ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2 Issue 1 Jan-March 2014
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Data Encryption hiding technique in non-standard cover files

[I] V.Vijayalakshmi, [II]P.Mahalakshmi, [III]S.Thamizharasan
[I]Asst Prof., Dept. of CSE, Christ College of Engineering and Technology, Pondicherry, India
[II]Student M.Tech, Dept. of CSE, Christ College of Engineering and Technology, Pondicherry, India

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, a form of security through obscurity. One of the main fields of information security is the concept of hidden exchange of information. For this purpose, various methods including cryptography, steganography, coding and many methods are used. Steganography has many technical challenges such as high hiding capacity and imperceptibility. The present work shows how to embed secret message into anon standard cover file .txt. The secret message(SM) is encrypted using DES algorithm and shifting cipher method where both of them uses randomly generated key. The encrypted secret message is substituted inside the cover file (CF) by inter word spacing technique. This steganography method was proposed for embedding encrypted SM into CF and hence the security of steganography was very high. As the file(SM) is encrypted or compressed and then embedded into the cover file (CF), the security of the secret file(SM) is enhanced.

Keywords
Steganography, message, cryptography, encryption, hiding, compress

I. Introduction

Information hiding for programmers is done to prevent system design change. If the design decisions are hidden, then certain program code cannot be modified or changed. Information hiding is usually done for internally changeable code, which is sometimes specially designed not to be exposed. Change resilience of classes and ease of use by client objects are two by products of hidden data. One advantage of information hiding is yielding flexibility, such as allowing a programmer to more readily modify a program. This may also be done by placing source code within modules for easy access at a later stage as the program is developed and evolved. Steganography approach is done in standard cover files such as image, audio and video and some non-standard cover files. Hiding information inside in standard files such as images, video and audio is a popular technique now-a-days. An image with a secret message inside can easily be spread over the worldwide web. Hiding information inside audio files can be done in several different ways. Using the least-significant bit is possible, as modifications will usually not create audible changes to the sounds. It is possible to encode messages using frequencies that are inaudible to the human ear. Using any frequencies above 20,000 Hz, messages can be hidden inside sound files and will not be detected by human checks.

II. Related work

Advanced Steganography Approach for Hiding Encrypted Secret Message In LSB, LSB+1, LSB+2 and LSB+3 [1] bits in non-standard cover files method is to hide secret message in encrypted form in non-standard cover files such as .exe, .com, .pdf, .doc, .xls, .mdb, .ppt files is proposed. Advanced Steganography algorithms using encrypted secret message [2] a method for hiding any file in any standard cover file such as image, audio, video files is proposed. Advanced Symmetric key cryptography using extended MSA method: DJSSA symmetric key algorithm [3] method which is an extension of MSA algorithm dealing with multiple encryption and decryption of files such as image file, sound file, video file, text file, executable file or any other file to make the system more secured. New Symmetric key cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm [4] it is a new advanced symmetric key cryptographic method in which a new bit manipulation method for data encryption and decryption is used. A Challenge in hiding encrypted message in LSB, LSB+1 Bit positions in various cover files[5] The encrypted information is hidden in any form of cover files such as .exe files, Microsoft office files, .dbf files, image files, audio files and video files. A Novel approach to Format Based text Steganography [6] an algorithm on text steganography using the combination of line shifting and word shifting methods with copy protection techniques is proposed.

Information hiding in text using typesetting tools with stego encoding [7]**.** A method is proposed to embed secret message in text files by modifying the inter-word spaces of formatted text. A new steganographic method of data hiding in Microsoft word documents by a change tracking technique [8] the information is embedded in the degeneration stage of document transformations with steganographic effects is proposed. Text steganography through Indian languages using feature coding method [9] in this scheme the advantage of the existence of many feature code able characters in Indian languages are used for text steganography. Novel text steganography through special code generation [10] a text steganography is achieved through changing the vowel, consonant of the cover text according to the embedding sequence is proposed. Information Hiding: A New Approach in Text Steganography [11] an overview of text steganography and a brief history of steganography along with various existing text-based steganography techniques is presented. Data Hiding in Binary text documents [12] A new method for data hiding in binary text documents by embedding data in the 8-connected boundary of a character is proposed. An enhanced embedding method using inter-sentence, inter-word, end-of-line and inter-paragraph spacing [13]. A novel embedding mechanism which uses hybrid methods of inter-sentence, inter-word, end-of-line and inter-paragraph spacing to

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

embed the secret message is proposed. Between the paragraphs 6 points space has to be left before and after the paragraph.

