Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 6 (2011) pp. 478–482

A New Approach to Hindi Text Steganography using Hindi Karak Kriyaye

Kalavathi Alla[1], R. Siva Rama Prasad [2] and Rangaswami Reddy Kandula[3]

[1] Vasireddy Venkatadri Institute of Technology, Nambur,
Guntur Dt,Andhra Pradesh, 522508, INDIA
Kalavathi_alla@yahoo.com

[2] Acharya Nagarjuna University, Nagarjuna Nagar,
Guntur, Andhra Pradesh, INDIA
raminenisivaram@yahoo.co.in

[3] BSNL,Guntur,
Andhra Pradesh, INDIA
rsrkandula@gmail.com

Abstract: **Steganography is a communication through covert channels, is highly desirable when the more existence of an encrypted message might provide useful information for eavesdroppers. Text is ideal for steganography due to its ubiquity. However text steganography scheme do not necessarily provide a sufficient redundancy for covert communication. Here we are proposing a new Hindi natural language linguistic steganography based on karak kriya (vibhakti) classification. Hindi karak kryaye are classified using a four level classification. One can easily hide secret information in Hindi text wherever the user finds karak kriyaye in the Hindi text. The scheme is blind, so that the original carrier message is not require for decoding or during the transmission in the communication network. This method is a practical and effective method for covert communication over Hindi plain text channels.**

Keywords: Information Security, Cryptography, Hindi text, Karak kriyaye, Vibhakti, Covert channel, Text Steganography

I. Introduction

Today, huge amount of information is exchanged over the Internet in the form of text. For protecting copyright text, watermarking technologies are concerned by researchers. In order to embed invisible watermarking mechanisms in plain text Mikhail J. Atallah et al [1] put forward the concept of Natural Language Watermarking. Text steganography, in which the goal is to embed seret messages in plain text files, is among most challenging types of steganography. One reason text steganography is difficult is that text contains little redundancy compared to other media. Another is that humans are sensitive to abnormal-looking text. Because the grammatical and orthographic characteristics of every language is different, text steganogrpahy schemes must be specifically designed to exploit the specific characteristics of target language. There have been several successful attempts to design text steganography schemes, for English [2,3], Japanese [4], Korean [5], Chinese [6], Persian and Arabic [7], Thai [8,9,10], Hindi [11,12].

In this paper we introduce a new stegnaographic scheme for Hindi plain text files. The scheme classifies various karak kriyaye which are used in the construction of sentences in Hindi language. We found that this schemes's effect on carrier text is unnoticeable to human observers.

II. Background : Hindi

Hindi is the National language of India. Standard Hindi is also known as Manak Hindi, Nagari Hindi, High Hindi and Literary Hindi. It is one of the 22 official languages used in India and is used as a primary official and National language of the Republic of India. The constitution of India, adopted Hindi as a National Language in 1950 and declares Hindi in the Devanagari Script as the official language of the Union. The Constitution of India has specified the usage of Hindi and English to be the two languages of communication for the central government. The spoken Hindi dialects are formed using an extensive dialect continuum of the Indic language family. The combined population of Hindi and Urdu speakers is the fourth largest in the world. According to 2011 Indian census, 551 million people in all over the world speak Hindi language. Hindi is an official language in the following states of India: Arunachal Pradesh, Bihar, Jarkhand, Uttarakhand, Madhya Pradesh, Rajasthan, Uttar Pradesh, Chattisgarh, Himachal Pradesh, Haryana and Delhi. Similarly Hindi is considered as a co official language in several states. Hindi language has a standardized and sanskritized record of the Hindustani language derived from the Khariboli dialect of Delhi and the surrounding western Uttar Pradesh and Southern Uttarakhand region. After independence, the Government of India passed a rule to standardize Hindi as a separate language by associating the following conventions : First and foremost

