International Journal of Computer Applications Technology and Research
Volume 6–Issue 6, 259-263, 2017, ISSN:-2319–8656

Text Steganography Using Compression and Random Number Generators

Mohamed Y Elmahi
Elimam Almahdi University
Kosti, Sudan

Talaat M.wahbi
Sudan University of
Science and Technology
Khartoum, Sudan

Mohamed H. Sayed
The National Ribat University
Khartoum, Sudan

Abstract: A lot of techniques are used to protect and hide information from any unauthorized users such as Steganography and Cryptography. Steganography hides a message inside another message without any suspicion, and Cryptography scrambles a message to conceal its contents. This paper uses a new text steganography that is applicable to work with different languages, the approach, based on the Pseudorandom Number Generation (PRNG), embeds the secret message into a generated Random Cover-text. The output (Stego-Text) is compressed to reduce the size. At the receiver side the reverse of these operations must be carried out to get back the original message. Two secret keys (Hiding Key & Extraction Key) for authentication are used at both ends in order to achieve a high level of security. The model has been applied to different message languages and both encrypted and unencrypted messages. The experimental results show the model's capacity and the similarity test values..

Keywords: Text Steganography, Pseudorandom Number Generators (PRNGs), Huffman Compression Algorithm, Cryptography, Capacity ratio, Jaro-Winkler distance

1. INTRODUCTION

Information hiding is a powerful technique used in information security, It takes two general approaches Cryptography and Steganography to hide internet communications [1]. The word steganography comes from two roots in the Greek language, "Stegos" meaning hidden/covered/roof, and "Graphia" simply meaning writing [2]. The history of steganography can be traced back to around 440 B.C.

Steganography is a popular technique of information hiding approaches, the purpose of it to covert communication to hide the existence of a message from a third party. Steganography can be classified into four types image, text, audio and video steganography that is depending on the cover media used to embed secret message [3] (as shown in Figure 1). Due to the significance of the information Cryptography and Steganography are ways of secure data transfer over the Internet [4].

2. RELATED WORKS

Text steganography plays significant role in covert information on Internet. Text steganography although can be broadly classified into three types. Firstly, the Format based, which changes the formatting of the cover-text to hide the data.

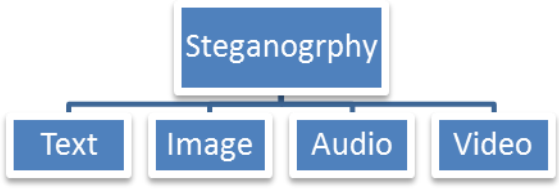

Figure. 1 Steganography Types

Secondly, Random and Statistical generation to avoid comparison with a known plaintext, steganographers often resort to generating their own cover texts. Lastly, Linguistic methods specifically consider the linguistic properties of generated and modified text, in this method a pre-selected synonyms of words are used [3-5].

A lot of studies cover text steganography such as:

Shirali-Shahreza, M.H. and M. Shirali-Shahreza [5] deal with the issue of text steganography, their model focuses on the letters that have points on them (example English Language had two letters i,j. while Arabic language has 15 pointed letters out of its 28 alphabet letters). Point steganography hides information in the points of the letters specifically in the points' location within the pointed letters. After converting the message into bits, if the bit is one the point in the cover text is shifted up, otherwise, the concerned cover-text character point location remains unchanged.

Gutub,A. and M. Fattani. A in [6], "That Benefiting from Shirali-Shahreza [5] proposes a new method to hide information in any letters (Unicode system) instead of pointed ones only. This model uses the pointed letters with extension after the letters to hold secret bit 'one' and the un-pointed Letters with extension to hold secret bit 'zero.

Bhattacharyya, S., I. Banerjee, and G. Sanyal [7] proposes a new method of information hiding in a text by inserting extra blank spaces (one or two spaces) between the words of odd or even size according to the embedding sequence (binary number) of the message.

Banerjee, I., S. Bhattacharyya, and G. Sanyal [8] do same as in [7], except it focuses on the first character of the words in the text cover, if it is a vowel or consonant instead of odd or even size.

Bhattacharyya, S [9]., design a secret key steganographic model combining both text and image first uses a plain text as the cover data and the secret message is embedded in the cover data to form the stego text which in turn is embedded into the cover image to form the stego image. The proposed text steganography scheme has been inspired by the author's previous work [8]. Here data embedding in an image

has been done through Pixel Mapping Method (PMM) within the spatial domain of any gray scale image.

3. HUFFMAN COMPRESSION ALGORITHM

Data compression schemes can be divided into two broad classes: lossless compression schemes, and lossy compression schemes. Lossy compression techniques involve some loss of information. Lossless compression techniques involve no loss of information.

Huffman coding is a lossless data compression. It uses a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It was developed by David A. Huffman[10].

