A Novel Arabic Text Steganography Method Using Extensions

Wael Al-Alwani, Abdulelah Bin Mahfooz, and Adnan Abdul-Aziz Gutub

***Abstract*—**This paper presents a new steganography approach suitable for Arabic texts. It can be classified under steganography feature coding methods. The approach hides secret information bits within the letters benefiting from their inherited points. To note the specific letters holding secret bits, the scheme considers the two features, the existence of the points in the letters and the redundant Arabic extension character. We use the pointed letters with extension to hold the secret bit 'one' and the un-pointed letters with extension to hold 'zero'. This steganography technique is found attractive to other languages having similar texts to Arabic such as Persian and Urdu.

***Keywords*—**Arabic text, Cryptography, Feature coding, Information security, Text steganography, Text watermarking.

I. Introduction

THE daily increasing number of networks users, the Internet for example, drives the process of enhancing security into more serious measures as more victims and attackers are brought into these networks. In the first quarter of 2008, more than 1.3 billion Internet users were exchanging data [1] that can be very sensitive for each user. So, data's confidentiality must be preserved. One way to assure the confidentiality property of the exchanged data is Steganography.

Steganography is described as the art and science of hiding data in an unremarkable cover object so that an eavesdropper will have no suspicions regarding this communication [2]. Steganography systems can be implemented in texts, pictures, and sound files. In this paper we will deal with Steganography in text files.

The steganography system has two inputs as shown in figure 1: a plain text, "Cover Object", **and** a message, "Secret Object", which is confidential. Steganography algorithm comes into picture to do the embedding part for the two inputs and then outputs the "Stego Object" which is nothing but the "Cover Object" with the, almost, undetectable hidden message i.e. "Secret Object".

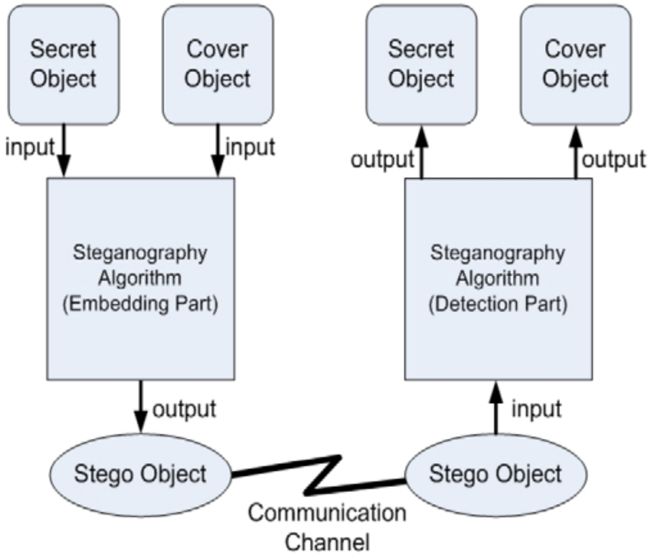

Figure 1: Steganography system

The algorithm implemented in a steganography system has to consider three features: capacity, security, and robustness [3]. Capacity is the amount of the Secret Object's bits that can be hidden in the Cover Object. Security is to make an eavesdropper unable to detect the existence of the Secret Object's data in the Stego Object. Robustness is the amount of tolerance a Stego Object can have when being modified by an attacker to keep the integrity of the embedded Secret Object valid [2].

Section 2 of this paper discusses related published work in Arabic text Steganography. Also, it lists advantages and disadvantages of each work. In section 3, our new Arabic text Steganography method is discussed. Finally, the paper ends with a conclusion in section 4.

II. Published Security Method Specified For Arabic Text

Recent works have focused on the development and the potential applications of Arabic script steganography. The first proposal in this field was done by Shirali-Shaherza [5]. Their schema was based on hiding binary values into Arabic or Persian scripts using a feature coding method. This method

Manuscript received March 4, 2007. This work is discussed by the Crypto-Group at Computer Engineering Department of King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia.

Adnan Abdul-Aziz Gutub is with the Computer Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, SAUDI ARABIA (phone: +966-3-860-1723; fax: +966-3-860-3059; e-mail: gutub@kfupm.edu.sa).

Wael Al-Alwani and Abdulelah Bin Mahfooz are fresh graduates with B.S degree in Computer Engineering from KFUPM, Dhahran, Saudi Arabia.

depends on the points inherited in the Arabic and Persian letters.

The points' location within the pointed letters hide information as follows: First, the length of the hidden information is looked at as binary with the first several bits. Then, the medium text is scanned. Whenever a pointed letter is detected, the location of the point may be affected if hidden binary value is one or zero. The location of the point is slightly shifted up if the hidden bit value is one as shown in figure 2; otherwise, the location remains unchanged.

This schema has many advantages; for example, it has high capacity in storing large number of hidden bits as the Arabic language has 15 out of 28 letters which have points. However, one of the main disadvantages of this method is in robustness. The output font is not standard so the receiver will not be able to extract the secret message if the output font is not installed on his machine. Also, the hidden information can be lost in any retyping or scanning process.

