

UTILIZING EXTENSION CHARACTER ‘KASHIDA’ WITH POINTED LETTERS FOR ARABIC TEXT DIGITAL WATERMARKING

Adnan Abdul-Aziz Gutub, Lahouari Ghouti, Alaaeldin A. Amin, Talal M. Alkharobi
Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, SAUDI ARABIA
gutub, lahouari, amindin, talalkh @kfupm.edu.sa

Mohammad K. Ibrahim
School of Engineering and Technology, De Montfort University, Leicester LE1 9BH, United Kingdom
ibrahim@dmu.ac.uk

Keywords: Arabic text, Cryptography, Feature coding, Information security, Steganography, Text watermarking.

Abstract: This paper exploits the existence of the redundant Arabic extension character, i.e. Kashida. We propose to use pointed letters in Arabic text with a Kashida to hold the secret bit ‘one’ and the un-pointed letters with a Kashida to hold ‘zero’. The method can be classified under secrecy feature coding methods where it hides secret information bits within the letters benefiting from their inherited points. This watermarking technique is found attractive too to other languages having similar texts to Arabic such as Persian and Urdu.

1 INTRODUCTION

DIGITAL WATERMARKING, in today’s electronic era, is the process of embedding data called a watermark into a multimedia object such that the watermark can be detected whenever necessary. It is the ability of injecting information in redundant bits of any unremarkable cover media. Its objective is to keep the injected data undetectable or unchangeable without destroying the cover media integrity. Digital watermarking can be implemented by replacing unneeded bits in image, sound, and text files with secret watermarking data. Watermarking main benefits can be in copyright protection and related issues. It gives an idea about the possible unauthorized replication and manipulation of electronic data. It can protect the intellectual property (IP) rights specifically the digital rights management (DRM) systems necessities.

Capacity, security, and robustness (Chen, 2001), are the three main aspects affecting digital watermarking and its usefulness. Capacity refers to the amount of watermarking bits that can be hidden or injected in the cover medium. Security relates to the ability of an eavesdropper to figure-out or modify the watermarking information easily. Robustness is concerned about the resist possibility

of destroying the watermarking data. Watermarking is different than Steganography and cryptography although they all have overlapping usages in the information hiding processes (Provos, 2003). Watermarking and Steganography security hides the awareness that there is information in the cover medium, where cryptography reveals this knowledge but encodes the data as cipher-text and disputes decoding it without permission. Watermarking is different from steganography in its main goal. Watermarking aim is to protect the cover medium from any modification with no too much emphasis on secrecy. It can be observed as steganography that is concentrating on high robustness and low security.

Languages and their structures play differences in the preferred watermarking system. Normally no single technique is to be used for all languages (Provos, 2003). 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. Our proposed approach exploits the existence of the redundant Arabic extension character, i.e. Kashida and the pointed letters, benefiting from the steganography method presented in (Gutub, 2007). We propose 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 will be used for analysis and comparison with another method proposed for Arabic text secrecy.

The paper flow is as follows. The next section, Section 2, presents some idea of watermarking. A proposed method by Shirali-Shahreza for hiding information in Arabic texts is detailed in Section 3. Section 4 discusses our new Arabic text watermarking technique using Kashida (character extensions) and pointed letters. Some comparisons and experimental results are presented in Section 5. The conclusion is given in Section 6.

2 DIGITAL WATERMARKING

Digital watermarking process enables injecting watermarking information as redundant bits of any cover media. Its applications varies but utilized best for protecting data originalities in case of a violation as real copyright protection (Provos, 2003). Its original aim is to protect the cover medium from claiming its credit by others, with low emphasis on secrecy.

Most watermarking and security research use cover media as pictures (Chandramouli, 2001), video clips (Doërr, 2003) and sounds (Gopalan, 2003). However, text digital watermarking is not normally preferred due to the difficulty in finding redundant bits in text files (Gutub, 2007). The structure of text documents is related to what is seen much more than all other cover media types, making the hiding of information in other than texts easy without a remarkable alteration. The advantage to prefer text watermarking over other media is its usage popularity, smaller memory occupation, and simpler communication (Shirali-Shahreza, 2006).

3 SHIRALI-SHAHREZA ARABIC TEXT HIDING METHOD

Shirali-Shahreza (2006) proposed a special character feature security method for Arabic and Persian letters. Their scheme depends on the points inherited in the Arabic and Persian letters (Shirali-Shahreza, 2005), which are some who very similar. The concentration in this study will be on it related to Arabic language.