The algorithm constructs a tree that is used to represent the characters in the file to be compressed; in the tree (a binary tree) all characters are stored at the tree leaves, each characters has an associated weight equal to the number of times the character occurs in a file. The characters of large weight numbers have less representation bits.

4. RANDOM NUMBER GENERATORS

Random numbers play a significant role in the use of encryption for various network security applications. Random number generators (RNG) are of three types; the first types are the true random number generators (TRNGs) that their output cannot be reproduced. TRNGs are based on physical experiment such as coin flipped 80 times and the Result recorded as binary bit. So it is impossible to generate bit same bit again by using of the same way.

The second types are Pseudorandom Number Generators (PRNG) generates sequences which are computed from an initial seeds, and produces a sequence of output bits using a deterministic algorithm. Typically, PRNG can work by feedback path. PRNG uses the flowing formula:

$$S[i+1] = S[i]*A + b \bmod m; i = 0,1,2,3....$$

$$S[i]; A; B \in \{0,1,2.....m-1\}$$

A; B; m are integer constants.

Third Pseudorandom Number Function (PRF) is used to produce a pseudorandom string of bits of some fixed length such as fixed length keys.

5. THE PROPSED MODEL

The model focuses on text steganography specifically the second type (Random and Statistical generation); the proposed model facilitates text steganography using to be side to side with cryptography to secure sent traffic. The model is divided into two major sites the sender site (Embedding+ Compression) that deals with the embedding processes of the secret message, and the receiver site (Decompression+ Extraction, in reverse order) that deals with the extraction processes to obtain the Secret message safely again. applied to reduce the size. As shown in Figure 2

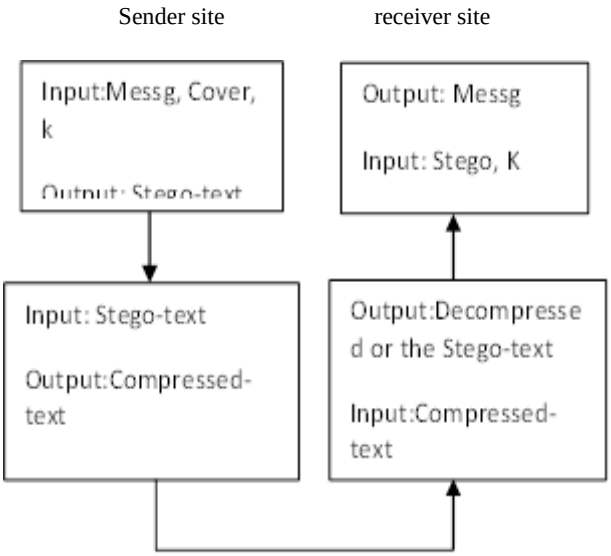

Figure. 2 The Proposed Model

The idea in the Model is to produce random text characters called Cover-text to hide the Secret-Message randomly (different positions) into the Cover-text by the use of the Pseudo Random Generation (PRNG).

The Embedding Algorithm (Sender Site):

-enter the Secret Message.

-calculate the message characters number

-generate the Cover-text from the secret Message and it is more than the message characters number

-enter a number to be the Hiding key.

-from the key generate an array of random number by the equation . A, N Are const.

-generate the binary array (just zeros and ones) by the reminder (mod) of 2 of the above step output numbers.

-generate the binary array until the ones (1s) numbers in the array equal the number of the Secret-Message characters.

-if the binary element equal to 0 writes one character from the Cover-text into the Stego-text, else write from the Secret-Message into the Stego-text.two texts merged into one text file randomly.

-do until the last character in the Secret-Message embedded. The output is a mix of random Cover-text with random position of every single character of the Secret-Message into the new file called Stego-text although it totally seems to be random text.

-Compression process (of Stego-text) it is the last step in the sender site. Figure 2

The Extraction Algorithm (Receiver Site):

- Decompression Stego-text file that is received from sender.

-receive the Decompressed file (or the Stego-text) from Second level of the receiver site.

- enter the extraction key.

-generate array of integers from the key by the same equation in-generate a binary array from the above array of integers by the same way of the first level in the sender site.

-if the binary array element equal 1 write from the stego-text to new text file called The Secret-message, else write one character from the stego-text to new text file called Cover-text2.

-do until the last character in the Stego-text.

Suppose that we have a Message (Hello World) we want to hide by using our model, First extract unique characters from the Message (Helo wrd), Second generate random Cover-text from the unique message characters (oWloWHd W oldHo), Third generate embedding bits from the hiding key (01100101010000100010110011), Fourth use the bits sequence to mix the message and the cover-text to produce Stego-text (oHeWllolWoHd W olWdorHold), fifth compress the Stego-text. The receiver must reverse all these steps to obtain the message again if he has the Extraction key that decided by sender. If the sender decides to send the same message again the cover-text and the stego-text will be different from the above one.

