Secure Text Steganography

P. Akhilandeswari and Jabin G. George

Abstract Steganography is an art in which the data can be hidden in other data as cover, the text files are commonly used for hiding data. The main aspects of steganography are the capacity and security, where the capacity refers to how much data can be hidden in the cover carrier, while the security concerns with the ability of disclosing or altering the data by unauthorized party. The aim of this project is to implement an algorithm to reduce the size of objects created using steganography. In addition, the security level of each approach is made more secured. This project presents an overview of text steganography and various existing text-based steganography techniques. Highlighted are some of the problems inherent in text steganography as well as issues with existing solutions. A new approach is proposed in information hiding using interword spacing which reduces the amount of information to hide. This method offers generated stego text of maximum capacity according to the length of the secret message.

Keywords Cover · Security · Interword spacing · Stego text · Capacity

Introduction

In the field of Data Communication, security issues have got the top priority. So, of late, the degree of security provided by a security tool has become the main evolutionary criteria of it. Classical cryptography is one of the ways to secure plain text messages. Along with that, at the time of data transmission, security is also

P. Akhilandeswari (✉) · J. G. George
Department of Computer Science, SRM University, Chennai, India
e-mail: akilasharanju@gmail.com

J. G. George
e-mail: jabing28@gmail.com

S. Sathiakumar et al. (eds.), *Proceedings of International Conference on Internet Computing and Information Communications*, Advances in Intelligent Systems and Computing 216,
DOI: 10.1007/978-81-322-1299-7_1, © Springer India 2014

implemented by introducing the concept of steganography, watermarking, etc. In these types of combined approach, there exist some drawbacks.

In remote networking, at the time of transmission of hidden encrypted text message, if the eavesdroppers get the track of the hidden text, then they could easily get the encrypted text [1]. Now breaking of encrypted text message can be achieved by applying some brute force technique. So, there remains some probability of snooping of information. So, these types of techniques incur another level of security which can route the Cryptanalyzer or Steganalyzer in a different direction [2].

Steganography

Steganography or Stego as it is often referred to in the IT community, literally means, "Covered writing" which is derived from the Greek language. Steganography is defined by Markus Kahn [3] as follows: "Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to Cryptography, where the unauthorized party is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any unauthorized party to even detect that there is a second message present" [4].

Existing System

Peter Wayner's Mimicry Algorithm

Peter Wayner [5] proposed a mimicry algorithm that aimed at text in his book Mimic Functions, Cryptologia XVI-3. His approach is to produce mimicked text that looks similar to the real structure of the original text. Peter Wayner used a set of grammatical rules to generate stego text and the choice of each word determines how secret message bits are encoded.

Merits:

- Use of grammar makes easy to debug and reduce the errors.
- Secret message can be encoded into something innocent looking, in a form of spam, which nobody will notice that there is a secret message being concealed.

Demerits:

- Complex logic
- The stego object file is larger.

Brassil's Document Coding Method

Brassil et al. [6] gave the initial idea of document coding methods in their paper by proposing life-shift coding, word-shift coding, and feature coding (character coding) to discourage illicit dissemination of document distributed by computer network [7]. Line-shift coding is a method of altering a document by vertically shifting the locations of text lines to uniquely encode the document. Word-shift coding is a method to alter a document by horizontally shifting the locations of words within text lines to uniquely encode the document. Character coding or feature specific coding is a coding method that is applied only to the bitmap image of the document and can be examined for chosen character features, and those features are altered, or not altered, depending on the codeword.

A document is marked in an indiscernible way by a codeword identifying the registered owner to whom the document is sent. If a document copy is found that is suspected to have been illicitly disseminated, that copy can be decoded and the registered owner identified [8].

Merits:

- Easy to implement
- Large number of characters can be embedded.

Demerits:

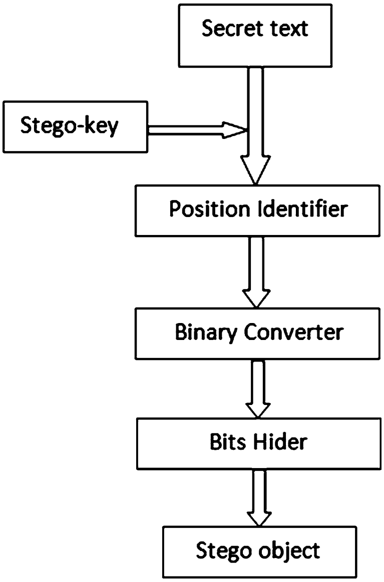
- This can be done in hard copy documents only.
- It can be observed easily.

Proposed System

Objective

In the proposed system, I have started an overview of text steganography and various existing text-based steganography techniques. Highlighted are some of the problems inherent in text steganography as well as issues with existing solutions.

A new approach is proposed in information hiding using interword spacing, which reduces the amount of information to hide. This method offers generated stego-text of maximum capacity according to the length of the secret message. This proposed system also analyzed the significant drawbacks of each existing method and how this new approach could be recommended as a solution.



Sender Side Manipulation

In sender side, the sender has to give the information which is to be hidden and also the stego-key which is usually a password. The following figure illustrates the overall concept of this proposal. The flow diagram shows the general blocks present in the flow of hiding process. Each block is explained as follows:

Algorithm

Step 1: Get the secret message file and stego-key from sender in order to hide the information.
Step 2: Find the size of the file. According to the size, generate the cover text file dynamically.
Step 3: Give the cover text file along with the password and secret information to Position Identifier which returns the position vector which contains the occurrence byte number in the cover text file of each character present in secret file.
Step 4: Give the position vector to binary convertor which returns the binary value of each element of the position vector.
Step 5: Then Manchester encoding is performed for the sequence of bits.
Step 6: Finally, those bits are hided in the cover text file using different hiding methods by Bits Hider.

