IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. VII (May – Jun. 2015), PP 70-74
www.iosrjournals.org

A Comparative Result Analysis of Text Based Steganographic Approaches

K.Aditya Kumar Prof.Suresh Pabboju
Research Scholar, Osmania University Hyderabad
Head of IT Department Chaitanya Bharathi Institute of technology Hyderabad

Abstract: *Today in this digital era everything (data) can be digitized and can be transmitted over the communication networks. But not withstanding with this advantage it also has a downside i.e. the digitized data can be easily accessed illegally, it can be tampered with and copy. The issue of providing security to the information has become increasingly important with the development of computer and expanding its use in different areas of one's life and work[4]. One of the grounds discussed in information security is the exchange of information through the cover media. In this, different methods such as cryptography, steganography, etc., have been used. Many approaches were introduced for making the data secure. But the efficient utilization of those approaches isn't defined i.e. which approach to be used according to the situations. This paper implements five approaches of Text Steganography to compare their Time and Space complexities by from considering three Text Steganography algorithms from the study of Monika Agarwal's paper entitled "Text Steganographic Approaches :A Comparison"[1] and bring out the comparative results of algorithm for some particular secret text. The results of the comparisons are shown in form of bar charts and pie charts.*

Key Words: *Steganography, Text Steganography, Missing Letter Puzzle, Hiding Data in Wordlist, Hiding Data in Paragraphs, Hiding Data in Characters, Hiding Data in Spaces*

I. Introduction

Steganography is the art and science of hiding the data or information within another data or information without changing the meaning of original data or information, hence the meaning of the Steganography comes like "Covered Writing". The data or Information which is hidden in another data or Information is available only to the recipient of the data during their exchange. The goal of Steganography is to transmit the data through a carrier, which may be a text, image, audio or video such that the existence of the message cannot be detected. Unlike Cryptography, which is used to ensure the confidentiality of the message content to be transmitted securely, which the same is achieved in steganography by adding more security by hiding in an another cover media which is not visible to the third person other than the sender and receiver. Steganography can be derived in to image ,audio, video and text steganography depending on the cover media used to hide the data or information.

II. Text Steganography

Of all the types of Steganography methods Text Steganography plays a trickiest part due to deficiency of redundant information present in other forms of Steganography methods .Text Steganography can be of different forms like hiding in spaces or in any part of the document or changing the meaning of words to shortcuts or may be changing the synonym from US to UK words or hiding in punctuation marks or may be in the source code of page etc[5][7]. In Text Steganography approach one can hide data or information by making changes in the structure of the document without making a notable changes in the concerned output[6].

III. Related Work

From the study of the paper of Monika Agarwal's paper entitled "Text Steganographic Approaches :A comparison "[1] compared three basic text Steganographic algorithms like 1.Missing letter puzzle 2.Hiding Data in Word List and 3. Hiding data in Paragraphs on different criteria's like the ability of a cover medium to hide secret information, a four kinds of measures to find distance between given two sequence of values and so on.

3.1 First Approach: Missing Letter Puzzle

This is a approach which comprises different words in which one or more letters missing, which is replaced with a symbol to indicate the letter missing[1]. This puzzle is solved y replacing the symbol with a letter.

3.2 Second Approach: Hiding Data in Wordlists

This approach hides message without considering any special symbol as previous approach[1]. In this approach a character is stored in a certain word which has a certain length. by masking sum of digits of an ASCII value of starting character of a word is deterimined.for example the starting letter is determined as 'a' if the sum is '1' and for 'b' it is '2' and this goes on.

3.3 Third Approach: Hiding Data in Paragraphs

This approach uses a pre-determined cover file which can be any English text which can be from any source. This approach hides a message by using starting and ending letter of the words in a cover file. by considering binary value of a character. After converting the cipher text to a stream of bits, each bit has to be hidden by picking a word from the cover file and using either the start or the end letter of that word depending on the bit to be concealed.

IV. Proposed Work

This paper implements five text based Steganographic approaches by considering three text Steganographic algorithms from the study of Monika Agarwal's entitled "Text Steganographic Approaches: A Comparison"[1] along with these three algorithms an additional of two algorithms are considered for comparison thereby making a comparative analysis among them by considering their time and space complexities. These comparative studies are developed in java. The five approaches which are used for comparison are: Missing Letter Puzzle, Hiding Data in Wordlist, Hiding Data in Paragraphs, Hiding Data in Characters, and Hiding Data in Spaces[7][8].