language grammar must be standardized; second is to change the orthography of language characters by improving the shape of Devanagari Characters, and by introducing diacritics to express sounds from other languages; third and the last convention is to standardize vocabulary, replacing most of the so far used Persian words with new coinages from Sanskrit. Its alphabet set is similar to Devanagari script. This kind of alphabetic set is similar to many Indian languages, including Sanskrit, Hindi, Telugu, Marathi and many more. Although Sanskrit is an ancient language and is no longer spoken, but written material still exists. Hindi is a direct descendant of Sanskrit through Prakrit and Apabhramsa. It is a very expressive language, which has been influenced and enriched by Dravidan, Turkish, Farsi, Arabic, Portugese and English. Standard Hindi derives much of its formal and technical vocabulary from Sanskrit. Standard or shuddh(pure) hindi is used only in public addresses and radio or TV news. There are five principal categories of words in standard Hindi: they are tatsam, ardhatatsam, tadbhav, deshaj, and videshi, Tatsam words are directly spelled the same as spelled in Sanskrit. They include words inherited from Sanskrit via Prakrit which was survived without modification (e.g. Hindustani nam/Sanskrit nama and prarthana). Pronunciation, however, conforms to Hindi norms and differs from classical Sanskrit. Among nouns, the tatsam words could be the Sanskrit uninflected word-stem. Ardhatatsam words were directly borrowed from Sanskrit in the middle Indo-Aryan stage, and these words have undergone some sound changes before borrowing [13]. Tadbhav words are spelled differently from Sanskrit and derived from a Sanskrit prototype by phonological rules(e.g. Sanskrit karma, “deed” becomes Pali Kamma and similarly Hindi kam, “work”). Deshaj words were not borrowings and do not derived from Indo-Aryan words. They belong to onomatopoetic words. Videshi words are borrowed from other than Indo-Aryan sources like Perisan, Arabic, Portuguese, and English. Hindi is the world’s third most commonly used language after Chinese and English, and there are approximately 551 million people all over the world that speak and write in Hindi. Thus, the research on Hindi Text steganography attracts a lot of interest.

III. Related works

Hindi text steganography can be broadly classified into two categories: feature coding and linguistic based methods.

A. Feature Coding

Feature coding methods use the language orthographic properties. By slightly modifying Hindi orthographic nature of characters information can be hidden. By using these methods information can be hidden in the form of text images. Once we convert the text image into editable format, they loss their originality. Feature coding can be applied to almost all the languages based on the natural language orthographic characteristics. Hindi characters can be written between top base line and bottom base line. Some characters have modifies which fall above the top line and are known as top modifiers and some of the characters have modifiers which fall below the base line and are known as bottom modifiers. By shifting the original position of top modifiers and bottom modifiers one can easily hide secret information in Hindi text. If there is a change in the position of the modifier, bit ‘0’ can be embedded with this, otherwise bit ‘0’ can be embedded. Once we modify the cover text(Hindi text) based on the input secret message, we can convert this file into image. And this stego file is transmitted in the communication network. At the receiver side, after receiving the stego file, they have to measure the originality of orthographic feature. But if we try to convert this file into editable format, then the concealment of information is lost.

B. Linguistic Methods

These methods are prepared based on the linguistic properties of natural language. Each language is equipped with its own distinguished properties. Some of the methods are listed below which are implemented on the above characteristic. Kalavathi et al proposed 5 different methods to hide secret information using Hindi text steganography.

1) Hindi Text Steganography using HHK Scheme

It is a Hindi Hexadecimal Katpayadi Scheme. Katpayadi scheme is the ancient security method which is used to hide integer values inside the Devanagari characters. This kind of mechanisms were mostly used in Vedic Mathematics . The approximate value of PI can be stored in upto 32 decimal places. It is higly impossible for a human to remember such a long value. Therefore these values are stored in Vedic Hymns by following the Devanagari character encoding scheme. Authors have extended this scheme by giving 15 different codes to Hindi vowels and consonant characters. Therefore, each four bit information of secret text can be easily hidden in a Hindi cover text. It makes a perfect covert channel and it is unsuspicious to the third persons or eavesdroppers. Because this cover message is a meaningful Hindi text [12]. Now this embedded cover text acts as a stego message and is transmitted across the network in the communication channel.

2) Hindi Text Steganography using Matraye

Hindi matras are sings in the Devanagari written script. There are totally 57 symbols in the Hindi language, and matras are a sort of short way to connect vowels and consonants. A vowel is pronounces alone but a vowel sound (matra) is pronounced together with a consonant. Each vowel is represented by its sign (matra). In Hindi vowels have two forms : an independent character is used when a vowel comes in a beginning of the word and a dependent sign (matra) is used when a vowel follows a consonant [11].

Regular Hindi letters can typically be divided into three strips: top, core and bottom. For example, the Hindi Word अकुलीन is a five character word . It has three strips. The header line always separates the top strip and core strip, while there is no corresponding feature to separate the bottom strip and core strip. The top strip contains the top modifiers, and bottom strip contains lower modifiers. In a Hindi word, top and bottom strip are not always necessary, but depend on top and lower modifiers. Hindi Unicode system has a unique code for each kind of pattern. Therefore by classifying them in two different categories like top modifiers and bottom modifiers, this algorithm can be implemented. Authors have encode the top modifiers with a bit ‘1’ bit and bottom modifiers with a ‘0’ bit .

