SUST Journal of Science and Technology, Vol. 27, No. 1, 2017, P: 71-80

Steganography in Bengali Unicode Text

SUPRADIP DEY

We present a novel approach for information hiding or steganography in Bengali digital text. One important and interesting feature of Bengali alphabet is that certain characters defined in Unicode also have composite forms, which means those can be written using two different codes in Unicode. In other words, we can say that these characters have a single form as well as a composite form. The main idea of the proposed method is to exploit this special feature of Bengali alphabet to hide secret information in form of bits. One of these two forms can be used to represent the bit '0' and the other form can represent the bit '1' in a document without any risk of understanding by any intermediate user. The results show that the proposed method is very prominent to be a successful steganography technique.

Keywords: *Steganography, Bengali Unicode, Composite form, Cover media*.

1. INTRODUCTION

Steganography can be defined as a method of hiding secret data within a cover media so that other individuals fail to realize their existence. We can say that steganography is the science of hiding information. It is often confused with cryptography as both are used to protect confidential information. The difference between the two is in the appearance of the processed output; the output of steganography operation is not obviously visible but in cryptography the output is twisted so that it can draw attention. If a nefarious government or Internet service provider (ISP) is looking for encrypted messages, they can easily find them. Whereas the goal of cryptography is to make data unreadable by a third party, the goal of steganography is to hide the data from a third party. Steganlysis is the process to detect the presence of steganography. Image, audio and video are some popular media for steganography. On the other hand, text is ideal for steganography due to its ubiquity and smaller size compared to these media. However, text communication channels do not necessarily provide sufficient redundancy for covert communication.

Bengali is a native language to Bangladesh, the Indian state of West Bengal, and parts of the Indian states of Tripura and Assam. It is written with the Bengali script. With nearly 230 million total speakers, Bengali is ranked as the sixth spoken language in number of speakers in the world. Location of Bengali in Unicode is 0980 – 09FF. Other than the inherent complex language constructs, the alphabets and characters used in written form show significant complexity. Some diacritics (shorthand form of vowel signs when used with consonants) are positioned in the *left* of the accompanying consonant; some are positioned in the *right*, some others are placed *below*, and some others have positions in both *left and above* or *right and above*. All of these vowel signs or diacritics have unique code in Unicode.

In contrast, some diacritics have two parts; one is placed before the consonant and the other sits after. There is a unique code reserved for each of these signs. However, these signs can be also represented by combining two other codes and is officially permitted. For example, the sign 'ো' can be also written combining 'ে' and 'া'. On the other hand some consonants also have two parts; one is the main body and the other is a special symbol “Nukta”, which is a dot placed under the main body of the character. As a character each of them has a unique location in Unicode. The main body of all these characters has the shape of some other character, which has unique locations in Unicode and the “Nukta” itself has a unique Unicode. So these characters can be represented either by their unique code or by combination of two Unicode’s. For example ‘ড়’ can be also written by combining ‘ড’ and ‘ ় ’ without noticed by the reader. Again, some character combinations are usually represented with some special ligatures but the non-ligature forms are also acceptable. The non-ligature form can be easily obtained by inserting a zero width non-joiner (ZWNJ) character (Unicode 200C). The ligature form and the non-ligature form (for example ে and ো) can be interchangeably used to represent the bit 0 and 1. Table 1 summarizes all of these options.

Table 1: Bengali Unicode characters with two forms.

Single Form		Composite Form	
Char.	**Unicode**	**Char.**	**Unicode**
(ex.)	09CB (09AE + 09CB)	(ex.)	09C7 + 09BE (09AE + 09C7 + 09BE)
(ex.)	09CC (09AE + 09CC)	(ex.)	09C7 + 09D7 (09AE + 09C7 + 09D7)
ড়	09DC	ড়	09A1 + 09BC
ঢ়	09DD	ঢ়	09A2 + 09BC
য়	09DF	য়	09AF + 09BC
র	09B0	র	09AC + 09BC
	0997 + 09C1		0997 + 200C + 09C1
	09B6 + 09C1		09B6 + 200C + 09C1
	09A8 + 09CD + 09A4 + 09C1		09A8 + 09CD + 09A4 + 200C + 09C1
	09B8 + 09CD + 09A4 + 09C1		09B8 + 09CD + 09A4 + 200C + 09C1
	09AA + 09CD + 09A4 + 09C1		09AA + 09CD + 09A4 + 200C + 09C1
	09B9 + 09C1		09B9 + 200C + 09C1
	09B9 + 09C3		09B9 + 200C + 09C3

After the introduction of Unicode as well as some phonetic Bengali typewriting software, Bengali computing has raised to a peak. It is reflected in the number of Bengali web pages in the Internet. Similarly more Bengali documents are used in offices and organizations than ever. This vast amount of Bengali documents may be easily used as a covert media for the proposed steganography system.

