Text Steganography: using Nonlinear Word Positions (NWP)

Sabyasachi Samanta[1], Saurabh Dutta[2], Goutam Sanyal[3]

Haldia Institute of Technology, Haldia, WB, INDIA[1]
E-mail id: sabyasachi.smnt@gmail.com

Dr. B. C. Roy Engineering College, Durgapur, WB, INDIA[2]
E-mail id: saurabh.dutta@bcrec.org

National Institute of Technology, Durgapur, WB, INDIA[3]
E-mail id: nitgsanyal@gmail.com

Abstract-

Usually, the steganographic algorithms employ images, audio, video or text files as the medium to ensure hidden exchange of information between multiple contenders and to protect the data from the prying eyes. This paper presents a survey of text steganography method used for hiding secret information inside some cover text. Here the text steganography algorithms based on modification of word style et cetera, has advantages of great capacity, good imperceptibility and wide application range. The nonlinear word positions of different pages are targeted through out the cover text with insignificant modification. As compared to other methods, we believe that the approaches proposed convey superior randomness and thus support higher security.

Key words: Text steganography, Nonlinear word position (NWP), Security, Data hiding

I. INTRODUCTION

Steganography is derived from the Greek word *steganos* which literally means "Covered" and *graphy* means "Writing", i.e. covered writing. Steganography refers to the science of "invisible" communication. Digital form of media as a cover-object being use in steganography are images, video clips, music or sounds. Text steganography also have been used since 2000 bc as a cover media. Text steganography is the most difficult kind of steganography, due largely to the relative lack of redundant information in a text file as compared to image or sound. Recently there have been several successful attempts to design text steganographic schemes for English, Japanese, Korean, Chinese, Thailand, Persian, and Arabic [1] [5] [10] [11].

Here we have proposed a new method to hide information in any word instead of pointed ones only. We have pointed words all through the text in a number of pages nonlinearly. First, we have taken a large text document with a number of pages. Let, all the lines contains almost same number of words in every line what we done to align the text both the left and right margins except the last line of any paragraph. This creates a clean look along the left to right side of the page. After that the message is taken. Initially, the string length is calculated. The corresponding 8-bit data for length is positioned into array. The characters are converted into its 8-bits data using ASCII-8. Here we have taken 2-bit at a time to hide information changing style on selected words of cover text of selected page. So, calculate the number of word positions we have to strike. The cover text is taken as normal text with Times New Roman font and size of 10. Here we have taken four different styles to hide the data with in text. We have used the styles like Bold, Italic and Underline. For two data bits may occur at 4 different orders starting from 00 to 11 i.e. 4 different combination of style of cover text can represent 8-bit embedding. The presence of 0 in LSB (Least Significant Bit) is as word starting with vowel and 1 as word starting with consonant. The presence of 0 in MSB (Most Significant Bit) is as word with odd number of letters and 1 as word with even number of letters. Here two sections are taken in Table 1.1.First two MSB columns for array data bits and

remaining two LSB columns for targeted word style. Depending on the data bits from array and selected word the font styles are included with Bold, Italic and Underline. For Example, if any two array data bits are of 11 and targeted word starting with vowel and even numbers of letters, the corresponding word will be altered to Arial Narrow and Bold.

Here no predetermined word is taken in our algorithm. The word position is calculated using the key. From the key, the exponential values are calculated. From the exponential value the word position, line number and page number are calculated. Then taking 2-bit data from the array the corresponding approach is chosen using the Table 1.1.

Table 1.1: Encryption Table

Array Data Bits		**Targeted Word**		**Style**	
		Word size odd / even	**Word Starting with vowel / Consonant**		
0	0	0	0	Times New Roman(No Change)	No Change
0	0	0	1		Bold
0	0	1	0		Italic
0	0	1	1		Underline
0	1	0	0	Arial	No Change
0	1	1	1		Bold
0	1	1	0		Italic
0	1	1	1		Underline
1	0	0	0	Cambria	No Change
1	0	0	1		Bold
1	0	1	0		Italic
1	0	1	1		Underline
1	1	0	0	Arial Narrow	No Change
1	1	0	1		Bold
1	1	1	0		Italic
1	1	1	1		Underline

If any exponential value hits the space between two words of the text document then the next word position is calculated and corresponding style is implemented. Or if it hits any line contains less number of words then the specified number of characters then the next line is taken to change. If that word is already changed then the next word is taken and earlier method is applied [6] [7] [8] [12].