4.1 Fourth Approach: Hiding Data In Characters

This approach makes use of a pre-defined cover file which can be any meaningful piece of English text and can be drawn from any source (For example, a paragraph of a newspaper/book). The approach works by hiding message a bit in each character in the words of a cover file. This approach works on the character's binary[7]. After converting the cipher text to a stream of bits, each bit is hidden by picking a character from the cover file and using the ASCII value of the character odd number or even number is generated depending on the bit to be concealed. Bit 0 or 1 is hidden by reading a character, sequentially, from the cover file and generating even or odd number, respectively. Since no change is made to the cover, the cover file and its corresponding stego file are exactly the same.

4.1.1 Hide Algorithm

1. Get a cover file.
2. Convert the input file to its binary equivalent (bin).
3. Read a bit (x) from the bin.
4. Read a character from the cover file and get its ASCII value (n).
5. Save the character into stego file.
6. m=255-n.
7. If x=0, generate an odd number (num) within the range of m.
8. Else, generate an even number (num) within the range of m.
9. num=num+n.
10. Write character equivalent of num in the stego key.
11. Repeat steps from 2 to 10 till the end of input file.
12. Send the stego file and the stego key to the receiver**.**

4.1.2 Seek Algorithm

1. Read a character from the stego key and convert it into ASCII equivalent (num).
2. Read a character from the stego file and convert it into ASCII equivalent (n).
3. num=num-n.
4. If num is even then bit b = 0.
5. Else then bit b = 1.
6. Write b in a file.
7. Execute above steps repeatedly till the end of the stego key.
8. Convert the file into its character equivalent.

4.2 Fifth Approach: Hiding Data In Spaces

This approach makes use of a pre-defined cover file which can be any meaningful piece of English text and can be drawn from any source (For example, a paragraph from a newspaper/book). The approach works by hiding message a digit in each space present in the cover file. This approach works on the ASCII value of a character. After converting the cipher text to equivalent ASCII value, each digit of the ASCII value is hidden in the spaces of the cover file and using the using the position of the space operations are performed on the digit to be stored. Since no change is made to the cover, the cover file and its corresponding stego file are exactly the same

4.2.1 Hide Algorithm

1. Get a cover file.
2. Initialize i to 1.
3. Convert the input file to its ASCII equivalent (asc).
4. Read a digit (d) from the asc.
5. Read the position (p) of spaces form the cover file.
6. d = d power i.
7. i=i+1.
8. d=d+p.
9. Write character equivalent of p in the stego key.
10. If i=4, then i=1.
11. Repeat steps from 3 to 9 till the end of input file.
12. Send the stego file ,stego key to the receiver.

4.2.2 Seek Algorithm

1. Initialize i to 1.
2. Read a character from the stego key and convert it into ascii equivalent(d).
3. Read the position (p) of spaces form the cover file.
4. d = d - p.
5. d = i^{th} root of d.
6. Write d in a file.
7. Execute above steps repeatedly till the end of the stego key.
8. Convert the file into its character equivalent.

V. Results And Discussions

The execution starts with Fig 1, where the input type, algorithms to be compared are to be selected, text input must be given (either in file/direct text format) and cover file must also be provided (if required). Then the selected algorithms are executed, compared and their execution time, size of the file(s) generated are displayed in a tabular form as shown in Fig 2. Now it is the choice of the user to select which kind of graphical representation is required (out of the implemented ones) and proceed further to the graphical representation of the output generated as shown in Fig 3 and Fig 4. The generated graphs can be saved by clicking the save button in Fig 3 and Fig 4.The Table 1: shows different algorithms and their results for the given input data.