The capacity of the proposed method is also very lucrative. According to Bengali linguist Munier Chowdhury and other research works on Bengali, the character 'র' has a frequency around 7%, 'য়' has around 1.9%, 'ো' has around 0.7%, and 'ড়' has around 0.25% [1]. This means the candidate characters for the proposed method are nearly 10% of total characters in Bengali text. So, we can assume that in every 4000 characters we are able to hide 400 bits. More precisely, 50 bytes of secret information can be hidden in a 8 Kbyte document. On an avergae, a standard and average Bengali news paper article's size is 10 KBytes and a pass-word or a pin number or a short secret message requires less than the mentioned 50 bytes.

Beside these, text steganography with a regional language have certain other benefits. To suspect hidden information a person must be a native or specialized user of this language and requires specialized knowledge of Bengali Unicode. So, the proposed technique is a safer method of secret data communication over an international network like the Internet.

It is important to mention that the whole encoding and decoding process is done by software system at the two communication ends. So there is no scope of introduction of any kind of bit errors due to the steganography process except those bit errors introduced by the underlying communication system and the hardware. However, it is possible for an unintended user to eavesdrop and hack to change randomly some of the special characters of the stego text containing the secret bits and corrupt the hidden secret information. Integration of a simple and conventional error detection method such as the checksum can ensure the correctness of the decoded secret data with high probability. Generally, this issue is more related to network security, compared to steganography and out of the current scope of discussion.

2. RELATED WORK

Text steganography can be broadly classified into three types: format-based, random and statistical generations, and linguistic method. Format-based methods use and change the formatting of the cover-text to hide data. They do not change any word or sentence, so it does not harm the 'value' of the cover text. Random and statistical generation methods are used to generate cover text automatically according to the statistical properties of language. These methods use example grammars to produce cover text in a certain natural language. The linguistic method considers the linguistic properties of the text to modify it. The method uses linguistic structure of the message as a place to hide information. Following is the list of works has been done on hiding information or text steganography carried out.

2.1 Through Markup Languages

Some markup language feature can be used to hide information [2]. For instance, case insensitivity of HTML tags can be exploited. For example, the tag
 can be also used as
 and
. As a result one can do text steganography in HTML documents by changing the small or large case of letters in document tags. In some cases the positions of tags are also used. For example <B><U> </B></U> or like this <U><B> </U></B>. Information can be extracted by comparing the tags positions.

2.2 Using Specific Characters in a Text

In this analytical, complicated and time-consuming method, some specific characters from certain words are selected [3]. For example, the first words of each paragraph are selected in a manner that by placing the last characters of the selected words side by side forms the secret information.

2.3 Line Shifting Method

In printed document the lines of the text are vertically shifted to some degrees [4][5] to hide secret data. If the text is retyped or if character recognition programs (OCR) are used, the hidden information would get destroyed.

2.4 Word Shifting Strategy

Information is hidden in the text by shifting words horizontally and by changing distance between words [4][6]. This method is acceptable for texts where the distance between words is varying. This method can be identified less, because change of distance between words to fill a line is quite common. Retyping of the text or using OCR programs destroys the hidden information, like the previous two cases.

2.5 Exploiting Punctuation Signs

Appropriate placement of some punctuation signs such as full stop (.) and comma (,) can hide information in a text file [3]. This method requires identifying proper places for putting punctuation signs.

2.6 Using Synonyms

By using the synonym of words for certain words one can hide information in the text [5][7]. A major advantage of this method is the protection of information in case of retyping or using OCR programs. However, this method may alter the meaning of the text.