By the unknowing of the message and the cover-text the attackers cannot extract the message from the stego-text, by combining this with encrypted text message

6. RESULTS and DISCUSSION

The different experiments are done to different files of secret message (177, 49 character). The model capacity (which is an ability of a cover media to hide secret information) and similarity (which is the differences between cover text and Stego text) tested. Also Arabic & Encrypted file can be examined (work in this model) too.

The capacity ratio is calculated by dividing the amount of hidden bytes over the size of the cover text in bytes [21].

Capacity ratio = (amount of hidden bytes) / (size of the cover text in bytes)

177/181=0.98 (percentage 98%)

Jaro-Winkler distance for measuring similarity between two strings (s1, s2), it uses as a duplicate detection, Jaro-Winkler value is a ratio between 0 (no similarity) and 1 (an exact match). The Jaro Winkler distance (dj) formula is

$$dj = \frac{1}{3}\left[\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right]$$

Where m is number of matching characters, t is the number of transformation. The match Range computed by

$$matchRange = \left[\frac{max\lfloor|s1|\rfloor, \lfloor|s2|\rfloor}{2}\right] - 1$$

Table 1: Result Table

Mess	Cover	Stego	Comps	Ca-Ratio	JW-Ratio
49	47	96	444	1.04	0.5843
49	40	89	407	1.23	0.5622
49	35	84	381	1.40	0.5714
177	181	358	1599	0.98	0.6397
177	169	346	1557	1.05	0.6260
177	165	342	1547	1.07	0.6144

Table 2: Average Result Table

Mess	Cover	Stego	Comps	Ca-Ratio	JW-Ratio
49	41	90	411	1.22	0.57
177	172	349	1568	1.03	0.63

In all the tables above from left to right columns Mess-No are the values of the message characters number Cov-No are the values of the cover characters number, by adding these two values that gave us the Stego-No values, after the compression of the Stego-No files the values in bits found in the column Comp-Bits, Ca-Ratio are the capacity test values. JW-Ratio are the similarity test values between cover text files and stego text files.

If the values in the column Mess-No multiplied by 8 (suppose it is ASCII), that gave us the message size in bits instead of characters number (e.g. 49x8=392 or 177x8=1,416), if the comparison made between the those values (message value in bits) and the values of the column Comp-Bits in the same row, you will find small differences may be less or greater than the message bits value. Those differences are due to two reasons, first the random generation of the cover characters, second the Compression Algorithm that depends on the character frequency in the file, repeated characters which represented as less bits.

7. CONCLUSION AND RECOMENDATION

The purpose of this study is to conceal the sensitive information from an unauthorized use by hiding the Secret Message into Cover-Text generated randomly with the ability of extracting the Secret Message again. The study utilizes a compression algorithm as the next step that adds a good feature to the model. The essential role of a good compression algorithm is to reduce the size of the files, the increasing demand for the compressed data is to hasten the transfer rate

and operation, after the compression process is done the output of the compressed file is different from the original file itself, but it is near to the original Secret message's size.

In other site Text Steganography (in the model) is measured, such as the number of the Cover-Text characters that is used to randomize the message inside it, it used as rate of security (the larger number of the Cover-Text the better security indicator). Although the Embedding operation is faster(by embedding 8bit at a time instead of 2 or 3bit at a time than the other).

if the sender decided to send the same message more than one the output of the stego-text will be differ (due to of the random cover-text generating, and although Huffman Compression algorithm).

The model eliminates the overhead of find suitable cover-text to hide different messages (difference in type or size), the sender and the receiver exchanges stego-text files only any third party cannot obtain the secret messages again because of hidden cover-text files that produced and without having the extraction keys. The model does not use cover text data set.

8. FUTURE WORKS

Message permutation before the sending can be applied to the model, also we can embed the last character instead of the first one

First alter the design to accept different type of messages such as images, audio, etc. Second apply the random embedding operations to the binary representation (e.g ASCII) outputs of the secret message and the cover-text to hide the data more deeply. The model can also be implemented to the data transferred through network or internet whether it is plaintext or encrypted data.

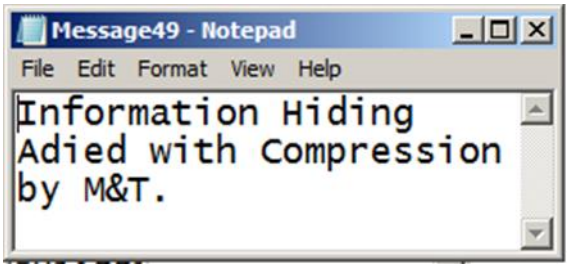

Figure. 3 49 Characters Message

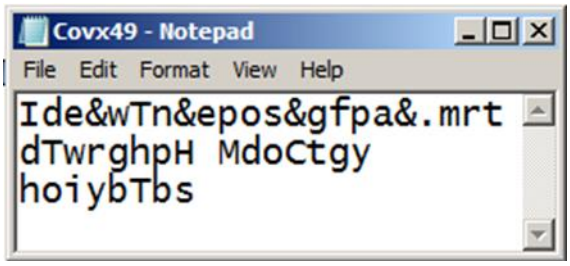

Figure. 4 Cover-text of the Message Above.