Section 2 represents the related work. Section 3 represents the scheme followed in the encryption technique. Section 4 represents an implementation of the technique. Section 5 gives you an idea about the experimental results. Section 6 is an analytical discussion on the technique. Section 7 draws a conclusion.

II. RELATED WORKS

There are so many techniques for hiding information with in text. As example, five methods are represented in this section.

A) HTML Documents:

Secret information can be hidden within HTML Tags as they are case insensitive. For example, the tag
 can be also used as
 and
 and the tag <p align="center"> as <p align="cenTER">, as <p align="Center"> and as <p aLigN="center">, are all equally applicable. Extraction of information can be easily done by comparing these tag words with the tag words in normal case [3].

B) Line and Word Shifting Strategy:

Shifting text lines vertically and shifting words horizontally may help to hide some information with in cover text. Varying of distance between lines and words may puzzle the viewers. Shifting the lines up or down slightly with a fixed space (say 0.003 inch) and modifies the distances between the words, intended to hide the information. However, using this method, there is great possibility for the hidden information to be destroyed. Also, at the time of using character recognition programs, such as OCR, the data become lost or cannot be traced accurately [2].

C) Approach based on curves in a character:

In this approach, English letters are divided into two groups based on the shape i.e. whether a character has a curvature in its shape or not. Characters like ‘B’, ‘C’ have rounded shape where as the letters ‘A’, ‘E’ etc. does not have so. Letters with full/partial curvature hides 0 and without any sort of curvature hides 1 [9].

D) *Shifting letter points and extensions:*

Both Arabic and English languages have points in their letters and the number of pointed letters differs 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. That letters are utilized for steganography and information security. By changing the point location within the pointed letters information hiding is achievable [4].

Text steganography change some of the features of the text characters. Text steganography can hold a large quantity of secret information without making ordinary readers aware of the existence of such information in the text.

III. THE SCHEME

This section represents a description of the actual scheme used during "Text Steganography: using Nonlinear Word Positions (NWP)" technique. Section 3.1 describes the encryption technique using three algorithms 3.1.1, 3.1.2 & 3.1.3 while section 3.2 describes the decryption technique using algorithm 3.2.1 [2] [3] [6].

3.1 Encryption of message bits about the cover text

3.1.1 Create an array from message data

Step I: Take input from keyboard or special characters and compute the length (chlen).

Step II: Convert the length (chlen) into its 8-bit binary equivalent. Store that data bits to earr[bit] as LSB (Least Significant Bit) to earr[1] and MSB (Most Significant Bit) to earr[8] respectively.

Step III: Convert each character to 8-bit (using ASCII-8) binary equivalent and store to earr [ ] as LSB to earr[1+(i*8)] and MSB to earr[8+i*8].

Step IV: Repeat *Step III for* i=0 to (N-1).

Step V: Stop.

3.1.2 Selection of NCP using key

Step I: Calculate number of characters (p) to attack as 4-bit is taken at a time. So p= (bit /2).

*Step II***:** Take the key (K) and calculate the exponential value using

$E = K^p$ [i.e. pow (k, p)].

Step III: Store the exponential long double values into file one by one.

Step IV: Repeat Step II to Step III for i= (1 to p) and go to next step.

Step V: Read the values as character up to "e" of the every line of the file and store it to another file with out taking the point [.].

Step VI: Modify the value as numeric and store it to an array arrxyz[p].

Step VII: Take most three significant digit to arrx[p], next three digits to array arry[p] and least significant digit to arrz[p].

Step VIII: Repeat Step V to Step VII up to end of the file.

Step IX: Stop.

3.1.3 Replacement of array elements about the cover text

*Step I: S*elect the word position (cp) =mod[*arrx* mod *nowl*], line number (ln) =[*arry* mod *linp*] and page number (pn) =[arrz mod *plmt*].

Step II: Taking 2-bit at a time from the array (earr[]) select the approach from the *Table-1.1*. Apply the corresponding style on the selected character.

Step III: Repeat *Step II to Step VI* for i=1 to p.

Step IV: Stop.

3.2 Decryption of the data bits from the image

3.2.1 Regain of replaced message from the stego-text

Step I: To get the character position, line and page number from the stego-text go through *Step I* to *Step IV* of *Algorithm 3.1.2* and *Step I* to *Step III* of *Algorithm 3.1.3*.

Step II: By comparing it with the normal case, collect the data bits from the corresponding characters and store it to darr[]respectively.

Step III: Calculate length taking darrlen [1] as LSB and darrlen [8] as MSB (chlen) of message.

Step IV: Taking data values from the decrypted array darr[ ], LSB as darr[8*i+1] and MSB as darr[8*(i+1))] respectively, convert to its equivalent ASCII-8 character. And store the character to an array msg[len].