2.7 Feature Coding Method

Some of the features of the text can be altered to hide some information [8]. For example, the end part of some characters such as h, d, b or so on, is elongated or shortened a little thereby hiding information in the text. Retyping the text or using OCR program destroys the hidden information.

2.8 Using Abbreviation

Use of abbreviations can hide some information, though with very less capacity [3]. For example, only a few bits can be hidden in a file of several kilobytes.

2.9 White Space Strategy

We can add some extra white spaces in the text [3][9] to hide some information. These white spaces can be placed at the end of each line, at the end of each paragraph or between the words. However, some text editor programs automatically delete extra white spaces and thus destroy the hidden information.

2.10 Displacing Letter Points and Diacritics

Arabic, Persian and Urdu texts are used for this technique [10][11][12] [13][14]. One of the characteristics of these languages is plenty of points in its letter. One point letters are used to hide the information by shifting position of point a little bit vertically high with respect to the standard point position in the text. The same technique is applied for the vowel signs to hide information.

2.11 By Extension Letter

Text steganography is applied on Arabic text [15] for this algorithm. Arabic language has a special extension character, which can be arbitrarily inserted between characters for formatting purposes.

2.12 MS Word Document

In this technique three bytes of secret information can be hidden as RGB color values of each invisible character such as the space, the tab and the new line characters in an MS Word document [16].

2.13 Cricket Match Scorecard

In this method, data is hidden in a cricket match scorecard by adding a meaningless zero before a number to represent bit 1 and leaving the number as it is, to represent bit 0 [17].

2.14 CSS (Cascading Style Sheet)

This technique encrypts a message using RSA public key cryptosystem and cipher text is then embedded in a Cascading Style Sheet (CSS) by using End of Line on each CSS style properties, exactly after a semicolon. A space after a semicolon embeds bit 0 and a tab after a semicolon embeds bit 1 [18].

2.15 SQL Queries

The proposed method in [19] encodes input text messages into SQL carriers made up of SELECT queries. In effect, the output SQL carrier is dynamically generated out of the input message using a dictionary of words implemented as a hash table and organized into 65 categories, each of which represents a particular character in the language.

2.16 MS Excel Document

Various techniques of steganography in MS Excel documents and relative benefits have been discussed in [20][21]. For example, the text direction in a cell can slightly be rotated based on the intended bits to hide.

3. STEGANOGRAPHY IN BENGALI

Most of the techniques described above are applicable for Bengali text also along with their inherent advantages and disadvantages. Methods in sections 2.2 through 2.14 can be easily implemented for steganography in Bengali with a little modification or no modification at all. A feature coding based approach is implemented to hide secret message in the text by shifting the specific 'matra' towards left, by shifting the points of some characters and by shifting the character 'ref' respectively [22]. A new linguistic approach for Steganography through Indian Languages such as Bengali by considering the flexible grammar structure of the languages is presented in [23]. In this work, the bits of the binary stream are encoded to some part-of-speech and create meaningful sentences starting with a suitable word belonging to the mapped part-of-speech. A quantum approach based text steganography technique has been proposed with the help of Bengali language [24]. This approach uses two specific characters and two special characters like inverted commas (opening and closing) in Bengali language and the mapping technique of quantum gate truth table. Another text steganography technique has been proposed considering the structure of Bengali alphabet, which hides secret message through changing the pattern of Bengali alphabet letters [25]. Considering the availability of more characters and flexible grammar structure of Indian Languages including Bengali, another steganography method hides the secret message in the text by creating meaningful sentences after finding the longest common subsequence of two binary string among which one is the secret message and another may be any binary string [26].

As with other languages Bengali language is rich with large set of synonyms. Flower can be represented by at least 4 Bengali words: ফুল, পুষ্প, কুসুম, and sun can be translated to the distinct words সূর্য, রবি, সূর, অরুণ, ষ্ণ, অর্ক, ভানু, ভাস্কর, . So a synonym based text steganography system in Bengali has high capacity. Additionally, the language has a large number of synonyms due to the inheritance from Sanskrit. The words চন্দ্র (originally Sanskrit) and চাঁদ (in pure Bengali) represent the word moon. Another source of availability of large number of synonyms are the "Shadhu" form and the "Cholito" form of the verbs in Bengali, though mixing of both the forms in the same document or speech is highly discouraged. For example, খাইয়াছি and খেয়েছি are the two forms to mean the finished work of eating in the first person.

