A Capital Shape Alphabet Encoding(CASE) Based Text Steganography

Sunita Chaudhary
Department of Computer Science
Sobhasaria Engineering College, Sikar
NH - 11, Gokulpura
Sikar, Rajasthan, India
er.sunita03@gmail.com

Peeyush Mathur
Department of Computer Science
Sobhasaria Engineering College, Sikar
NH - 11, Gokulpura
Sikar, Rajasthan, India
peeyush14_mathur@yahoo.com

Tarun Kumar
Department of Computer Science
Govt. Engineering College, Bikaner
Karni Industerial Area, Pugal Raod
Bikaner, Rajasthan, India
ertarunkumar@yahoo.co.in

Richa Sharma
Department of Computer Science
Govt. Engineering College, Bikaner
Karni Industerial Area, Pugal Raod
Bikaner, Rajasthan, India
sharma.richa676@gmail.com

Abstract

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message. Steganography is a form of security through obscurity. Steganography algorithms uses cover media such as text, image, audio and video etc. to hide the data. User relies on change in the structure of these mediums and features of the target medium in such a manner as is not identifiable by human. In this paper we also present and evaluate my contribution to design the new approach for text Steganography and named it as CASE (Capital Alphabet Shape Encoding) approach. This approach is a combination of random character sequence and feature coding method.

Keywords— Steganography; features; encoding; decoding; cover text.

1. Introduction

Steganography means conceal communication. It is derived from a work by Johannes Trithemus (1462-1516) titled "Steganographia". The word Steganography comes from Greek and meaning of steganography is "concealed writing". Steganography is used to transmit a message through some innocuous carrier i.e. text, image, audio or video over a communication channel in order to effectively conceal the existence of the message. Text steganography is a process to hide the secret information within text (i.e. character based) messages. Text steganography is the most difficult kind of steganography [1]. Text steganography is considered as difficult one is due to the lack of redundant information in a text file, while there is a lot of redundancy in a picture or a sound file, which can be used in steganography [1] [2].

1. Related work

For text steganography there are many methods available [3] [4]. Some method change format of text while some method change actual word to hide secret data. White space is used to hide secret data, in open space methods [5], while in syntactic method punctuations are used to hide secret data [5]. In acronyms method [2] and semantic method [4], actual word or phrase replacement is used to hide secret data. By using characteristics of that particular language data is hidden in Persian/Arabic Text steganography [4] and Hindi Text steganography [6].

If we use open space methods [6] or feature coding method [4] in text steganography, and if somehow format of file is changed then it results in data loss. In acronyms and semantic method, meaning of information can be changed because these methods use actual word replacement or punctuation to hide secret data. So a method is needed by which secret data survive after changing format of file and meaning of text will not changed.

So considering some problems like format changing, changing meaning of secret data, etc. in existing text steganography methods; we have proposed new CASE approach for text steganography. We have used text features of English letters to hide secret data. Letters of English alphabet based on kind of round shape or curve, vertical and horizontal line are grouped in first approach. Like some letters in English language contains kind of close round shape or curve. Like some letters in English contains only one straight vertical line. In second approach, we are grouping numbers and symbols. In CASE categorization, based on different text features of English letters we categorize English letters into eight groups. We have studied the implementation some existing methods mentioned in paper [4] and we have compared these methods with proposed CASE approach. In the proposed CASE approach randomness is used but it aids to provide more security to secret information. For various existing methods and proposed CASE approach we have measured number of bytes hide, time overhead and memory overhead. Our results shows that, very less time overhead and memory overhead is required to implement proposed CASE approach compared to existing methods, and also we can hide more number of bytes using proposed approach. Required cover text size is also very small in proposed approach.

2. Proposed approach

In this approach, we introduce new encoding technique to hide the secret message in cover text. We will call this technique Capital Alphabet Shape Encoding (CASE). In this method every character of secrete message is encoded in the form of 8-bit binary number after that the equivalent ASCII character is replace the original character. In this the left most 0th bit will represent the alphabet group or digit/symbols group (0 value for alphabet set and 1 value for digit or symbols).

In case of alphabet we made eight groups of English letters based on features of letters. While making group we consider only Capital letters of English alphabet. The left most 1st, 2nd, 3rd bit of 8 bit number represents the group number. The next 4th bit will represent the sentence case of letter. If it is 1, than it represents upper case letter and if it's 0 than letter will be lower case letter. Table I, II, III shows eight bit encoding format of alphabets, digits and symbols. In this approach all alphabets are divided into groups and every alphabet has its position in corresponding group. This position will represent the last three bit of 8-bit number. By using this approach, we can hide all eight bits of one letter of secret message into one letter of cover text at a time.