3) Hindi Text Steganography using Bar Characters

A vertical bar does not appear at the left end of a Hindi Character. If a vertical bar is present, it either appears at the right end (End Bar) or in the middle (Middle Bar) of a Hindi Character. Based on the presence and it's position of vertical bar and the conjunction number of character with the header line, all the core Hindi characters can be divided into the following six groups: Open Header, One Conjunction End Bar, More Conjunction End Bar, Middle Bar, No Bar and Special case. Out of these 6 classifications, we can combine One Conjunction End Bar, More Conjunction End Bar and Middle Bar Characters under a single category that is Bar Characters. Categorization is shown in the Table 1. The last three characters in the special case with an end bar, but after removing the header line and computing the vertical projection, each of these four characters will be split into two parts. Therefore these characters are treated as special case characters [12].

Open	Open Header	अ थ ध भ
Bar Characters	One Conjunction	च ज ञ त न व
	More	ख घ झ प म य ष स
	Middle Bar	ऋ क फ
No Bar	No Bar	इ ई उ ऊ ए ऐ ङ ट ठ
Special Case	Special Case	ग ण श

Table 1. List of bar characters

4) Hindi Text Steganography using Bar Characters

Every language has its own characteristics and has two types of letters: vowels and consonants. Letter may be a pure vowel or pure consonant. A consonant word can be modified by using matraye or it may be combined with another consonant letter that is a compound letter. The letters which are from pure vowel and consonants fall in the category letters and the letters which are formed with matraye or with any other consonants fall under the category diacritics. Mostly All Indian Languages are built with similar vowels and Consonant structures. All these characters are universally recognized by the Unicode character set. The method that we are proposing here for Hindi Language, is perfectly suits with the other Indian Languages like Telugu, Sanskrit, Marathi, Gujarati and for many more. Using this two level classification, one can hide secret information in Hindi Text which acts as a cover text in the transmission. Generally information is transferred in the form of binary encoding. This may be intercepted by any one in the web. To prevent this attack, we can apply text steganographic method to hide information. That is instead of transmitting information in the binary version of English text, we can hide these bit streams using the assigned Unicode values of Hindi Letters and Letter diacritics [11]. According to the present proposed algorithm a pure letter(vowel or consonant) is encoded with a bit '0' and letter diacritics or compound letters are encoded with a bit '1'. By hiding information in this way, we can achieve our primary goal that is to make a perfect covert channel between peers. This algorithm has some sort of complexity in generating the cover text. According to the properties of security algorithms, the more complex is the more secure.

IV. Proposed Methods

A. Hindi Text Steganography using Karak Kriyaye

The syntactic and semantic functions of noun phrases are expressed by case-suffixes, postpositions and derivational processes. There are two cases: direct and oblique. Case-suffixes and post positions are used to express syntactic and semantic functions. Case suffixes are defined as bound suffixes which do not occur independently as words and are added only to the noun phrases. Case suffixes added to the oblique forms of nouns agreeing in number and gender. Case-suffixes followed by postpositions indicate various relationships between the noun phrases and the verb phrases.

Postpositions have specific semantic functions. They express semantic dimensions of a noun such as benefaction, manner, or location. The main postpositions are: '**ने**' *ne* 'ergative marker'; '**को**' *ko* 'to'; '**केलिये**' *keliye* 'for'; '**पर**' *par* 'on'; '**मे**' *me* 'in'; '**से**' *se* 'from', 'with'; '**का/के/की**' *ka/ke/kee* 'of'. The postpositions are written as separate words.

Ex: **लोहित ने , लोहिता की, लोहित के**

Based on Hindi linguistic properties, karak kriyaye(vibhakti) is classified in the following way :

S.No	Type	English Version	Symbol	In English
1	Karta karak	Nominative	**ने**	ne
2	Karm karak	Objective, accusative	**को**	ko
3	Kakan karak	Instrumentative	**से,के द्वारा**	se, ke dwara
4	Sampradan karak	Dative	**को, केलिये**	ko, keliye
5	Apadan karak	Ablative	**से**	se,
6	Sambandh	Genitive, Possessive	**का, के, की**	ka, ki, ke
7	Adhikaram karak	Locative	**मे, पर**	me, par
8	Sambodhan	Vocative	**हे, अरे**	hey!, array!