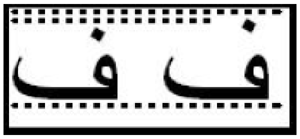

Figure 2: Shifting the letter "Fa'a" point up to hide bit value equals 1

Another proposal [6] aimed to utilize the advantages of diacritics in Arabic scripts to implement steganography. There are eight different diacritical symbols in Arabic, and they are used in this approach to hide binary bits in the original cover media. The team, who proposed this approach, found that in Standard Arabic, the frequency of one diacritic, namely Fatha, is equal to the total frequency of the other seven diacritics. So, they assigned in this approach the diacritic Fatha to the bit value equals one, and the remaining seven diacritics were assigned to the bit value equals zero.

To implement this approach, a diacritized Arabic text is used as a Cover Object. Then, a computer program reads the first bit of data needed to be embedded. If the first bit was a one and the first diacritic in the cover media was Fatha, the diacritic is kept in the cover media and the index for both the embedded data and the cover media is incremented. However, if the diacritic is not Fatha and the bit value is one, the diacritic is removed from the cover media and, in the same time, the index of the cover media only is incremented to read the next diacritic. The same process is implemented for the bit value zero, except that a zero will search for all the other seven diacritics instead of the Fatha. An example to this approach can be seen in figure 3.

The use of diacritics in written Standard Arabic language is optional, and it is uncommon nowadays to write a diacritized Arabic text. So, a drawback for this approach is the high probability to raise suspicions for an eavesdropper about the existence of a secret message when using this approach. Another important drawback is that the receiver has to have the original text so that the extracting algorithm can compare the diacritics in the Stego Object with the original Cover Object to extract the Secret Object.

Cover Object	حَدَّثَنَا سُفْيَانُ عَنْ يَحْيَى
Secret Object	**E7 (= 11100111)**
Stego Object	حَدَّثَنا سُفْيَان عَن يَحْيى

Figure 3: Hiding "E7" using diacritics approach

Another proposed approach [4] uses the redundant Arabic extension character "-". Arabic language has 28 letters and an extension character which is considered as a redundant character meant for formatting the Arabic electronic typing. However, due to the nature of Arabic writing, the extension character can not be added at the beginning or ending of words. It can be added between connected letters in a word. Note that adding the extension character does not affect the Arabic context meaning [4].

In this approach, if an extension is used after a pointed Arabic letter in the cover medium, this means that a secret bit which equals one is hidden. In the other hand, a secret bit equals zero is hidden if an un-pointed Arabic letter is followed by an extension as shown in figure 4. Using this character features security and robustness. But, it might have some drawbacks in capacity of the cover medium if the size of secret bits in the Secret Object is large.

Secret bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Steganographic text	مـن حسن اسلام المرء تـركـه مـالا يـعـنـيه ↑↑ ↑ ↑ ↑ ↑ 1 1 0 0 1 0

Figure 4: Hiding secret bits using extension character

III. PROPOSED ARABIC TEXT STEGANOGRAPHY METHOD

The secret object is hidden in the form of zeros and ones which represents the 8-bit Unicode of each character (using the UTF-8 encoding scheme). A common drawback in all previous approaches is that they embed the bits without having some optimization. Meaning that, these approaches will embed the 8 bits of each character in the Cover text.

Our novel method benefits from the work with the extension character as proposed by Gutub and Fattani [4].

We will add one extension representing bit=0 and two consecutive extensions when bit=1. The extension will be placed after any letter that can hold it. The optimization part of the algorithm deals with the message to be hidden, i.e. the Secret Object. As mentioned before, Arabic language has 28 letters. But there are special forms of a letter like in letter "Alef (ا)": (أ , إ , آ , …) which are used in Arabic writing and each has a Unicode representation. So, the number of letters and forms sums up to more than 32 and less than 64 and as each letter and form is represented by 8 bits, we used a mapping table in which each letter was assigned, instead, a 6-bit code to save 2 bits. In the mapping table, we assigned the 6-bit codes starting from 000000 and incremented by one to all letters and forms ordered alphabetically. Hence, saving 2 bits by implementing the mapping table means, enhancing the capacity feature of this method.

There is one concern is that when the Cover Object for example is one page long and the Secret Object is only one word, then, the Stego Object will only contain extensions in the first few lines. So, this will be suspicious for an eavesdropper who might infer the existence of a hidden message in the text.

So, we solved this by creating a special character named as “finishing character” which has the code 111111 and it will be embedded just after the last letter of the message, i.e. the Secret Object. After this “finishing letter”, the algorithm will randomly add extensions to the whole text just to get rid of any kind of suspiciousness. Finally, extracting the message from the Stego Object is done by collecting the extensions back and when 6 bits are collected, the algorithm checks the mapping table to determine the corresponding letter. When it detects the finishing character, it stops.