Table1. 8-Bit encoding format in case of alphabet

0th bit	1st bit	2nd bit	3rd bit	4th bit	5th bit	6th bit	7th bit
Alphabet (0)	Group no.			Case	Group position in alphabet Group.		

Table2. 8-Bit encoding format in case of digit

0th bit	1st bit	2nd bit	3rd bit	4th bit	5th bit	6th bit	7th bit
digit (1)	Group Position in Digit Group				Always '0'	Always '0'	Always '0'

Table3. 8-Bit encoding format in case of symbols

0th bit	1st bit	2nd bit	3rd bit	4th bit	5th bit	6th bit	7th bit
Symbol (1)	Always '0'	Always '0'	Always '0'	Group Position in Symbol Group.			

As shown in Table1, in first group, we include neither those letters which have round shape or any curve and nor vertical and horizontal straight line. We can use any letter from this group to hide "000" bit. Candidates for this group are V, W, X, Y. In second group, we include those letters which have one or two vertical straight line. We can use any letter from this group to hide "001" bit. Candidates for this group are K, M and N. In the third group, we include those letters which have only one or more horizontal straight line. We can use any letter from this group to hide "010" bit. Candidates for this group are A and Z. In forth group, we include those letters which have both one and more than one straight vertical and horizontal line. We can use any letter from this group to hide "011" bit. Candidates for this group are E, F, H, I, L and T.

In fifth group, we include those group, we include those letters which have only curve or round shape. We can use any letter from this group to hide "100"bit. Candidates for this group are C, O, Q, S and U. In sixth group, we include those letters which have both curve and straight vertical line. We can use any letter from this group to hide "101"bit. Candidates for this group are B, D, P and R. In seventh group, we include those letters which have curve and straight horizontal line. We can use any letter from this group to hide "110" bit. Candidate for this group is G. In last but not the least eighth group, we include those letters which have curve, and both straight vertical and horizontal line. We can use any letter from this group to hide "111" bit. Candidate for this group is J. For example if the secret letter is H then by using CASE approach it will be encoded as 00111010 and its ASCII equivalent is 58 which is given by ':'. After encoding, now letter ':' will be mixed up with the cover text. And this letter will hides all the 8 bits of original letter H into cover text.

2.1. *Hiding the Message*

In the CASE text stenography approach, first we encode all the characters of the secret message with new proposed encoding technique which is based on the shape of the alphabet characters. Second we hide this message with the cover text by mixing it with the contents of cover text, i.e. first we encode the secret character and then we perform process of finding the ASCII equivalent of the 8-bit format of the secret character and hide it with the contents of cover text.

To hide it or embed it with the cover text, we made a new technique. In this technique we encode the first three letter of the cover text by using CASE approach and then count the bits having value 1. This count value is the key value for hiding the data. After calculating the key value message is mixed up, one character of message comes after key number character of cover text and this process of embedding are repeated until whole message is hidden in the cover text.

3. Implementation

For implementation of CASE approach we develop two algorithms for hiding and retrieving which are implemented in JSP using NetBeans IDE.

3.1. Pseudo code for message hiding

```
Procedure CASE_steno_hide (String msg, String covertext)
begain;
For i=0 to msg.length()
 ch=msg(i);
encode_msg=encode_msg+char(encode(ch));
End For
key_msg=covertext.substring(0,3);
For i=0 to key_msg.length()
ch=key_msg(i);
en_key=en_key+encode(ch);
```

End For

keyarray []=en_key.toByteArray();

key=0;

For i=0 to keyarray.length()

IF keyarray[i] == 1 then

key++;

End IF

End For

IF key < 5 then

key=5;

End IF

For i=0,j=0 to i <covertext.length()

hidden_msg=hidden_msg+covertext.substring(i,i+k)+enco

de_msg(j);

i=i+k+1;

j++;

End For

return hidden_msg;

End Procedure

3.2. Pseudo code for message retrive

Procedure CASE_steno_unhide(String hidden_msg)

begain

key_msg=hidden_msg.substring(0,3);

For i=0 to key_msg.length()

ch=key_msg(i);

en_key=en_key+encode(ch);

End For

keyarray []=en_key.toByteArray();

key=0;

For i=0 to keyarray.length()

IF keyarray[i] == 1 then

key++;

End IF

End For

IF key < 5 then

key=5;

End IF

For i=key to hidden_msg.length()

encode_msg=encode_msg+hidden_msg(i);

i=i+key;

End For

For i=0 to encode_msg.length()

orignal_msg=orignal_msg+decode(encode_msg(i));

End For

return orignal_msg;