Table 2. Hindi Karak Kriyaye classification

These can be further simplified as :

S.No	Type	Symbol	Code
1	Karta karak	ने	00
2	Karm, sampradan, apadan karak	को	01
3	Karan, adhikaran karak	से, मे, पर	10
4	Sambandh karak	का, के, की	11

Table 3. Karak kriya encoding

Message Encoding Algorithm :

Step 1 : Read the plain text message as plain_text that is to be transmitted in the network.
Step 2 : Consider n=length of plain_text
Step 3 : loop i=0 to n
i) Convert each character of the plain_text to its equivalent binary store in ascii_bin_string
Step 4: End.

Message Embedding Algorithm :

Step 1 : Read the ascii_bin_string;
Step 2 : Find the length of ascii_bin_string and store in ascii_bin_length;
Step 3 : Loop j=0 to ascii_bin_length
a) Select the cover text to embed the message. Check whether the selected text is capable of embedding the ascii_bin_string using the encoding shown in Table . If not possible repeat this step. Otherwise continue.
Step 4 : Send the embedded text as stego_text in the communication channel.
Step 5 : End.

Message Extraction Algorithm

Step 1 : Read the stego_text received from the sender.
Step 2 : Convert the string into various tokens based on white space character as delimiter using string tokenizer function and store each token is string s1.
Step 3 : Initialize plain_bin_string, retrieved_string as a null string;
Step 4: For each token of the string do the following steps
i) Find n=length of token string s1
ii) If the length ==2 then find the Unicode equivalent of first character and store in unicode_value.
a) If unicode_value ==ne[i] then append '00' to plain_bin_string;
b) Else If unicode_value ==ko[i] then append '01' to plain_bin_string;
c) Else If unicode_value == semepar[i] then append '10' to palin_bin_string;
d) Else If unicode_value == kakeki[i] then append '11' to plain_bin_string.
iii) Else go to step 4
Step 5 : Find n1=length of the plain_bin_string.
Step 6 : loop j=0 to n1
i) Find the substring of plain_bin_string of length 8 and store in bin_sub_string;
ii) Find the equivalent ascii character of bin_sub_string ;
iii)Append each character to retrieved_text;
Step 7 : Print the retrieved_text as the plain text decrypted from the method.
Step 8 : End.

Ex:
Plain Text : hide
Ascii binary string : 01101000011010010110010001100101
Cover Text :

01 10 10
भगवान राम को भारत देश मे हर एक गॉव मे मंदिर है ।
00 01
भगवान राम ने बहुत अच्छे काम किये । राम को तीन भाइयाँ है ।
10 10
तीन भाइयों मे लक्ष्मण हर वक्त राम से मिलझुलकर रहता था।
01 01
एक बार राम को वनवास जाना पडा । राम सीता को अयोध्या
10 01
मे छोडकर जाना चाहता था । किंतु सीता को यह पसंद नही थी।
00 01 10 01 11
सीता ने भोगभाग्यों को छोडकर वनवास मे राम को सपर्य की ।
01 11
एक दिन सीता को दुष्ट रावण बंदीकर के ले गया ।

Furthermore, to strengthen this algorithm can be applied on the cipher text of plain text. Plain text can be taken as an input into the encryption algorithm, and then the cipher text can be taken as an input to this text steganography algorithm. At the receiver side, receiver has to extract cipher text from the stego text and then applied decryption algorithm to decode the original plain text message.

V. Experimental Results

The proposed method is implemented using Java with the help of Unicode support system. The interface has 3 methods which works both at the sender side and receiver side. Three buttons working procedure is explained above. This algorithm is also tested with different kinds of text messages, and its total number of bits required. It varies from content to content since text messages may be constructed in different ways. These statistics may not be equal for all types of messages.