Although, both Arabic and English languages have points in their letters, the amount of pointed

letters differ too much. English language has points in only two letters, small "i" and small "j", while Arabic has in 15 letters out of its 28 alphabet letters as shown in Figure 1. This large number of points in Arabic letters made the points in any given Arabic text remarkable and can be utilized for information security and watermarking as presented by Shirali-Shahreza in their “new approach to Persian/Arabic text steganography” (Shirali-Shahreza, 2006).

un-pointed letters	pointed letters
ا ح د ر س ص ط ع ك ل م ه و	ب ت ث ج خ ذ ز ش ض ظ غ ف ق ن ي

Figure 1: Arabic letters

Shirali-Shahreza proposed to hide information in the points of the Arabic letters. To be specific, they hide the information in the points’ location within the pointed letters. First, the hidden information is looked at as binary with the first several bits (for example, 20 bits) to indicate the length of the hidden bits to be stored. Then, the cover medium text is scanned. Whenever a pointed letter is detected its’ point location may be affected by the hidden info bit. If hidden value bit is one the point is slightly shifted up; otherwise, the concerned cover-text character point location remains unchanged.

This point shifting process is shown in Figure 2 for the Arabic letter ‘Noon’. “In order to divert the attention of readers, after hiding all information, the points of the remaining characters are also changed randomly” (Shirali-Shahreza, 2006). Note that, as mentioned earlier, the size of hidden bits is known and also hidden in the first 20 bits.



Figure 2: Point shift-up of Arabic letter ‘noon’

This method of point shifting may have its advantages in security and capacity; it features good secret storing of large number of hidden bits within any Arabic text. However, it has main drawback in robustness making it unpractical. For example, the hidden information is lost in any retyping or scanning. The output text has a fixed frame due to the use of only one font. In fact, this information security method is appropriate with its font type of characters, which is not standard and can be lost or changed easily.

4 PROPOSED ARABIC TEXT WATERMARKING METHOD

Benefiting from Shirali-Shahreza (Shirali-Shahreza, 2006) point steganography and trying to overcome the negative robustness aspect, we propose a new method to hide info in any letters instead of pointed ones only. We use the pointed letters with extension (Kashida) to hold secret bit ‘one’ and the un-pointed letters with Kashida to hold secret bit ‘zero’. Note that the Kashida doesn’t have any affect to the writing content. It has a standard character hexadecimal code: 0640 in the Unicode system. In fact, this Arabic Kashida character in electronic typing is considered as redundant character only for arrangement and format purposes.

The only bargain in using Kashida is that not all letters can be extended with this extension character due to their position in words and Arabic writing natural structure. The Kashida can only be added in locations between connected letters of Arabic text; i.e. Kashida cannot be placed after letters at end of words or before letter at beginning. Our proposed watermarking hypothesis is that whenever a letter cannot have an extension or found intentionally without extension it is considered not holding any secret bits.

This proposed digital watermarking method can have the option of adding Kashida before or after the letters. To be consistent, however, the location of the extensions should be the same through out the complete document with watermarking.

Assume we add Kashida after the letters. Figure 3 shows an example to detail this watermarking process. We first select the secret bits to be hidden (say 110010) looking from the least significant bits to be started with. The first secret bit found is ‘0’ to be hidden in an un-pointed letter. The cover-text is scanned from right to left due to Arabic regular direction. The first un-pointed letter in the cover-text is found to be the first, known as ‘meem’. This ‘meem’ should hold the first secret bit ‘0’ noted by adding extension character after it.

The second secret bit is ‘1’ and the second letter of the cover-text is pointed known as ‘noon’. However, this letter position cannot allow extension, forcing us to ignore it. The next possible pointed letter to be extended is ‘ta’. Note that a pointed letter ‘noon’ before ‘ta’ is not utilized due to its unfeasibility to add extension character after it.

The same watermarking example of securing: 110010 in the Arabic text, illustrated earlier, is

readjusted assuming the extensions added are before the letters, as shown in Figure 4.

Watermarking bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Output text	من حسن اسلام المرء تركه مالا يعنيه 

Figure 3: Watermarking adding Kashida after letters

To add more security and misleading to trespassers, both options of adding extensions before and after the letters can be used within the same document but in different paragraphs or lines. For example, the even lines or paragraphs use watermarking of extensions after the letters and the odd use extensions before or visa versa.