Figure 5 shows how the method works. The first letter in the Secret Object is “Ba’a” which has the code 000001 in the mapping table. We can see that the first and second letters in the first word in the Cover Object can hold extensions, so one extension is added after each one representing the first and second 0 bits of the code **00**0001, as shown in the Stego Object. The third letter can not hold an extension, so it is kept as is. The second and third words in the Stego Object hold the 4th, 5th, and 6th zero bit of the code, i.e. 00**00**1. Finally, the fourth word is holding two consecutive extensions representing the remaining bit of the code which is 1, i.e. 00000**1.**

Secret Object	بدأ اختبار
Cover Object	ميزة هذا النوع من المعالجات أنه يقضي مدة ثابتة في تنفيذ أي تعليمة، ومقدار هذا الوقت هو دورة واحدة يحدد زمنها أطول تعليمة من مجموعة التعليمات وهي تعليمة القراءة من وحدة الذاكرة، وهذا يعني سهولة كبيرة في تصميم المعالج
Stego Object	مـيـزة هذا الـنـوع مـن الـمـعـالـجـات أنــه يـقـضــي مـدة ثـابـتــة فـي تـنـفـيـذ أي تـعـلـيـمـة، ومـقـدار هذا الـوقــت هــو دورة واحـدة يـحـدد زمـنـهـا أطول تـعـلـيـمـة مــن مـجـمـوعـة الـتـعـلـيـمـات وهـي تــعــلــيــمــة الــقــراءة من وحـدة الـذاكـرة، وهـذا يـعـنـي سـهـولـة كـبـيـرة فـي تـصـمـيـم الـمـعـالـج

Figure 5: Embedding secret bits using extensions

We found that adding one extension when the bit of the Secret Object is 0 and two consecutive extensions when the bit is 1 after any letter or form that can accept extensions is somehow suspicious. An eavesdropper will find extensions at any applicable position for an extension in the Stego Object.

We proposed a double impact solution that can solve this suspiciousness problem and can enhance more the capacity feature. We modified the algorithm of our method to not add an extension when the letter code in the mapping table has two consecutive zero bits. In other words, the code of letter “Ba’a” which is 000001 will not be embedded in the Cover Object as: 1 extension, 1 extension, 1 extension, 1 extension, 1 extension, 2 extensions. It will be embedded instead as: no extension, no extension, 1 extension, 2 extensions. Obviously, we saved two more bits (extension) locations and allowed the absence of extensions in the Stego Object to remove the suspiciousness that may arise.

A final modification was made to enhance the capacity more. We modified the method to check the Secret Object, before embedding it, and make statistics on it to find the most frequent (existing) letters. Then, the method will assign the letter codes which have more consecutive zeros in the mapping table to the most frequent letters. For example, if we find that the Secret Object has the letter “Lam” (ل) as the most frequent letter to occur and then the letter “Noon” (ن), then the method will assign the codes 000000 to “Lam” and 000001 to “Noon” in the mapping table. So, we will enhance the capacity by using the frequency of occurring of the letters. Note that calculating the frequencies is a dynamic process and each time the Secret Object changes, the calculation is applied and hence, the assignment of codes for most occurring letters is changed accordingly.

IV. Conclusion

This paper presents a novel steganography scheme useful for Arabic language electronic writing. It benefits from the feature of having points within more than half the text letters. We use pointed letters to hold secret information bit ‘one’ and the un-pointed letters to hold secret bit ‘zero’. Not all letters are holding secret bits since the secret information needs to fit in accordance to the cover-text letters. Redundant Arabic extension characters are used beside the letters to note the specific letters holding the hidden secret bits. The nice thing about letter extension is that it doesn’t have any affect to the writing content.

This method featured security, capacity, and robustness, the three needed aspects of steganography that makes it useful in hidden exchange of information through text documents and establishing secret communication. This steganography technique is also useful to other languages having similar texts to Arabic such as Persian and Urdu scripts, the official languages of Iran and Pakistan, respectively. These characteristics and features promises that this novel Arabic text steganography method using letter extensions attractive for information security.

ACKNOWLEDGMENTS

Thanks to King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, for its support of all research work.

REFERENCES

[1] Internet World Statistics Website. <http://www.internetworldstats.com/stats.htm>.

[2] N. Provos and P. Honeyman, "Hide and Seek: An Introduction to Steganography", *IEEE Security & Privacy*, pp. 32-44, May/June 2003.

[3] B. Chen and G.W. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," *IEEE Trans. Information Theory*, Vol. 47, No. 4, pp. 1423-1443, 2001.

[4] Adnan Gutub and Manal Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria, May 25-27, 2007.

[5] M. Hassan Shirali-Shahreza, Mohammad Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography," *5th IEEE/ACIS International Conference on Computer and Information Science (ICIS-COMSAR 06)*, pp. 310- 315, July 2006.

[6] Mohammed Aabed, Sameh Awaideh, Abdul-Rahman Elshafei, and Adnan Gutub, "Arabic Diacritics Based Steganography", *IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*, Pages: 756-759, Dubai, UAE, 24-27 November 2007.