An Approach to Reduce the Size of Stego Object and a Secure Text Steganography

Now a days the text steganography is widely used along with the cryptography or stand-alone method [9]. Depending on the type and size of the information to hide, the various methods have been proposed and implemented. This has several cases as follows:

If the secret information is very small (like any important personal message) where secrecy plays vital role, the message is hidden in text file because comparing to other medium like image, audio or video, hidden information in text file is hard to detect.

- If the secret information is medium, the capable size image is taken as the cover medium.
- If the secret information is large, the audio or video is taken as the cover medium depending upon the relative size of information.

In this project work, a very secure text steganography has been implemented that is explained in two phases.

Phase I Sender side process like encryption in cryptography
Phase II Receiver side process like decryption in cryptography

Process of Position Identifier

The main work of Position Identifier is to read each character from the secret file and to find the occurrence of that character in the cover text file. It returns the number of bytes traversed from the start of the cover text file as the first occurrence. It forms all byte number in the form of vector. The following flow diagram explains the concept clearly and the algorithm also given for ease of understanding [10].

Process of Manchester Encoding

The Manchester encoding is the traditional concept but it includes extra security layer in the flow of hiding process. The process Manchester encoding is that if it encounters high to low transition it encodes it as low transition. That is if the sequence of bits is ‘10’ then ‘0’ is encoded. If it encounters low to high transition it encodes it as high transition. That is if it is ‘01’ then ‘1’ is encoded. Other then these two sequences (00 and 11) simply discarded. The following flow diagram illustrates the concept clearly and algorithm also given.

Process of Bits Hider

The Bits Hider can use different hiding methods to hide the two bits namely 0 and 1. The very popular methods are as follows:

Receiver Side Manipulation

The receiver side process is opposite to sender side process. That is first Bits Extractor extracts the bits from stego object file. Then Decimal Converter converts the binary number to decimal number which is position vector values. It assembles the values and returns the position vector. Next Character Identifier identifies the character at the position value from position vector. It returns sequence of character. Then the Character Assembler assembles the character as secret message. The following flow diagram illustrates the concept and algorithm also given.

Algorithm

Step 1: Read the stego object file character.

Step 2: Give these characters to Bits Extractor which identifies the bits and return the Bit Sequence. The process of Bits Extractor can be understood from the opposite process of Bits Hider in the sender side manipulation.

Step 3: Give this Bit Sequence to Decimal Converter which returns the Position vector consisting of all decimal numbers. The process of Decimal Converter can be understood from the opposite process of Binary Converter in the sender side manipulation.

Step 4: Give this position vector to Character Identifier which identifies the character at the positions in the position vector. The process of Character Identifier can be understood from the opposite process of Position Identifier.

Step 5: Finally, Character Assembler gets the character from Character Identifier, and assembles them. It returns the Secret Text.

Process of Bits Extractor

The Bits Extractor can use different extracting methods to extract the single space, double space, and triple space. The very popular methods are as follows:

Process of Manchester Decoding

The Manchester decoding is the traditional concept but it includes extra security layer in the flow of extracting process. The purpose of Manchester decoding is that if it encounters high to low transition it encodes it as low transition.

Expected Output

The experiment on various sizes of input file our method shows reduced usage of space to hide. This is explained as follows:

For example, consider the following line of text we are going to hide: “I will come on Monday”. This sentence has 22 characters, that is, size of the file is 22 bytes. In order to hide these characters by this proposed method, the cover medium is taken as pangram sentences, which have all 26 letters in English. So the position vector contains 22 entries where each one entry for one character. The value ranges from 0 to 32 because the first sentence itself contains all 26 letters and we are looking for first occurrence. So to hide the numbers 0 to 32 requires 5 bits. So for 22 numbers totally 110 bits needed to hide. So it consumes around 14 bytes in memory. By traditional schemes they are directly hiding the character in binary form so it requires (22 × 8 = 176) 176 bits to hide. It consumes 22 bytes. So the amount of space saved is 8 bytes.

References

1. Por, L.Y., Delina, B.: Information hiding: a new Approach in text steganography. Paper presented at the 7th WSEAS international conference on applied computer & applied computational science’08, Hangzhou, China, 6–8 April 2008
2. Chapman, M., Davida, G., Rennhard, M.: A practical and effective approach to large-scale automated linguistic steganography. In: Proceedings of the Information Security Conference, October 2001, pp. 156–165. Malaga, Spain, October 1-3
3. Johnson, N.F.: Steganography. http://www.jjtc.com/stegdoc/index2.html. Accessed November 1995
4. Bennett, K.: Linguistic steganography: survey, analysis, and robustness concerns for hiding information in text. Purdue University, CERIAS Technical Report, 2004
5. Wayner, P.: Disappearing Cryptography: Being and Nothingness on the Net. Academic Press Inc., New York (1996)
6. Brassil, J., Low, S., Maxemchuk, N.F., O’Garman, L.: Electronic marking and identification techniques to discourage document copying. IEEE J. Sel. Areas Comm. **13**, 1495–1504 (1995)
7. Spammimic. http://www.spammimic.com (2000). Accessed 1 July 2009
8. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Syst. J. **35**, 313–336 (1996)
9. Provos, N., Honeyman, P.: Hide and seek: an introduction to steganography. Security & Privacy, IEEE **1**(2), 32–44 (2003)
10. Mohammed Al-Mualla and Hussain Al-Ahmad.: Information hiding: steganography and watermarking. http://www.emirates.org/ieee/information_hiding.pdf (2009). Accessed 1 July 2009