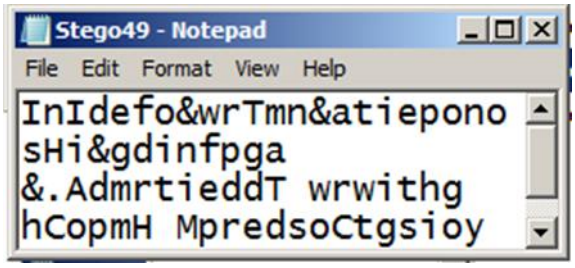

Figure. 5 Stego text

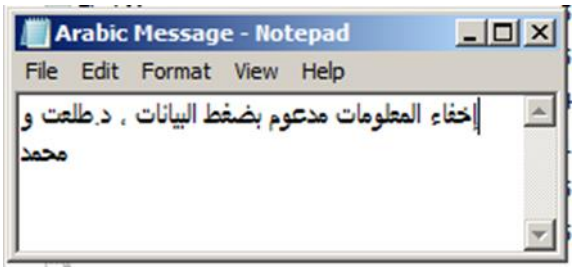

Figure. 6 Arabic messages

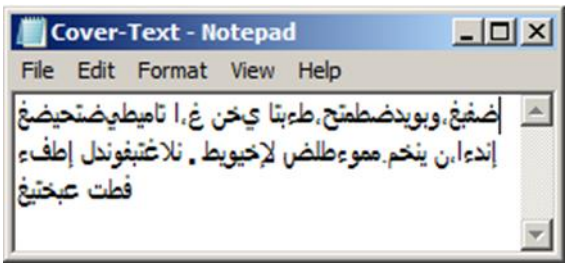

Figure. 7 Arabic Cover-text

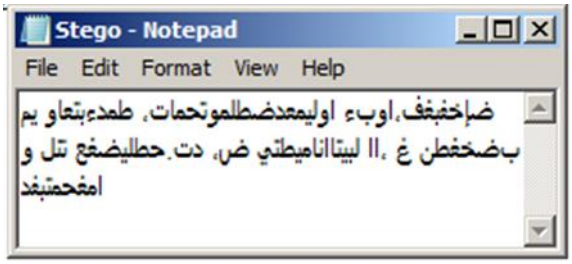

Figure. 8 Arabic Stego- text

9. REFERENCES

[1] Al-Najjar, A.J. 2008, *The decoy: multi-level digital multimedia steganography model*. in *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. World Scientific and Engineering Academy and Society.

[2] Krenn, R., 2004, *Steganography: Implementation & Detection.* found online at<http://www.krenn.nl/univ/cry/steg/presentation/2004-01-21-presentation-steganography.pdf,

[3] Isbell, R., 2002, *Steganography: hidden menace or hidden saviour*. Steganography White Paper.

[4] Agarwal, M., 20135(1) TEXT STEGANOGRAPHIC APPROACHES: A

COMPARISON. International Journal of Network Security & Its Applications,.

[5] Shirali-Shahreza, M.H. and M. Shirali-Shahreza, 2006, *A new approach to Persian/Arabic text steganography*. in *Computer and Information Science, 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COMSAR* .

[6] Gutub, A. and M. Fattani, 2007, *A novel Arabic text steganography method using letter points and extensions*. in *WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria*.

[7] Bhattacharyya, S., I. Banerjee, and G. Sanyal, 2010 *A novel approach of secure text based steganography model using word mapping method (WMM).* International Journal of Computer and Information Engineering, **4**(2): p. 96-103.

[8] Banerjee, I., S. Bhattacharyya, and G. Sanyal, 2011 *Novel text steganography through special code generation*. in *Proceedings of International Conference on Systemics, Cybernetics and Informatics (ICSCI-2011), Hyderabad, India*.

[9] Bhattacharyya, S.,2011, *Data hiding through multi level steganography and SSCE.* Journal of Global Research in Computer Science.

[10] *Huffman Compression Algorithm*. [cited 2014 2/3/].

[11] *A. Haidari, A. Gutub, K. A. Kahsah, and J. Hamodi, July 2012 "Improving security and capacity for Arabic text steganography using "*kashida" extensions," *2009 IEEE/ACS Int. Conf. on Computer Systems and Applications, 2009, pp. 396-399.*Sharon Rose Govada, Bonu Satish Kumar, Manjula Devarakondaand Meka James Stephen:Text Steganography with Multi level Shielding. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3,