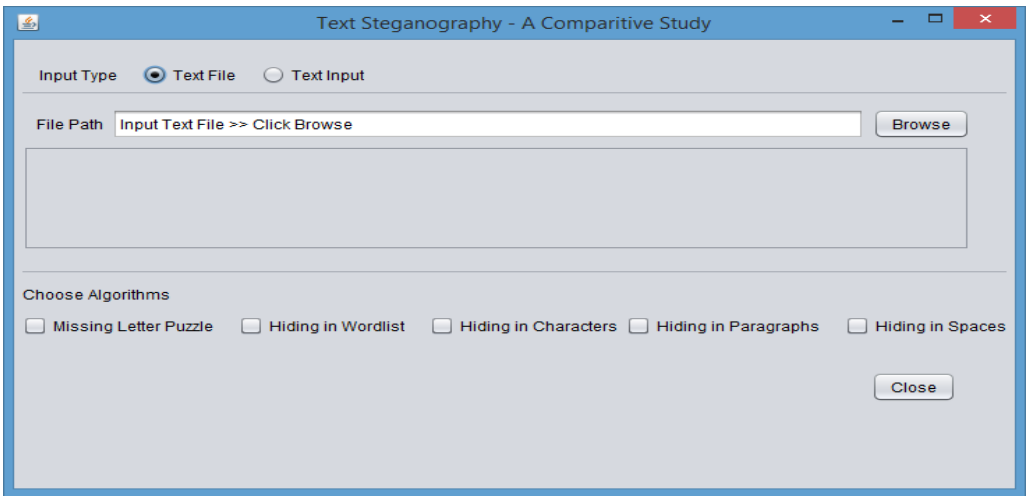

Fig 1:Input File Path

- Fig 1 shows either of the input type must be selected. (text file is default)
- A minimum of two algorithms are to be selected for the for initiating the process.
- Cover file must be provided if required.

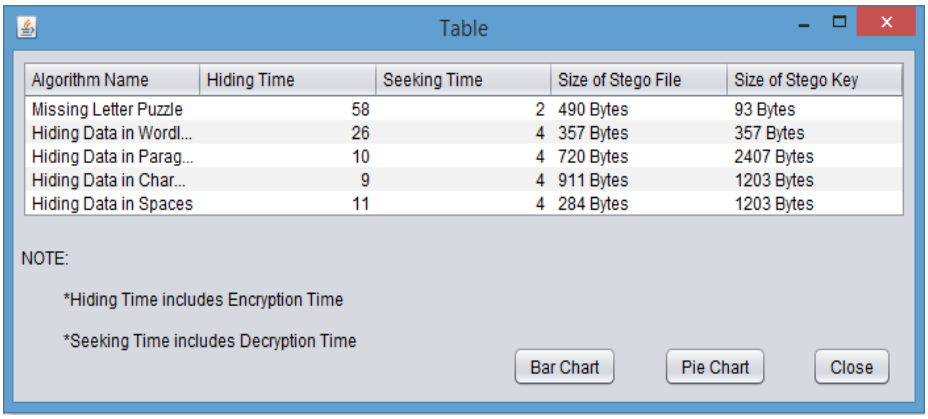

Fig 2:Table For Comparative Analysis

- The Fig 2 contains the list of algorithms selected for comparison along with their hiding and seeking time (in milliseconds), size of the stego file and stego key (in bytes)
- Hiding time includes Encryption time and Seeking time includes the decryption time.