To determine the embedding capacity achievable with out algorithm, we collected different Hindi text documents and tested. On average, the number of embeddable secret message bits was 0.4% of the original carrier (stego) text size, and per sentence embedding capacity is 0.4%

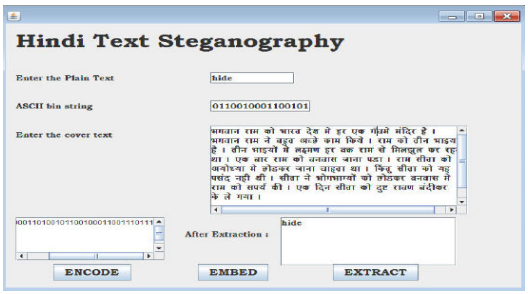

Figure1 Hindi Text Steganography Karak

Total number of secret Message Bits	Stego text size	Total no of bits used	Embedding capacity of secret messages based on original	Per line embedding capability of secret message
32	7.26	512 bits	0.43 %	0.4 %
256	70	8	0.35 %	0.43 %

Table 4. Embedding capability of stego text

VI. Conclusion

This paper proposed a new blind steganographic scheme for Hindi text that exploits redundancies in the form of karak kriyaye. It was observed that the modifications made when information bits are embedded in the carrier text are unnoticeable to intruders, and that the embedding capacity of 0.4 % makes the scheme practical and effective for covert communication. Simultaneously ensuring covertness, privacy, and authenticity will be the focus of our future work.

Acknowledgment

We are grateful to the Department of Science and Technology, Delhi, India as this work was funded by DST under Women Scientists Scheme (2011). We also thankful to the management of Vasireddy Venkatadri Institute of Technology, Guntur, AP for supporting me to do this work.

References

[1] Atallah M, McDonough, C. Nirenburg S, Raskin V. Natural language processing for Information assurance and security an overview and implementations. New Security Paradigms workshop. Ireland,2001:51-65.

[2] J.T. Brasil, S.Low, and N.F. Maxemshuk, "copyright protection for the electronic distribution of text documents, "Proceedings of the IEEE vol.87,no 7, pp,1181-1196, July 1999.

[3] D. Huang and H. Yan, "Inter word distance changes represented by sine waves for watermarking text images, "IEEE Transaction on Circuits and Systems for Video Technology, vol 11, n0 12,pp.1237-1245, December 2001.

[4] T. Amanao and D.Misaki, "A feature calibration method for watermarking of document images", in proceedings of the Fifth International Conference on Document analysis and recognition ICDAR'99, September 1999, pp.91-94.

[5] Y.W. Kim and I.S, Oh, "Water marking test document images using edge direction histograms," pattern recognition letters, vol 25, no. 11, pp.1243-1251, August 2004.

[6] W. Zhang , Z. Zeng, G. Pu, and H. Zhu, "Chinese text watermarking based on occlusive components", The second Information and communication Technology ICTA'06, vol 1, pp. 1850-1854, April 2006.

[7] M.H. Shirali-Shahreza "A new approach to Persian/Arabic text steganography," in proceedings of the International Conference on Computer and Information Science July 2006, pp.310-315.

[8] T. Karoonboonyan, "Standardization and implementation of Thai language", in National Electronics and Computer Technology Center, Bangkok, 1999.

[9] T. koanantakool, 'The keyboard layouts and input method of the Thai language", in Information processing institute for education and development Thammasat Universty, Bangkok, 1991.

[10] Nathawat Samphaiboon and Matthew N. Dailey, "Stegangoraphy in Thai Text", in proceedings of the ECTI-Con2008, IEEE Digital Library pp. 133-136.

[11] Kalavathi.Alla and Dr. R. Sivarama Prasad, "An Evolution of Hindi Text steganography" 6th International Conference on Information Technology: New Generations ITNG 2009, proceedings of the conference in IEEE computer society digital library,1577-1579. doi.ieee.computersociety.org/10.1109

[12] Kalavathi.Alla and Dr. R. Sivarama Prasad, " A new appraoch to Hindi Text Steganography using matraye, core classfication and HHK scheme". 7th International Conference on Information Technology : New Generations ITNG 2010, proceedings of the conference in IEEE digital library, doi.ieee.computersociety.org/10,1109/ITNG2010.162.

[13] http://en.wikipedia.org/wiki/Standard_Hindi

Author Biographies

Kalavathi Alla, She received her M.C.A., M.Phil degress in Computer Science and submitte Ph.D thesis in Acharya Nagarjuna University. She has 12 years of experience and working as an Associate Professor at Vasireddy Venkatadri Institute of Technology. She received a funded project from Department of Science and Technology under Women Scientist Scheme-A for the years 2011-2014. Her areas of interest are Information Security, Computer Networks, Mobile communications.

Ramineni Sivaram Prasad works as an Assistant Professor in Acharya Nagarjuna University. He published 40 research papers in National and International Journals. Attended 78 National and International seminars and presented papers on various issues. His areas of interest are Information security, E-Commerce, E-Governance and Foreign Trade & CSR.