Bengali speaking people use abbreviated forms of some commonly used word combinations in their daily lives and particularly in office environments. (জর য়ে), (কর), (ালয়), (গরিষ্ঠ সাধারণ নীয়ক) are a few examples of common abbreviations in Bengali. This indicates that the abbreviation based Bengali text steganography will be also strong.

By displacing letter points in the characters ড়, ঢ়, য়, and র; and the diacritics ু, ূ, and ৃ (for example in the conjuncts কু, কূ, and কৃ) we can implement a powerful Bengali text steganography system.

4. THE PROPOSED TECHNIQUE

We can call one form of the characters listed in Table 1 as the *Classical Form* and the other as the *Non-classical Form* and construct the code map in Table 2 based on this idea to commit the steganography process.

Table 2: Bengali Unicode characters with two forms.

Classical Form		Non-classical Form	
Char.	**Unicode**	**Char.**	**Unicode**
(ex.)	09CB (09AE + 09CB)	(ex.)	09C7 + 09BE (09AE + 09C7 + 09BE)
(ex.)	09C7 + 09D7 (09AE + 09C7 + 09D7)	(ex.)	09CC (09AE + 09CC)
ড়	09DC	ড়	09A1 + 09BC
ঢ়	09DD	ঢ়	09A2 + 09BC
য়	09AF + 09BC	য়	09DF
র	09B0	র	09AC + 09BC
	0997 + 09C1		0997 + 200C + 09C1
	09B6 + 09C1		09B6 + 200C + 09C1
	09A8 + 09CD + 09A4 + 09C1		09A8 + 09CD + 09A4 + 200C + 09C1
	09B8 + 09CD + 09A4 + 09C1		09B8 + 09CD + 09A4 + 200C + 09C1
	09AA + 09CD + 09A4 + 09C1		09AA + 09CD + 09A4 + 200C + 09C1
	09B9 + 09C1		09B9 + 200C + 09C1
	09B9 + 09C3		09B9 + 200C + 09C3

Let us say we will use the classical form of these special characters to reflect a bit 1 or Boolean true and use the non-classical form to reflect bit 0 or a Boolean false. We may have some secret information to hide in the document. We have to convert the secret information to its equivalent secret bit stream consisting of only 0 (Boolean false) and 1 (Boolean true). We will proceed with each bit in the bit stream and look for an occurrence of the special characters included in Table 1 (for example ‘ ো’, Unicode 09CB) in the document. If the bit is 1 we will use the classical form (ো, Unicode 09CB) and if the bit is 0 we will use the non-classical form ('ে' and 'া', Unicode 09C7 + 09BE). As these classical and non-classical forms are not differentiable by any human reader, this scheme will work with only digital documents. Any unintended user can never realize and detect this alteration of these special characters in the document. The intended receiver will do the reverse steps to recover the hidden information. In the document she/he will look for occurrences of the specific Unicode or Unicode combinations mentioned in Table 2 (for example ‘ো’, Unicode 09CB or 'ে' + 'া', Unicode 09C7 + 09BE respectively) in the received document. If the classical form is used he/she will add a 1 in the recovered bit stream and if the non-classical form of the Unicode combination is used a 0 will be added in the recovered bit stream. Algorithm 1 and algorithm 2 summarizes the encoding and decoding schemes of the proposed steganography technique.

To further strengthen the steganography process, the construction of Table 2 will be non-static. The 13 special characters can be permutated in 2^{13} or 8192 ways and Table 2 can be constructed in 8192 ways. Table 2 will be constructed dynamically in the beginning of each encoding process. The decoder side will construct the same table based on some prior information or based on some protocol. This will guarantee different encoding scheme in successive communication sessions. Table 2 is called the *code map* in the further discussion.

Algorithm-1*: Hiding Secret Bits*
Input: Cover digital document, Secret bit stream
Output: Stego digital document

1. Construct the code map (by an agreed up on prior information or protocol to choose 1 out of the 64 possible ways).

2. For each bit in the secret bit stream do

a. Repeat copying the characters from the input text to the output text until the occurrence of a special character in the code map.

b. If the bit is 0, add the classical form for the found special character to the out-

put text.

c. Else if the bit is 1, add the non-classical form for the found special character to the output text.