III. Existing System

In existing system steganography approach is done in standard cover files such as image, audio and video and some non-standard cover files. Hiding information inside in standard files such as images, video and audio is a popular technique now-a-days. An image with a secret message inside can easily be spread over the World Wide Web or in news groups. The use of steganography in news groups has been researched by German steganography expert NielsProvos, who created a scanning cluster which detects the presence of hidden messages inside images that were posted on the net. However, after checking one million images, no hidden messages were found, so the practical use of steganography still seems to be limited. Hiding information inside audio files can be done in several different ways. Using the least-significant bit is possible, as modifications will usually not create audible changes to the sounds. It is possible to encode messages using frequencies that are inaudible to the human ear. Using any frequencies above 20,000 Hz, messages can be hidden inside sound files and will not be detected by human checks. In these files the secret message(SM) is encrypted using MSA. The encrypted secret message substituted inside the cover file (CF) by changing the LSB, LSB+1, LSB+2 and LSB+3 bits of the cover file.

A. Issues in the Existing System

The biggest problem, steganography faces is that of size. There is a limit to the size of a file which you can embed information into cover file. For instance, if you take a 16 bit image where each pixel is 4 bytes in three colours RGB you can only reliable encode the lower byte before the colour changes become visible in the viewed image. This means that the images that are used to embed the data have to be one quarter larger than the encrypted data itself.

The various other issues are

- Some of the text steganography system will require a mechanism for recovery of synchronization when encryption is done in blank spaces.
- Some system uses more iteration for embedding.
- Capacity is not reduced, due to lack of compression techniques.
- Could accidentally degrade or render an image misleading.
- In Peer-to-peer private communications it doesn't hide the fact that an e-mail was sent, negating the purpose of secret communications.
- In copyright protection a form of this already exists, called digital watermarking, but requires use of separate hardware tools because steganography software can't use separate hardware tools. Steganography software also can't protect the watermark.
- For embedding in the text files by modifying the least significant or least significant bit+1 or least significant bit+2 or least significant bit+3 positions makes the hidden file inappropriate by making the changes visible to the user.

IV. Proposed System

Our system proposes a method to embed secret files such as standard files, non-standard files such as doc, ppt, pdf, xlsfiles and pure text file. The secret message is encrypted using the following encryption algorithms as per the choice of the user.

- Shifting cipher encryption
- Data Encryption Standard algorithm.

The encrypted secret message is hidden inside the cover file by representing as inter word spaces in the cover file. In order to do this the encrypted secret message is converted as binary text containing 0's and 1's and each single space in the cover file is represented as zero and double space indicates one. In this system we propose a method with compression of the hidden file before sending over the network. Likewise a decompression is performed before the extraction process. The advantage of compression is that, the amount of secret message in secret files can be increased (i.e.) more amount of data can be hidden.

Algorithm1: Shifting Cipher

Step 1: Write down each letter of the alphabet. This is called standard alphabet, then select a number that will be used as the key for (ex. no =7).

Step 2: Starting with the letter A in your standard alphabet, count six letters to the right and write the letter A underneath the G. Continue writing the remaining letters of the alphabet until you come to the end of the standard alphabet

Step 3: Once the end of the standard alphabet is reached, go back to the beginning and write in the remaining six letters. The message is ready to encrypt

Step 4: Locate the first letter from your plaintext in the standard alphabet. The letter below it is the cipher letter

First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2,…, 'z'=25. We can now represent the Caesar cipher encryption function, e(x), where x is the character we are encrypting, as:

$$e(x)=(x+k)(\text{mod } 26)$$

Where k is the key (the shift) applied to each letter. After applying this function the resultis a number which must then be translated back into a letter. The decryption function is :

$$e(x)=(x-k)(\text{mod } 26)$$

Algorithm 2: Data Encryption Standard (DES) is a block Cipher

Step 1: An algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length.