End Procedure

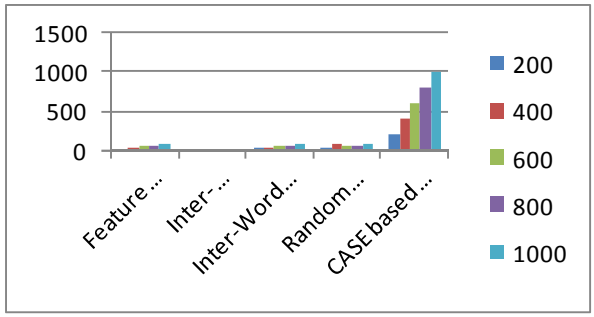

Fig. 1: Numbers of bytes hide by particular method

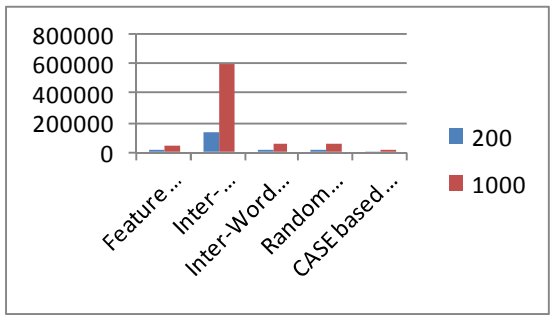

Fig. 2: Maximum cover text required to hide 200 bytes and 1000 bytes

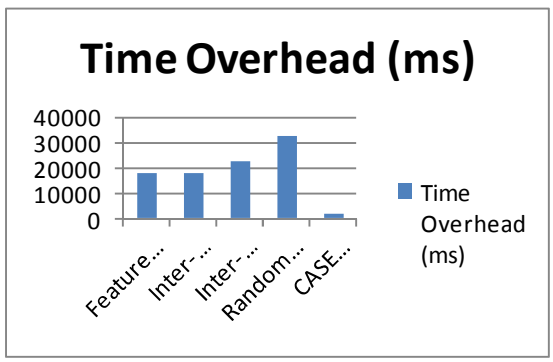

Fig. 3: Time and memory overhead of all methods.

4. Conclusion

In this paper, we have proposed new approach for text-based steganography for English language texts. In this approach, we exploit the shapes of the English characters to hide secret bits. Based on our survey of the existing Text Steganography approaches, we show that our proposed approach can hide more

number of bytes, it has very small cover text and required very less time overhead as compare to other techniques. Our analysis reveals that our approach imparts increased randomness in encoding because of which the same cannot be attacked easily. This approach is applicable to the soft-copy texts as well as hard-copy texts. In addition, the proposed approach is also immune to retyping and reformatting of text. However, one of the weaknesses of the proposed approach is that once known about their applicability, they can easily be attacked. Hence, it is essential to keep the application of a particular approach to a particular data set secret, while using them.

5. References

1. M. Shirali-Shahreza, "Text steganography by changing words spelling," In 10th International Conference on Advanced Communication Technology, Korea, 2008.
2. M. Shirali-Shahreza, and M. Shirali-Shahreza, "Text Steganography in SMS," In International Conference on Convergence Information Technology, 2007.
3. F. Khan, "Enhanced Text Steganography in SMS," In 2nd International Conference on Computer, Control and Communication, 2009.
4. M. Shirali-Shahreza, and M. Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography," In 5th IEEE/ACIS International Conference on computer and information science (ICIS COMSAk'06), 2006, 310-315.
5. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," In IBM System's journal, vol. 35 (Issues 3 &4), 1996, p.p.313-336. Available: http://www.informatik.unitrier.de/~ley/db/ journals/ibmsj/ibmsj35.html.
6. K, Alla, and Dr. R. Shivramprasad, "An evolution of Hindi text steganography," In 6th International Conference on Information Technology, 2009.
7. B. Dunbar, "A Detailed look at Steganographic techniques and their use in an Open-systems environment," SANS Institute, 2002.
8. K. Bennett, "Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text," Purdue University, CERIAS Tech. Report 2004-13, 2004.
9. A. Gutub, and M. Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions," World Academy of Science, Engineering and Technology, 2007.
10. L. Robert, and T. Shanmugapriya, "A Study on Digital Watermarking Techniques," In International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009.
11. J. Brassil, S. Low, N. Maxemchuk, and L. O'Garman, "Copyright protection for the electronic distribution of text documents," In Proceedings of the IEEE, VOL. 87, NO. 7, July 1999.
12. J. Brassil, S. Low, N. Maxemchuk, and L. O˝Gorman, "Electronic marking and identification techniques to discourage document copying Text Steganography," IEEE Journal on Selected Areas in Communications, VOL. 13, NO. 8, October 1995, p.p. 1495-1504.
13. Shraddha Dulera et.al."Experimenting with the Novel Approaches in Text Steganography" published on International Journal of Network Security & its application (IJNSA), Vol.3, No.6, November 2011, pp 213-225.