding standard used for presenting the texts for computer processing. This standard is compatible to the second version of ISO/IEC 10646-1:2000 and has the same characters and codes as ISO/IEC 10646.

Unicode enables us to encode all the characters used in writing the languages of the world. This standard uses the 16-bit encoding, which provides enough space for 65000 characters; that is to say, it is possible to specify and define 65000 characters in different moulds such as numbers, letters, symbols, and a great number of current characters in all different languages of the world. Moreover, because of the wideness of the space dedicated to the characters, this standard also includes most of the symbols necessary for high-quality typesetting. The languages whose writing systems can be supported by this standard are Latin (covering most of the European languages), Cyrillic (Russian and Serbian), Greek, Arabic (including Arabic, Persian, Urdu, Kurdish), Hebrew, Indian, Armenian, Assyrian, Chinese, Katakana, Hiragana (Japanese), and Hangeul (Korean). Moreover, there are a lot of mathematical and technical symbols, punctuation marks, arrows, and miscellaneous marks in this standard.

In Unicode, separate characters have been allocated for Persian letters which have semantic or shape significance different from Arabic letters despite unification of the codes of the common characters. That is to say, separate places have been allocated to Persian special letters (پ،چ،ژ،گ) and two other Persian letters (ک،ی) which are different from Arabic corresponding letters concerning exhibition. We have used these characteristics of Persian and
Arabic languages in our method and will explain it in the remaining part of the paper. In addition, all the current diacritics are present in this standard. The shapes of the digits have been taken into consideration because of their differences in Persian and Arabic scripts.

On the other hand, codes of the punctuation marks such as the comma, period, and space have the same Unicode code in both Latin and Arabic. Moreover, this standard has detailed and careful explanations about the implementation methods including the letters-connection method, and the exhibition of the right-to-left and bi-direction texts. Therefore, the programmers do not have to refer to the local guide [18].

3.3. Our Text Steganography Method

As described in Subsection 3.1, Arabic and Persian are only different in four letters (ک،پ،چ،ژ). However, they are also partially different in the shapes of some characters which we mentioned as an example in Subsection 3.2 concerning «ﯼ» and «ﮎ». As a result, these two letters have different character codes; although the common letters of Arabic and Persian have the same character codes in the Unicode Standard. The difference between Persian «ﯼ» and Arabic «і» is only in terms of the positions where «ﯼ» is used at the end of the word or used separately. In this case «ﯼ» is used as «і» with two dots below «і». However, the Persian «ﯼ» does not have dots in this position. However, if «ﯼ» is used in the middle or at the beginning of a word, two shapes of «cancellationToken» for the beginning of the word and «і» for the middle position (Table 1) is the same for Arabic and Persian languages. As a result, in the Unicode Standard, two different characters have been allocated to «ﯼ» for Arabic and Persian.

The letter «ﮏ» has the same state. This means that it has different shapes in Persian and Arabic. As a result, in the Unicode Standard, two different characters have been allocated to «ﮏ» for Persian and Arabic. If the letter «ﮏ» is used at the end of the word or comes separately, it is written as «ﮏ» in Persian but as «ک» in Arabic. However, in both Arabic and Persian, if «ﮏ» is used at the beginning, it would be written as «کـ» and if it is used in the middle, it would be written as «ـﮑـ» (Table 1).

<table>
<thead>
<tr>
<th>Word Position</th>
<th>Language</th>
<th>Letter «ﯼ»</th>
<th>Letter «ﮏ»</th>
</tr>
</thead>
<tbody>
<tr>
<td>At the beginning</td>
<td>Persian</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td></td>
<td>Arabic</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td>In the middle</td>
<td>Persian</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td></td>
<td>Arabic</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td>At the end</td>
<td>Persian</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td></td>
<td>Arabic</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td>Isolated</td>
<td>Persian</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
<tr>
<td></td>
<td>Arabic</td>
<td>ﯽ</td>
<td>ﯼ</td>
</tr>
</tbody>
</table>

The above-mentioned characteristic in «ﮏ» and «ﯼ» enables us to use them to hide information in Arabic and Persian texts written in Unicode format.