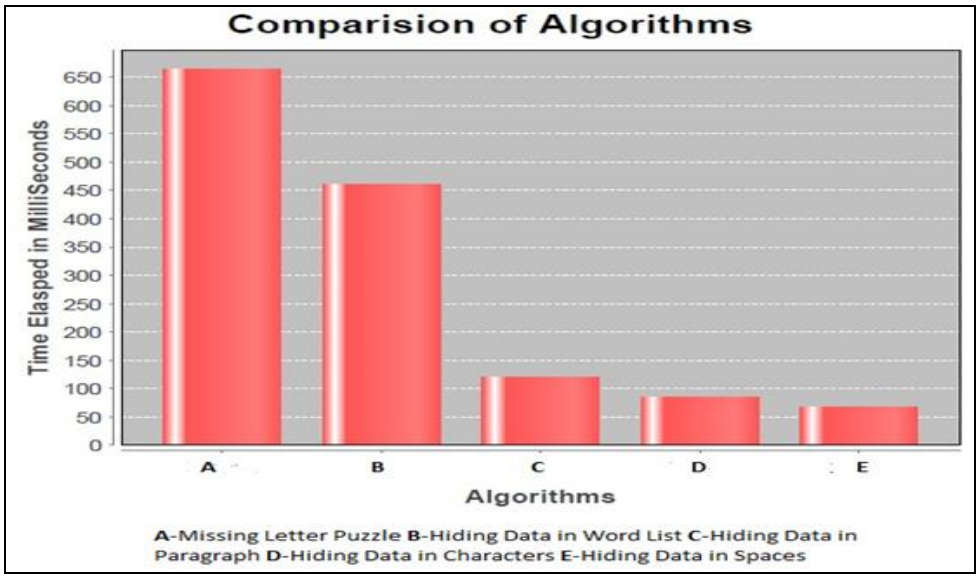

Fig 3: Bar Chart

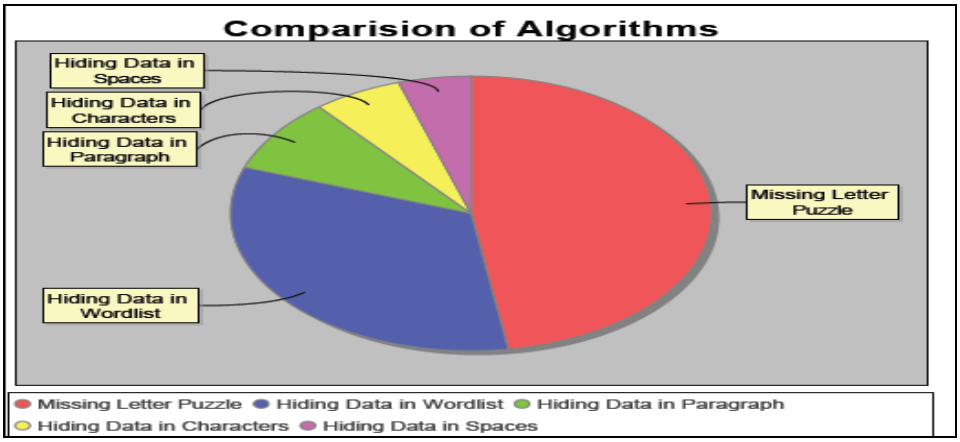

Fig 4: : Pie Chart

VI. Tabular Form

FILE SIZE	10KB			12KB			15KB		
ALGORITHM NAME	HIDING	SEEKING	FILE(S) SIZE	HIDING	SEEKING	FILE(S) SIZE	HIDING	SEEKING	FILE(S) SIZE
MISSING LETTER PUZZLE	11451	109	146780	13213	172	181281	17519	187	255962
HIDING DATA IN WORDLIST	9704	78	218110	12137	125	268862	16370	126	379410
HIDING DATA IN PARAGRAPH	671	156	680111	842	219	840380	983	265	1185965
HIDING DATA IN CHARACTERS	656	758	355167	827	140	438071	905	172	619360
HIDING DATA IN SPACES	484	109	268353	577	149	331730	593	187	467742

Table 1: Table of Different Algorithms And Results

VII. Conclusion

By this comparison, we come to a conclusion that-

1. If time taken for the process of comparison (includes hiding and seeking time) alone is considered then, Hiding data in spaces is an efficient algorithm that can be used.
2. If space required (includes size of stego key and stego file) alone is considered then, missing letter puzzle is an efficient algorithm that can be used.
3. If both, time taken for the process of comparison and space required are considered then, Hiding data in characters will be the efficient algorithm that can be used.

VIII. Future Enhancement

1. As we are only implementing five algorithms in this paper as of now, additional algorithms can be implemented and added in the future to make a bigger comparative study on many different algorithms.
2. We can also show results using different graphical representations other than bar and pie charts.

References

[1]. Monika Agarwal "Text Steganographic Approaches: A Comparison" of International Journal of Network Security and its Appications ,Vol.5.No.1,Janauary 2013.

[2]. F.A.P.Petitcolas R.J.Anderson, and M.G.Kuhn, "Informatio Hiding -A Survey", In proceedings of IEEE, Vol 87,pp.1062-1078,1999.

[3]. S.Changder, D.Ghosh, and N.C.Debnath, "Linguistic approachfor text steganography through India text", 2012 2nd Int.Conf. on Computer Technology and Development,2010,pp. 318-322.

[4]. R.J.Anderson, and F.A.P. Petitcolas, "On the limits of Steganography", IEEE Journal of Selected Areas in Communication, vol. 16, pp. 474-481,1998.

[5]. I. Banerjee, S. Bhattacharyya, and G. Sanyal, "Novel text steganography through special code generation," Int. Conf. on Systemics, Cybernetics and Informatics, 2011, pp. 298-303.

[6]. T. Y. Liu, and W. H. Tsai, "A new steganographic method for data hiding in Microsoft word documents by a change tracking technique," IEEE Transactions on Information Forensics and Security, vol.2, no.1, pp. 24-30, 2007.

[7]. 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.

[8]. S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O. Gorman, "Document marking and identification using both line and word shifting," INFOCOM'95 Proceedings of the Fourteenth Annual Joint Conf. of the IEEE Computer and Communication Societies, 1995, pp. 853-860.

[9]. William Stallings, Cryptography and Network Security: Principles and Practice 5/e., India, Prentice Hall, 2011.