Step 2: In this case, the block size is 64 bits.

Step 3: The block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the Feistel scheme.

Step 4: The Feistel structure ensures that decryption and encryption are very similar process but the only difference is that the sub keys are applied in the reverse order while decrypting.

The Feistel (F) Function

The F-function, depicted in Figure 2, operates on half a block (32 bits) at a time and consists of four stages:

Expansion

The 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted E in the diagram, by duplicating half of the bits. The output consists of eight 6-bit(8*6=48bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.

Keymixing

The result is combined with a sub key using an XOR operation. 16 48-bit sub keys one for each round are derived from the main key using the key schedule.

Substitution

After mixing in the sub key, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES - without them; the cipher would be linear, and trivially breakable.

Permutation

Finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box. This is designed so that, after expansion, each S-box's output bits are spread across 6 different S boxes in the next round.

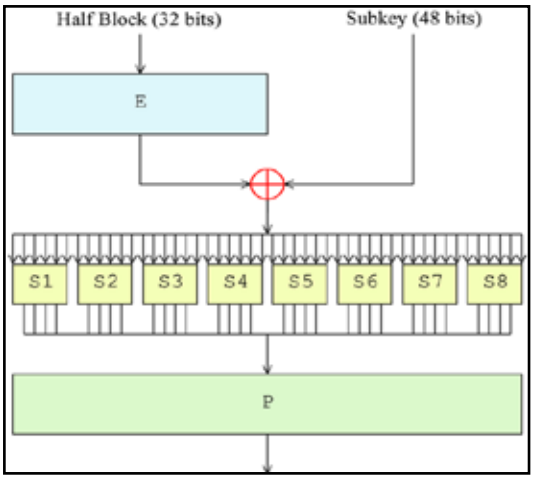

Fig. 1: The Feistel function of DES

Steps of Data Encryption Standard:

- Each block of message will be 64 bits. Do initial permutation on 64 bits data and divide it in to two halves.
- Left half 32 bits and Right half 32 bits.
- Expand right half up to 48 bits by expansion.
- Take 64 bits key (reduced to 56 bits by dropping bits at positions 8, 16, 24, …, 64) and select 48 bits by permuted choice.
- Do XOR of 48 bits right half and 48 bits key.
- Select 32 bits from step 5 by S-box substitution choice.
- Do P-box permutation (on 32-bits of step 6).
- Do XOR of 32 bits left half and 32 bits right half (from step-7)
- Result from step 8 will be new right half.
- Old right half from step 2 will be the new left half.

The above 10 steps make a cycle of DES.

Step 1 to 10 is for one cycle. There will be 16 such cycles. After completion of 16 cycles, we have to do final permutation on data bits to get decrypted data. In simple

$(L_0, R_0) \leftarrow IP\ (input)$

Take a S-box function f : $\{0,1\}^{48}$ x $\{0,1\}^{32}$ -> $\{0,1\}^{32}$ and 48-bit string keys $k_1,k_2,....k_{16}$ derived from the 56 bit key k, Repeat the following operation 16 times

$L_i = R_{i-1}$

$R_i = L_{i-1} f(k_i, R_{i-1})$

Output $\leftarrow IP^{-1}(R_{16}, L_{16})$

(i). Embedding

1. For the embedding process the two files cover file and the secret file is chosen.
2. The secret file is encrypted to make it unreadable and more secure according to the choice of algorithm by the user.
3. Then the data of the encrypted data is converted to binary data where the file contains only the characters '0' and '1'.
4. Then for each space in the cover file the secret file is checked. the various cases for creating stego file are

- If there is space in cover file and the secret file contains '0' then no change is done in the cover file.
- If there is space in cover file and the secret file contains'1' then an extra space is added to the cover file.