Step V: Repeat *Step IV* for i=3 to p.

Step VI: Finally place the characters one by one from the array msg[len] and assemble the original message.

Step VII: Stop.

IV. AN IMPLEMENTATION

Let the message to be encrypt is "India is great".

=12(Decimal equivalent)

=00001100(8 Bit Binary equivalent).

The array size will be= (8+ (12x8)).

Number of words required = (100/2) =50.

Let the size of text matrix is 56 x 16.

It signify that, at least 16 words in every line (horizontal direction) and 56 lines in every page (vertical direction) using Text size=10, Font=Times New Roman and Microsoft Office

2007. Page lay out as Normal, Top: 1", Bottom 1", Left 1"and Right 1".
Let the key = 6359.

Using the key we get the nonlinear character position as in Table 4.1.

Table 4.1: Nonlinear character position using key

Key, i	Exponential Value	Word Position	Page No.	Array Data to Replace
6359, 1	6.359000e+3	(19,04)	1	earr[1] : earr[2]
:	:	:	:	:
6359, 25	1.215403e+128	(09,12)	4	earr[99] : earr[100]

V. EXPERIMENTAL RESULT

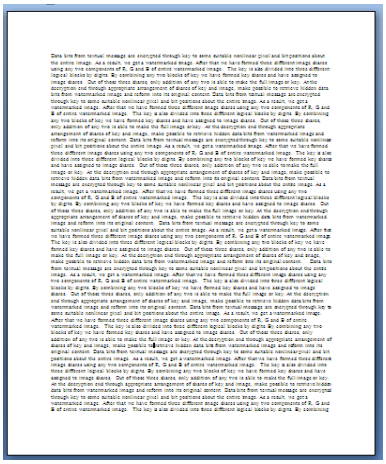

Fig 4.1: Cover Text

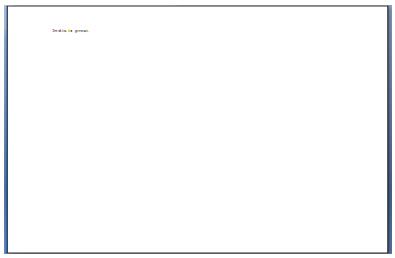

Fig. 4.2: Secret text

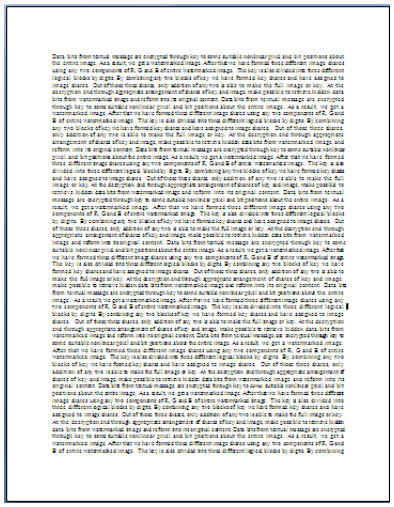

Fig. 4.3: Stego-text

VI. ANALYSIS

This method satisfies both the security aspects and the hiding capacity necessities. We have simulated the proposed system and the results are shown in the figures 5.1, 5.2 and 5.3 respectively. It generates the stego-text with least degradation of cover text and which is not very informative to people about the existence of any hidden data. This method is capable to hide 2-bit at a time through each and every word in the cover text, which reflects the high embedding capacity of the system. Also this method uses unlike character positions of unlike pages which reflects the high robustness of the system. Anybody may take more bits at a time including more number of pages and more styles. Also dissimilar styles may use for dissimilar communications. This method is also capable of checking the authenticity of the secret message send by the sender to the receiver.

VII. CONCLUSION

In this paper, we have proposed a new text steganography technique using Indian language. To do so first we have located nonlinear character position in dissimilar page order, which are calculated through a private key. After that we have encoded the data bits from message by changing the style of selected characters throughout the cover text. For extracting the message we have applied the reverse method using the key. This method featured all the needed aspects of steganography that makes it useful in hidden exchange of information through text documents. This steganography technique is also useful to any other languages like Japanese, Korean, Arabic etc.