3. For the rest of the characters in the input, add the classical form for the special characters and copy other characters to the output.

***Algorithm-2**: Extracting Bits from the Received Digital Document*
Input: Stego digital document
Output: Extracted secret bit stream

1. Construct the code map (by an agreed up on prior information or protocol to choose 1 out of the 64 possible ways).

2. For each character in the input stego text do

a. If the character is an ordinary one do nothing and go to step 2.

b. Else if the character is an occurrence of the special characters in the code map and is in the non-classical form; add a 1 to the secret bit stream.

c. Else if the character is an occurrence of the special characters in the code map and is in the classical form; add a 0 to the secret bit stream.

3. Delete the useless consecutive 0s at the end of the decoded secret message, which fail to construct a byte or construct a byte with all bits 0.

For example, the Bengali sentence ার নার মায় can be represented as ার নার ায় to hide the secret bit stream "010101". In the original sentence candidate characters are underlined and in the stego sentence the changed characters appear in bold face for better understanding of the reader.

An encoder and a decoder application are developed in Java for the proposed data-hiding scheme. The used font is "Ekushe Lalsalu". In the encoder a secret code is taken as an input from a text field. The input or carrier text can be written in the text area or can be loaded from a saved file in the disk. The output stego text is given on a separate text area, which can also be saved to the disk. The sender either can copy the stego text from the text area and paste it in the desired medium of communication, for example in an e-mail, or can directly send the saved file to the receiver. Similarly in the decoder application the stego text can be paste on a text area or can be loaded from a file. Fig-1 and Fig-2 shows the screen shots of the developed encoder and decoder application.

We observe in Fig-1 that the secret code is "123ab", so the secret bit stream is 00110001001100100011001101100001011000010. For the secret message we assume simple ASCII coding. The input plain text is in the left text area and the encoded stego text is in the right text area. For this example we assume the single form in Table 1 to be the classical form and the composite forms of the characters to be the non-classical form. The special characters containing bit 0 are shown by an underline and those containing bit 1 is shown with a surrounding rectangle. As it is seen, one form of the special characters is not differentiable from the other form.

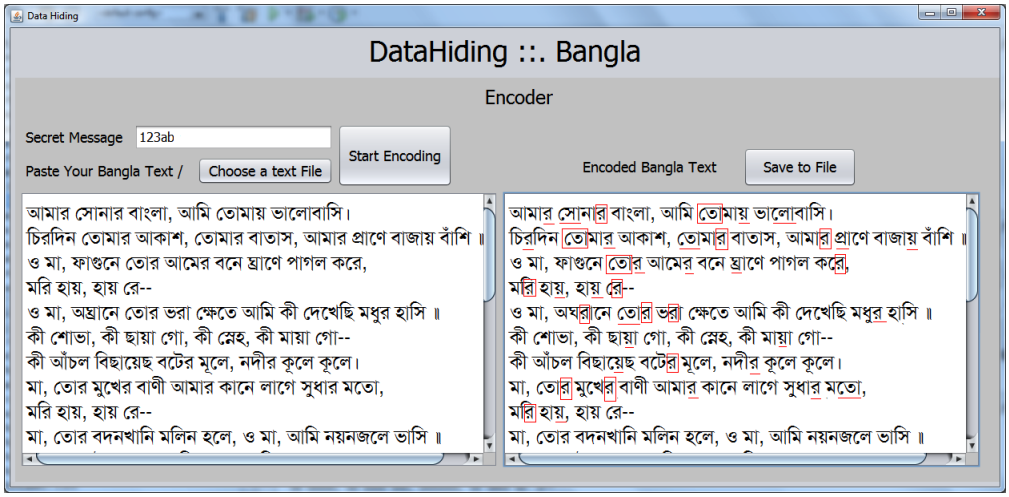

Fig- 1: Encoder of the data hiding application.