5. Thus the indication of one space in the cover file represents the binary '0' whereas the indication of two space represents the binary '1'.
6. The resultant file is saved as the stego file.

(ii). Compression

To make the system more secure and to store more information a compression technique is adopted. Considering various compression formats available such as Gzip, Huffman compression, LZW compression etc., the proposed system makes use of Deflator compression which is the combination of Huffman and LZ77 compression schemes.

The characteristics of Deflator compression are

- It is a lossless compression scheme in which that allows the exact original data to be reconstructed from the compressed data
- It provides better compression ratio and gives great flexibility to compress data

The comparison of the two ancestors of deflator compression is given below

a. Huffman Algorithm

- Elements of the alphabet are assigned with a weight which is the number that represents its relative frequency within the data to be compressed.
- At a time two elements are chosen with lowest weight and made to be leaf nodes of a tree such that one being the '0' branch and one the'1' branch
- The two nodes are combined and placed back with the other uncombined elements, and given a weight equal to the sum of the leaf nodes.
- When all the nodes are recombined into a Huffman tree, by starting at the root we can reach any element of the tree.
- In Huffman algorithm, a single set of elements and weight could generate multiple trees.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

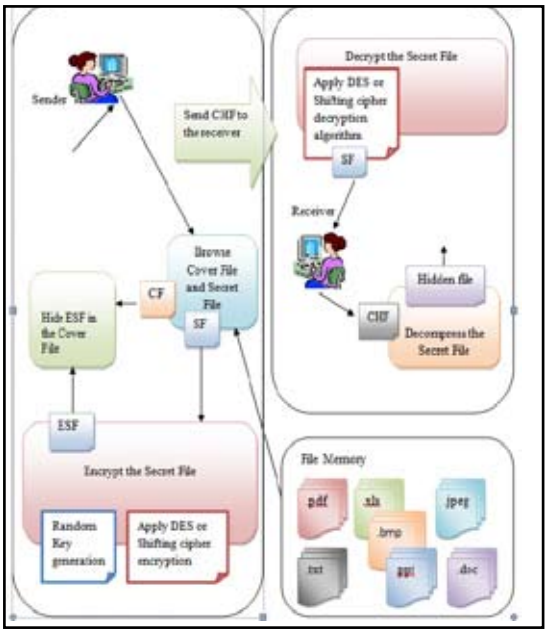

Fig. 2: Proposed Architecture

(b). Lz77 Compression

1. A 32K sliding window means the compressor have a record of what the last 32768(32*1024) characters.

- The sequence characters to be compressed is identical to the one in sliding window, the sequence of characters is replaced by two numbers
- Distance(how far back into the window the sequence starts)
- Length(number of characters the sequence is identical)

(c). Functions of Deflate Algorithm

It is a variation of Huffman algorithm with the following features

- Elements that have shorter codes are placed to the left of those with longer codes.
- Among elements with codes of same length, those that come first in the element set are placed to the left.

(d). Modes of Compression of Deflate Algorithm

- Not compressed (for data that's already compressed)
- Compression first with LZ77 and then with Huffman coding where the trees are defined by the deflate specification and does not needs extra space for storage.
- Compression first with LZ77 and then with Huffman coding where the trees are stored along with data.

(e). Decompression, extraction and decryption

1. On the receiver's side the file is first decompressed in order to extract the secret message. The decompression here is performed by reversing the compression process by making use of Inflator class of java.
2. For extraction the reverse of embedding is performed as follows,

- The stego file is read and the occurrences of two spaces are incorporated as the binary'1' and one space as the binary'0'.
- Step (a) is continued until the end of secret message is encountered.

3. For the decryption process, for both the algorithms the password is checked. The password acts as a key for decryption process. When the key matches with both sender and receiver, the encrypted message is decrypted.