The method can be described as follows:

If we come across any «ﮏ» or «ﯼ» (used at the beginning or in the middle) in the text, we choose one of the Persian or Arabic characters of «ﮏ» or «ﯼ» considering the information in question for hiding in the text. That is to say, if we want to hide the bit 0, we use the Persian characters of «ﮏ» or «ﯼ» and if we are going to hide the bit 1, we choose the Arabic characters of «ﮏ» or «ﯼ». This way, we hide the information in texts without changing the appearance of the text. Furthermore, the volume of the information hidden in the text is hidden at the beginning of the text so that we can extract the right amount out of the text in the future (see Figure 3).
To extract the information from the text containing information (stego text), we respectively investigate the letters of «ﮎ» and «ﯼ» in the text (only if these two letters have been used at the beginning or in the middle of a word). If the character is Persian «ﮎ» or «ﯼ», it means that the bit 0 is hidden in the text. If the character is Arabic «ك» or «ي», it means that the bit 1 is hidden in the text. By putting all the bits of 0 and 1 next to each other, we can extract the hidden information from the text.

4. EXPERIMENTAL RESULT

In this project, the information is hidden in Persian and Arabic texts using the Unicode Standard. In this project, two programs named «Steganography Program» and «Extractor Program» are written which are used, respectively, for hiding information in texts and extraction of information from the stego text. The structure of steganography and extractor programs are as follows:

The codes of Persian and Arabic characters of «ﯼ» in the Unicode Standard are 06CC and 064A, respectively. The codes of Persian and Arabic characters of «ﮎ» in the Unicode Standard are 06A9 and 0643, respectively. At first, the steganography program changes the secret text into a sequence of bits (0 and 1). The size of the hidden data is also hidden at the beginning of the text through this method, in order to prevent the extraction of the additional data at the time of the extraction of the data from the stego text.

The program searches for «ﮎ» and «ﯼ» in the cover text. The characters are at the beginning or in the middle of a word and are surely connected to the next letters. After finding one of these two letters of «ﮎ» or «ﯼ» (used in the middle or at the beginning of words), if the bit 0 is to be hidden, the Persian character of that letter —the code of 06CC for the letter «ﯼ» and the code of 06A9 for the letter «ﮎ»— is substituted for that character. However, if the bit 1 is to be hidden, the Arabic character of that letter — the code 064A for the letter «ي» and the code 0643 for the letter «ك»— is substituted for that character.

The extractor program acts conversely to the steganography program. After finding the letters of «ﮎ» and «ﯼ» used at the beginning or in the middle of words, it determines whether they are Arabic or Persian characters through their Unicode code. If the concerned letter is Persian — its code is 06CC or 0643 —, then it is clarified that the bit 0 has been hidden in the text. However, if the concerned letter is Arabic — its code is 064A or 0643 — then it is proved that the bit 1 has been hidden in the text. This procedure continues until all the bits hidden in the text are extracted. As mentioned earlier, the size of the hidden data is hidden at the beginning of the text. Therefore, at first, the size of the hidden data is extracted and used to extract the correct amount of data. Both of these programs, text steganography and extractor, have been written in Java programming language.

To test the programs, we chose several files containing text, image, and executable files. Then these files were compressed in order to decrease their size. Then the compressed files were hidden in some Arabic and Persian texts by the text steganography program. Afterwards, the files hidden in the texts were extracted with the Extractor program. Then these files were decompressed. By comparison of the output and input files, we found out that both files were the same.
Afterwards, we investigated some selected texts printed in some popular Iranian newspapers concerning their potential capacity for text steganography. In text steganography, the data must be hidden in a way that nobody notices it. As a result, when we want to do some text steganography in newspapers and magazines, it is better to hide the data in the inner pages in order to avoid any curiosity. Consequently, in this project we selected some sport articles printed in some popular Iranian newspapers for our investigation.

The Internet website of each newspaper and the potential capacity of each text concerning text steganography are shown in Table 2. All the texts have been selected from newspapers dated 13 August, 2006.