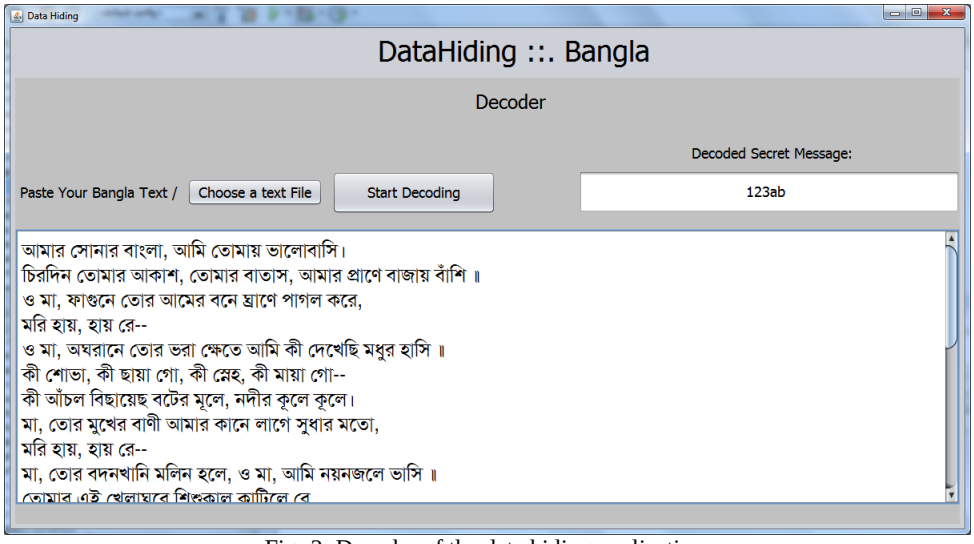

Fig- 2: Decoder of the data hiding application.

4. RESULTS AND DISCUSSIONS

To test the performance of the proposed method, several test runs were performed with the developed application software. Input texts were typed directly with the help of phonetic Bengali typing software, as well as copied from Bengali online newspapers. The developed applications hide and extract secret information with 100% accuracy in all of the performed test cases. In total 10 tests were performed.

Output of s a sample test was saved for an open survey. The participants were asked whether he/she sees or feel anything wrong with the supplied text or document, which is a primary sign ineffectiveness of a steganography method. The generated stego text was not suspected for secret message by anyone in the survey. The size of the population for the survey was 18, consisting of both Unicode aware and unaware peoples, but all with significant computer literacy. It is worth mentioning some comments from the user's subjective reply regarding the stego text: "Normal text document", "Nothing special", "Some bangla text", "Feels good", "Some text", "Bangla unicode text", "Looks all right". Only 1 person replied, "Weird way how text looks, looks normal though". A user can see some distortion to view Unicode Bengali documents if the necessary fonts are not configured accordingly. This experience is quite common for a regular user and is usually ignored.

5. CONCLUSIONS

We discussed a novel technique of hiding information in Bengali digital document. These documents are very common to any organization. The initial results achieved by our proposed system are very appealing and we are confident that our proposed method can be a good choice for text steganography.

Other interesting features of this language or the alphabet can be exploited efficiently for steganography purposes and deserves significant research. The proposed method can be further enhanced. For example, incorporation of an error detection method can make the system robust.

REFERENCES

[1] Md. Abdus Sattar, Al-Mukaddim Khan Pathan and Mohammad Ameer Ali, Development of an optimal bangla keyboard layout based on character and fingering frequency, National Conference on Computer Processing of Bangla, 2004, Independent University, Bangladesh, pp. 38-46

[2] K. Bennett, Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text, Purdue University, CERIAS Tech. Report 2004-13.

[3] T Morkel, JHP Eloff and MS Olivier, An Overview of Image Steganography, in Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005, pp. 1-11

[4] S.H. Low, N.F. Maxemchuk, J.T. Brassil, and L. O'Gorman, Document marking and identification using both line and word shifting, Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '95), April 1995,vol.2, pp. 853 - 860.

[5] A.M. Alattar and O.M. Alattar, Watermarking electronic text documents containing justified paragraphs and irregular line spacing, Proceedings of SPIE -- Volume5306, Security, Steganography, and Watermarking of Multimedia Contents VI, June 2004, pp. 685-695.

[6] Y. Kim, K. Moon, and I. Oh, A Text Watermarking Algorithm based on Word Classification and Inter-word Space Statistics, Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03), 2003, pp. 775–779.