V. Conclusion

An information hiding scheme for hiding standard non-standard file in similar cover file is introduced. This system tries to overcome the limitations of the existing scheme in which these non-standard files are hidden in non-standard file with a limitation of not being feasible for pure text files. In this project efforts are made to hide the non-standard file including pure text in a non-standard file by encrypting and compressing, so that it assures more security. In this system the sender sends data by encrypting using Shifting cipher algorithm or DES algorithm both of them generates a secret key generated by random key generation and then the encrypted secret file is embedded to the cover file by substituting in the inter-word spacing of the cover file. The recipient on the other hand, extract the encrypted secret message, decrypt by the same key to get the original message. This method is more secure as it provides facility to multiple encryption and decryption. The new technique proposed is the compression scheme in which the hidden message is compressed, ensuring more data to be transferred and enhancing further security. On the receiver side the file is decompressed and extracted before decryption thus providing a more efficient system. This application can be used in places where confidentiality of data is prime such as banking, defence and other core areas.

References

[1] *JoyshhreeNath, Sankar Das, ShalabhAgarwal and AsokeNath "Advanced Steganographic Approach for Hiding Encrypted Secret Message in LSB, LSB+1, LSB+2 and LSB+3 Bits in Non-standard Cover Files",2011*

[2] *JoyshreeNath and AsokeNath, "Advanced Steganography Algorithm using Encryptedsecret message",2011.*

[3] *DriptoChatterjee, JoyshreeNath, SoumitraMondal, SuvadeepDasgupta and AsokeNath, "Advanced Symmetric key Cryptography using extended MSA method: DJSSA symmetric key algorithm", 2011.*

[4] *NeerajKhanna,JoyshreeNath,JoelJames,AmlanChakrabarti, SayantanChakraborty and AsokeNath, "New Symmetric key Cryptographic algorithm using combined bitmanipulation and MSA encryption algorithm: NJJSAA symmetric keyalgorithm", 2011.*

[5] *JoyshreeNath, Sankar Das, ShalabhAgarwal, AsokeNath, "A Challenge in hiding encrypted message in LSB and LSB+1 bitpositions in various Cover Files", 2011.*

[6] *Sangita Roy and ManiniManasmita, "A Novel Approach to Format Based Text Steganography", 2011.*

[7] *Chen Chao, Wang Shuozhong, and Zhang Xinpeng "Information Hiding in Text Using Typesetting Tools with Stego-Encoding",2006.*

[8] *Tsung-Yuan Liu, and Wen-Hsiang Tsai "A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique",2007.*

[9] *S.Changder,S.Das and D.Ghosh, "Text Steganography through Indian Languages Using Feature Coding Method",2010.*

[10] *Indradip Banerjee, Souvik Bhattacharyya and Prof. GautamSanyal, "Novel Text Steganography through Special Code Generation",2011.*

[11] L. Y. POR, B. Delina, "Information Hiding: A New Approach in Text Steganography", 2008.
[12] Q. Mei, E. K. Wong, and N. Memon, "Data Hiding in Binary Text Documents", 2001.
[13] Lip Yee Por, KokOnnChee, Tan Fong Ang and DelinaBeh, "An enhanced embedding method using inter-sentence, inter-word, end-of-line and inter-paragraph spacing", 2011.

V. Vijayalakshmi she has completed her MCA from Sri Manakula Vinayagar Engineering College affiliated to Pondicherry University, India the M.Tech Computer Science & Engineering from Sri Manakula Vinayagar Engineering College affiliated to Pondicherry University, India. She has published 8 papers at international journals, 3 International Conferences, and 2 National Conferences. Her research area of interest is Cloud Computing, Web Service and Data Mining.

P. Mahalakshmi she has completed her B. Tech in Manakula vinayagar institute of technology affiliated to Pondicherry University, India and currently a final year M.Tech Student at the Computer Science of Christ College of Engineering and Technology affiliated to Pondicherry University, India.

S.Thamizharasan he has completed his B.Tech in Perunthalaivar Kamarajar Institute of Engineering and Technology affiliated to Pondicherry University, India and currently a final year M.Tech Student at the Computer Science of Christ College of Engineering and Technology affiliated to Pondicherry University, India.