<table>
<thead>
<tr>
<th>Newspaper</th>
<th>WebSite Address</th>
<th>Text Size (Kilobyte)</th>
<th>Text Capacity (bit) (Bit)</th>
<th>Capacity Ratio (Bit/Kilobyte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afarinesh</td>
<td><a href="http://www.iranpress.ir/afarinesh">www.iranpress.ir/afarinesh</a></td>
<td>56.7</td>
<td>1736</td>
<td>31</td>
</tr>
<tr>
<td>Hamshahri</td>
<td><a href="http://www.hamshahri.net">www.hamshahri.net</a></td>
<td>22.4</td>
<td>738</td>
<td>33</td>
</tr>
<tr>
<td>Javan</td>
<td><a href="http://www.javandaily.com">www.javandaily.com</a></td>
<td>32.8</td>
<td>1156</td>
<td>36</td>
</tr>
<tr>
<td>Jomhouri Eslami</td>
<td><a href="http://www.jomhourieslami.com">www.jomhourieslami.com</a></td>
<td>31.4</td>
<td>947</td>
<td>30</td>
</tr>
<tr>
<td>Keyhan</td>
<td><a href="http://www.kayhannews.ir">www.kayhannews.ir</a></td>
<td>46.9</td>
<td>1546</td>
<td>33</td>
</tr>
<tr>
<td>Quds</td>
<td><a href="http://www.qudsdaily.net">www.qudsdaily.net</a></td>
<td>69.8</td>
<td>2406</td>
<td>34</td>
</tr>
<tr>
<td>Sarmayeh</td>
<td><a href="http://www.sarmayeh.net">www.sarmayeh.net</a></td>
<td>37.0</td>
<td>1341</td>
<td>36</td>
</tr>
<tr>
<td>Shargh</td>
<td><a href="http://www.sharghnewspaper.com">www.sharghnewspaper.com</a></td>
<td>38.9</td>
<td>1302</td>
<td>33</td>
</tr>
</tbody>
</table>

5. ADVANTAGES

In this section, some of the advantages of our proposed method are described.

1- In this method, only the letters which have similar shapes in both Arabic and Persian are replaced with each other. Consequently, we did not make any apparent changes in the original text by hiding data, while in most text steganography methods, such as line shifting, word shifting, and especially the open spaces method, it is evident that the text has been changed. Therefore, even if the reader has the original text, it is impossible for him to realize the hiding of the data by merely observing the appearance of the text. However, the original texts are usually not available to the observer with text steganography methods. Therefore, the main goal of text steganography— that is, the impossibility of detection of the presence of data— has been achieved.

2- This method is not dependent on any special format and we can also save it in numerous formats such as HTML pages or Microsoft Word documents, because the stego Unicode texts will not change because of copying and pasting and the hidden data in texts remains intact. Even if we change the style of the texts in the formatted files such as Word documents, such as by changing the text font size or making it bold or italic, the hidden data still remains unchanged.

3- In case of printing the stego text containing the hidden data, the hidden data will be lost, because, as mentioned earlier, due to the hiding of the data in the text, the text’s appearance remains unchanged and only the internal structure of the saved file is changed. Consequently, the hidden data will be lost because of losing the internal data of the file and we cannot extract the hidden data from the printed copy of the text. As a result, this method is limited to hiding data in electronic documents (e-documents).

4- In some steganography methods, such as syntactic and semantic methods, the standard structure of the text will be disarranged and spelling and grammatical errors will be created in the text. However, in this method, the appearance of the text will not change at all and the text still remains standard.

5- The use of the Unicode Standard for writing texts in different languages has been welcomed warmly. Therefore, this method can be implemented and used on different systems and devices because most of them support the Unicode Standard. As a result, a wide range of users can use this method.

6- Arabic is the official language of Muslims and about two billion Muslims live throughout the world. In view of this, using Arabic Unicode Standard covers a wide range of users.
6. CONCLUSION

In this paper, a new method for steganography in Persian and Arabic texts is presented by using the Unicode Standard. In this method, we hide data in texts by changing the Persian and Arabic characters for the letters «ﮎ» and «ﯼ». Arabic and Persian letters «ﮎ» and «ﯼ» have the same initial and medial forms, but different final and isolated forms. Therefore, they have different Unicode codes.

The main goal in designing this method is perceptual transparency. Our method has an excellent perceptual transparency because the stego text which the user sees is exactly similar to the original text. As we mentioned in the introduction, the perceptual transparency is in contradiction to hiding capacity and robustness. Therefore, the hiding capacity of our method is not very high, although it is acceptable in comparison to other text steganography methods. In addition, our method is vulnerable to some attacks such as retyping. However, it is robust to digital copy-past operation, which means that copying and pasting the text between computer programs preserve hidden information.

The digits of Persian and Arabic are different from each other in the Unicode Standard, while only the three digits of 4, 5, and 6 have different shapes in Persian and Arabic. As a result, we can use the suggested method for text steganography for the rest of the Persian and Arabic similar digits.

Since the writing systems of Pashto (the official language of Afghanistan) and Urdu (the official language of Pakistan) are similar to Arabic and Persian, we can apply this method to these two languages too.

In addition to the mentioned items in part 3.1, Arabic and Persian have other specific characteristics which we can use for text steganography.

This method can also be used for secret communication and for the prevention of the illegal reproduction and distribution of the texts, especially e-documents.

REFERENCES