Watermarking bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Output text	من حسن اسلام المرء تركه مالا يعنيه 

Figure 4: Watermarking by adding Kashida before letters

5 EXPERIMENTATIONS

We compare the capacity of our Kashida approaches to the dots approach of (Shirali-Shahreza, 2006). First, we need to note that in our Kashida methods, hiding a bit is equivalent to inserting a character. The dots approach doesn’t suffer such increase in size due to hidden message embedding. In fact, the dotted approach can be viewed as an ideal (hence, unpractical) case for the Kashida method.

Since there are several scenarios for implementations, we count the number of usable characters per approach, independent from the scenario or the watermarking secret message to be embedded. For this goal to be realistic, we find utterances in the Corpus of Contemporary Arabic (CCA), by (Al-Sulaiti, 2004). The corpus is reported to have 842,684 words from 415 diverse texts, mainly from websites.

Our proposed work was implemented using (Al-Sulaiti, 2004) for comparison reasons with the dots proposal of (Shirali-Shahreza, 2006). We use p for the ratio of characters capable of baring a secret bit of a given level, and q for the ratio of characters capable of baring the opposite level. In the case of the dots approach, dotted characters may contribute to p while un-dotted characters may contribute to q . For the Kashida method, we study the two cases: the

case of inserting Kashida before, and the case of inserting them after, the required character. We count extendible characters before/after dotted characters for p and those before/after un-dotted characters for q . The last column assumes equal-probability of $(p+r)$ and q .

Table 1: Comparing Kashida with dots approaches

Approach	P	q	r	$(p+r+q)/2$
Dots	0.2764	0.4313	0.0300	0.3689
Kashidah-Before	0.2757	0.4296	0.0298	0.3676
Kashidah-After	0.1880	0.2204	0.0028	0.2056

The figures in Table 1 are quite near. As pointed out previously, the dots approach is actually the ideal unpractical case for our Kashida method. The program was also tested under various formats and results are reported in Table 2. It produced an average capacity of 1.22%.

Table 2: Kashida experiments for different file-types

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%

6 CONCLUSION

This paper presents a watermarking 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 Kashida (extension character) is used beside the letters to note the specific letters holding the hidden secret bits. The nice thing about Kashida is that it doesn’t have any affect to the writing content.

This method featured security, capacity, and robustness, the three needed aspects of data hiding and watermarking. Our proposed method is evaluated and compared to a previous method showing similar performance but with the advantage of using standard fonts. This Arabic text watermarking technique is also useful to other languages having similar texts fonts structure such as Persian and Urdu scripts, the official languages of Iran and Pakistan, respectively. These characteristics

and features promises that this Arabic text watermarking method using Kashida joint to pointed letters is attractive in the information security field.

It should be noted, at the end, that this research idea is not restricted only for the Arabic, Persian and Urdu scripts. Most of the Semitic languages can use some features in one form or another, and the proposed approach can be slightly modified to suit other languages requirements.

ACKNOWLEDGMENTS

Thanks to King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, for its support of all research work. Thanks to the COE 509 students of the Applied Cryptography course for all their inputs, feedback, and comments.

REFERENCES

- Al-Sulaiti, L., 2004. Designing and Developing a Corpus of Contemporary Arabic, *MS Thesis, The University of Leeds*.
- Chandramouli, R., and Memon, N., 2001. Analysis of LSB based image steganography techniques, *Proceedings of the International Conference on Image Processing*, Vol. 3, pp. 1019 – 1022.
- Chen, B., and Wornell, G., 2001. 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.
- Doërr, G., and Dugelay, J., 2003. A Guide Tour of Video Watermarking”, *Signal Processing: Image Communication*, Vol. 18, No 4, pp. 263-282.
- Gopalan, K., 2003. Audio steganography using bit modification, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '03)*, Vol. 2, pp. 421-424.
- Gutub, A., and Fattani, M., 2007. 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.
- Provos, N., and Honeyman, P., 2003. Hide and Seek: An Introduction to Steganography, *IEEE Security & Privacy*, pp. 32-44.
- Shirali-Shahreza, M., and Shirali-Shahreza, S., 2005. A Robust Page Segmentation Method for Persian/Arabic Document, *WSEAS Transactions on Computers*, vol. 4, Issue 11, pp. 1692-1698.
- Shirali-Shahreza, et. al., 2006. A New Approach to Persian/Arabic Text Steganography, *5th IEEE/ACIS International Conference on Computer and Information Science (ICIS-COM SAR 06)*, pp. 310-315.