[7] M. Niimi, S. Minewaki, H. Noda, and E.Kawaguchi, A Framework of Text-based Steganography Using SDForm Semantics Model, Pacific Rim Workshop on Digital Steganography 2003, Kyushu Institute of Technology, Kitakyushu, Japan, July 3-4, 2003.

[8] K. Rabah, Steganography-The Art of Hiding Data, Information Technology Journal, 3(3): 245-269, 2004.

[9] D. Huang, and H. Yan, Inter-word Distance Changes Represented by Sine Waves for Watermarking Text Images, IEEE Transactions on Circuits and Systems for Video Technology, 11(12): 1237-1245, December 2001.

[10] M. H. Shirali-Shahreza, and S. Shirali-Shahreza, A New Approach to Persian/Arabic Text Steganography, Proceedings of 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS, June 2006.

[11] M. H. Shirali-Shahreza, and S. Shirali-Shahreza, A Robust Page Segmentation Method for Persian/Arabic Document, WSEAS Transactions on Computers, 4(11): 1692-1698, Nov. 2005.

[12] Jibran Ahmed Memon, Kamran Khowaja, and Hameedullah Kazi, Evaluation of steganography for urdu /arabic text, journal of theoretical and applied information technology, pp 232-237

[13] Mohammed Aabed, Sameh Awaideh, Abdul-Rahman Elshafei, and Adnan Gutub, Arabic Diacritics Based Steganography, IEEE International Conference on Signal Processing and Communications (ICSPC 2007), pp. 756-759, Dubai, UAE, November 2007

[14] Adnan Gutub, Yousef Elarian, Sameh Awaideh, and Aleem Alvi, Arabic Text Steganography Using Multiple Diacritics, WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications, University of Sharjah, Sharjah, UAE, MARCH 2008.

[15] Adnan Abdul-Aziz Gutub and Manal Mohammad Fattani, A Novel Arabic Text Steganography Method Using Letter Points and Extensions, Proceedings of World Academy of Science, Engineering and Technology 21, pp. 28-31, May 2007

[16] Md. Khairullah, A Novel Text Steganography System Using Font Color of the Invisible Characters in Microsoft Word Documents, Proceedings of the 2nd International Conference On Computer And Electrical Engineering (ICCEE 2009), Dubai, UAE, pp. 482-486.

[17] Md. Khairullah, A Novel Text Steganography System in Cricket Match Scorecard, International Journal of Computer Applications, New York, USA, 21(9): 43-47, May 2011.

[18] H. Kabetta, B. Y. Dwiandiyanta, and Suyoto, Information hiding in CSS: a secure scheme text steganography using public key cryptosystem, Int. Journal on Cryptography and Information Security, vol.1, pp. 13-22, 2011

[19] Youssef Bassil, A Generation-based Text Steganography Method using SQL Queries, International Journal of Computer Applications, 57(12): 27-31, November 2012

[20] Yang Bin, Sun Xingming,Xiang Lingyun,Ruan Zhiqiang, and Wu Ruizhen, Steganography in Ms Excel Document using Text-rotation Technique, Information Technology Journal, 10(4): 889-893, 2011.

[21] Rajesh Kumar Tiwari, Microsoft Excel File: A Steganographic Carrier File, International Journal of Digital Crime and Forensics, 3(1): 37-52, January-March 2011.

[22] S. Changder and N. C. Debnath, A new approach for steganography in Bengali text, Journal of Computational Methods in Sciences and Engineering archive, 9(1): 111-122, April 2009.

[23] S. Changder, D. Ghosh, and N. C. Debnath, Linguistic approach for text steganography through Indian text, 2nd International Conference on Computer Technology and Development (ICCTD), Nov. 2010, pp. 318 – 322

[24] Indradip Banerjee, Souvik Bhattacharyya, and Gautam Sanyal, Text Steganography through Quantum Approach, 6th International Conference on Information Processing, ICIP 2012, Bangalore, India, August, 2012, pp 632-643

[25] Souvik Bhattacharyya, Indradeep Banerjee, and Gautam Sanyal, Bengali Steganography using CALP with a Novel Bengali Word Processor, INFOCOMP, v. 10, no. 4, p. 40-56, December of 2011.

[26] S. Changder, D. Ghosh, and N. C. Debnath, LCS based Text Steganography through Indian Languages, 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), July, 2010, Volume: 8, pp. 53 – 57.