Secure and Compressed Secret Writing using DES and DEFLATE algorithm

S. Smyrna Grace[1], T. Nalini[2] and A. Pravin Kumar[3]

[1,2]Department of Computer Science and Engineering, Bharath University, Chennai, Tamil Nadu, India

[3]Department of Computer Science and Engineering, Anna University, Chennai, Tamil Nadu, India

Abstract

A new secret writing using DES and DEFLATE algorithm is explored by improving the previous attempts in this sense in terms of security and compression. A Matching Pursuit algorithm using redundant basis decomposition technique is used to securely hide a message. The stability and computational complexity problems are solved by introducing new selection and update rules working entirely in the integer domain. Image decomposition is randomized in several ways thus improving the stego-message undetectability, and making the hidden message undetectable by targeted steganalyzers. A refinement decomposition step is applied to the three color bands to increase the stego-message payload. Due to the ever-increasing security threats, there is a need to develop algorithms with more complexity and advanced features for a secured message transfer. A comparative study is performed between Lempel-Ziv-Welch (LZW) data compression algorithm and deflate loss-less compression algorithm which proves the efficiency of deflate algorithm in both rate of compression and its compression speed.

Keywords: *DEFLATE, DES, Matching Pursuit, Redundant Basis Decomposition, Steganalyzer, Compression, and Security.*

1. Introduction

Security is the main concern over the transfer of files or messages from one system to another system, across any network. Security is nothing but protection of information from all kinds of threats. Security aspects come into play when it is necessary to protect information from the opponent who presents a threat to the confidentiality and eavesdropping.

The ultimate goal of steganography is that of hiding a message with in an innocuous signal in such a way that the very presence of the hidden message remains secret. Matching Pursuit algorithm uses redundant basis decomposition to securely hide a message within a cover color image is explored by improving previous attempts in this sense in terms of security and payload.

The stability and computational complexity problems of previous works are solved by introducing new selection and update rules working entirely in the integer domain, and by fully exploiting the availability of three color bands in such a way that all the available atoms in the three color bands are used to convey the stego-message. Image decomposition is randomized in several ways thus improving the stego-message undetectability, and making the hidden message undetectable by targeted steganalyzers explicitly developed to exploit the weaknesses of the Matching Pursuit algorithm.

The main purpose of this project is to produce a file that contains a hidden file of same type such that the file presence is not felt. The hidden file can be compressed and password protected if needed and can embedded with the master file. The file can be retrieved when needed by providing correct password.

Security is the main criteria while sharing files over network. Hence providing security is one of the needs of this system. The other needs of this system are providing a successful embedding of media file over another media file and performing decomposition in integer domain to reduce computational complexity. The media file here represents image, audio and video files.

Before getting to the essential, it would be appropriate to understand the possibilities of the proposed project. Steganography finds tremendous scope in areas, where there is a need to protect the privacy of information or securely transmit covert information. Consider the case of a spy satellite in orbit. It could be easily made to appear as a regular weather satellite and if a high capacity image Steganography system were available, the covert information the satellite gathers could easily be hidden in common place weather images [8].

Steganography techniques can also be used to hide classified patient information in X-ray and scan images of the patient. This provides a secure method of associating patient records with their own X-rays and scans.

Image Steganography [10] could also be used to embed secure information like customer name, account information and key presses in ATM camera feeds and numerous other legal applications. A credit card transaction on the Internet requires the credit card number to be transmitted from the buyer to the merchant and from the merchant to a transaction processing network. The system attempts to enforce confidentiality by encrypting the card number during transmission, by limiting the places where it might appear (in databases, log files, backups, printed receipts, and so on), and by restricting access to the places where it is stored.

Steganography is used by some modern printers, including HP and Xerox brand color laser printers. Tiny yellow dots are added to each page. The dots are barely visible and contain encoded printer serial numbers, as well as date and time stamps [8].

The rest of the paper is organized as follows. In Section II, a brief background on existing and proposed system is given. In Section III, the architecture of proposed system is given. The Section IV talks about expected result. The paper ends in Section V with some conclusions and hints for future research.

2. Background

The existing system was designed to overcome following issues,

- Instability of image decomposition.
- Minimum message carrying capacity during transmission.
- Computational complexity.

It revisits the system described by tackling the problems outlined above to develop a new MP algorithm (still referred to as MPSteg-color for the sake of simplicity) that permits us to take full advantage of the characteristics of the MP embedding domain. The main features of the new scheme (some of which are inherited from the algorithm proposed) can be summarized as follows:

- A particular MP decomposition strategy and a suitably tailored data hiding rule that permits us to solve the instability problems of MP decomposition are used.
- Several sources of randomization are included in the embedding algorithm to prevent targeted steganalyzers from detecting the presence of the hidden message; and
- A refinement decomposition step is applied to the three color bands to increase the stego-message payload.

As a further contribution with respect to previous works, the performance of existing scheme was thoroughly evaluated by testing the detectability of the hidden message against three classes of steganalyzers, namely: targeted steganalyzers explicitly designed to exploit the weaknesses of the MPSteg-color algorithm; steganalyzers developed to detect messages embedding by applying the embedding algorithm directly in the pixel domain; and general purpose steganalyzers [1].

The existing system was developed using Matlab which is an interpreted language. The main disadvantage of interpreted languages is execution speed. When a language is compiled, all of the code is analyzed and processed efficiently, before the programmer distributes the application. With an interpreted language, the computer running the program has to analyze and interpret the code (through the interpreter) before it can be executed (each and every time), resulting in slower processing performance [5].