REFERENCES

[1] M. Grace Vennice, Prof. Tv. Rao, M. Swapna, Prof. J. Sasi kiran "Hiding the Text Information using Stegnography ", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol. 2, Issue 1, Jan-Feb 2012, pp.126-131

[2] Herman Kabetta, B. Yudi Dwiandiyanta, Suyoto, "Information Hiding in CSS: A Secure Scheme Text-Steganography using Public Key Cryptosystem", International Journal on Cryptography and Information Security(IJCIS),Vol.1, No.1,December 2011, pp 13-22

[3] Debnath Bhattacharyya, Poulami Das, Samir Kumar Bandyopadhyay and Tai-hoon Kim "Text Steganography: A Novel Approach", International Journal of Advanced Science and Technology Vol. 3, February, 2009, pp. 79 - 86

[4] Adnan Abdul-Aziz Gutub, Manal Mohammad Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", World Academy of Science, Engineering and Technology 27 2007, pp. 28-32

[5] Md. Khairullah, "A Novel Text Steganography System in Cricket Match Scorecard", International Journal of Computer Applications (0975 – 8887) Volume 21– No.9, May 2011, pp. 43-47

[6] Souvik Bhattacharyya , Indradip Banerjee, Gautam Sanyal , "A Novel Approach of Secure Text Based Steganography Model using Word Mapping Method(WMM)" International Journal of Computer and Information Engineering 4:2 2010, pp. 96-103

[7] Sabyasachi Samanta, Saurabh Dutta, Goutam Sanyal, "An Enhancement of Security on Image Applying Asymmetric Key Algorithm", International Journal of Computer Applications (0975 – 8887), Volume 25– No.5, July 2011, pp. 19-23

[8] Atif Bin Mansoor, Zohaib Khan, Shoab Ahmed Khan, " CRYPTO-STEG: A Hybrid Cryptology – Steganography Approach for Improved Data Security", Mehran University Research Journal of Engineering & Technology, Volume 31, No. 2, April, 2012 [ISSN 0254-7821], pp. 219-226

[9] Shraddha Dulera, Devesh Jinwala, Aroop Dasgupta, "Experimenting with the Novel Approaches in Text Steganography", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011 pp. 213-225

[10] V. Lokeswara Reddy, Dr. A. Subramanyam, Dr. P. Chenna Reddy, "Implementation of LSB Steganography and its Evaluation for Various File Formats", International Journal Advanced Networking and Applications Volume: 02, Issue: 05, Pages: 868-872 (2011

[11] Dr. Ekta Walia, Payal Jain, An Analysis of LSB & DCT based Steganography", Global Journal of Computer Science and Technology, Vol. 10, Issue 1 (Ver 1.0), April 2010, pp. 4-8

[12] Tu-Thach Quach, "Optimal Cover Estimation Methods and Steganographic Payload Location", IEEE Transactions On Information Forensics And Security, Vol. 6, No. 4, December 2011, pp. 1214-1222

[13] Rongyue Zhang, Vasiliy Sachnev, Magnus Bakke Botnan, Hyoung Joong Kim, Jun Heo, "An Efficient Embedder for BCH Coding for Steganography", IEEE Transactions on Information Theory, Vol. 58, No. 12, December 2012, pp.7272-7279

[14] Chunfang Yang, Fenlin Liu, Xiangyang Luo, And Ying Zeng, "Pixel Group Trace Model-Based Quantitative Steganalysis for Multiple Least-Significant Bits Steganography" IEEE Transactions on Information Forensics and Security, Vol. 8, No. 1, January 2013, pp. 216-228

Sabyasachi Samanta is working as Assistant Professor at Dept. of IT, Haldia Institute of Technology Haldia, WB, and India. He has received M. Tech Degree in IT and currently pursuing Ph. D at National Institute of Technology, Durgapur, WB, India. His main research interest includes watermarking, steganography and cryptography.

Saurabh Dutta is a professor in Dr. B. C. Roy Engineering College. He holds a Ph. D Degree in Coputer Science. His research domain is information security and cryptology.

Gautam Sanyal is a member of the IEEE. He has received his B.E and M. Tech degree from National Institute of Technology (NIT), Durgapur, India. He has received Ph.D. (Engg.) from Jadavpur University, Kolkata, India, in the area of Robot Vision.

He possesses an experience of more than 25 years in the field of teaching and research. He has published nearly 68 papers in International and National Journals / Conferences. Three Ph. Ds (Engg.) have already been awarded under his guidance. At present he is guiding six Ph. Ds scholars in the field of steganography, Cellular Network, High Performance Computing and Computer Vision. He has guided over 10 PG and 100 UG thesis. His research interests include Natural Language Processing, Stochastic modeling of network traffic, High Performance Computing, Computer Vision. He is presently working as a Professor in the department of Computer Science and Engineering and also holding the post of Dean (Students' Welfare) at National Institute of Technology, Durgapur, India.