The proposed system uses three algorithms, they are,

- Matching Pursuit algorithm
- Data Encryption Standard algorithm
- Deflate algorithm

2.1 Matching Pursuit algorithm

Matching pursuit is a type of numerical technique which involves finding the "best matching" projections of multidimensional data onto an over-complete dictionary. It uses a greedy algorithm to select a subset of atoms capable of representing the to-be-decomposed image efficiently [2].

2.2 Data Encryption Standard (DES)

Data Encryption Standard is a widely-used method of data encryption using a private (secret) key that was judged so difficult to break by the U.S. government that it was restricted for exportation to other countries. There are 72,000,000,000,000,000 (72 quadrillion) or more possible encryption keys that can be used. For each given message, the key is chosen at random from among this enormous number of keys. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key [3].

The DES has a 64-bit block size and uses a 56-bit key during execution (8 parity bits are stripped off from the full 64-bit key). DES is a symmetric cryptosystem, specifically a 16-round Feistel cipher. When used for communication, both sender and receiver must know the same secret key, which can be used to encrypt and decrypt the message, or to generate and verify a Message Authentication Code (MAC). The DES can also be used for single-user encryption, such as to store files on a hard disk in encrypted form.

2.3 Deflate Algorithm

Deflate is a lossless data compression algorithm that uses a combination of the LZ77 algorithm and Huffman coding [4].

Compression is achieved through two steps,

- The matching and replacement of duplicate strings with pointers.
- Replacing symbols with new, weighted symbols based on frequency of use.

2.3.1 Duplicate string elimination (LZ77)

Within compressed blocks, if a duplicate series of bytes is spotted (a repeated string), then a back-reference is inserted, linking to the previous location of that identical string instead. An encoded match to an earlier string consists of a length (3–258 bytes) and a distance (1–32,768 bytes). Relative back-references can be made across any number of blocks, as long as the distance appears within the last 32 kB of uncompressed data decoded (termed the sliding window) [6].

2.3.2 Bit reduction (Huffman coding)

The second compression stage consists of replacing commonly used symbols with shorter representations and less commonly used symbols with longer representations. The method used is Huffman coding which creates an unprefixed tree of non-overlapping intervals, where the length of each sequence is inversely proportional to the probability of that symbol needing to be encoded. The more likely a symbol has to be encoded, the shorter its bit-sequence will be [7].

3. ARCHITECTURE

The Architecture of the proposed system is illustrated in the below figure.

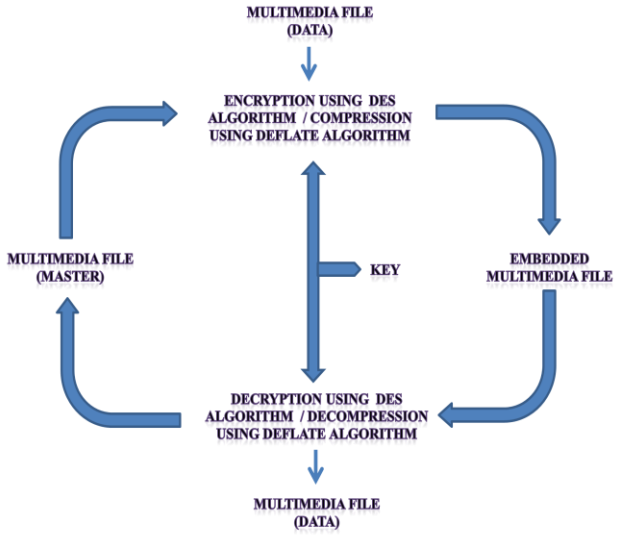

Fig. 1 Architecture of the system

This system can be split into,

- Source Selection
- MP Decomposition
- Embedding a Message
- MP Selection and update
- Retrieving the Message

3.1 Source Selection

In the first module we select the Master File, Data File and set the output path and name for the embedded output file. The master file can be a media file like audio, video , image or a text file like word document, text document etc.

Then we specify the output path and name of the output file. The third step involves the important part of selecting a data file. The data file can be a media file or a text file.

Usually the master file and data file are selected of same file type to provide a better embedding of data file into Master file. Here we also provide an advanced option to compress the data file and also protect it with password.

3.2 MP Decomposition

We first validate the conjecture that, due to the correlation between RGB color bands, computing the decomposition path on one band and using it on the other two does not impair the capability of the MP algorithm to extract the most important features of image blocks. Moreover, we give a measure of the payload allowed by the MP domain and the payload gain allowed by the decomposition refinement step. On one side this, is a good result showing a high degree of correlation; on the other side, it shows

that the decomposition path calculated on one band is capable of fully describe the content of the other bands, thus justifying the resort to a decomposition refinement step.

MP algorithms work by selecting an element of the basis of a time in a greedy fashion, with no guarantee of global optimality. In a steganographic scenario, the embedder usually first decomposes the image by using an MP algorithm, then modifies the decomposition coefficients to insert the stego-message, and then goes back into the pixel domain.

Due to the presence of the stego-message, when the decoder applies again the MP algorithm it may select from the redundant basis a different set of elements and in a different Order hence making it impossible for the decoder to correctly extract the hidden message [9]. Even though the subset of elements of the basis (and their order) is fixed, a change to one coefficient of the decomposition usually results in a variation of all the coefficients of the decomposition when the MP is applied to the modified image.

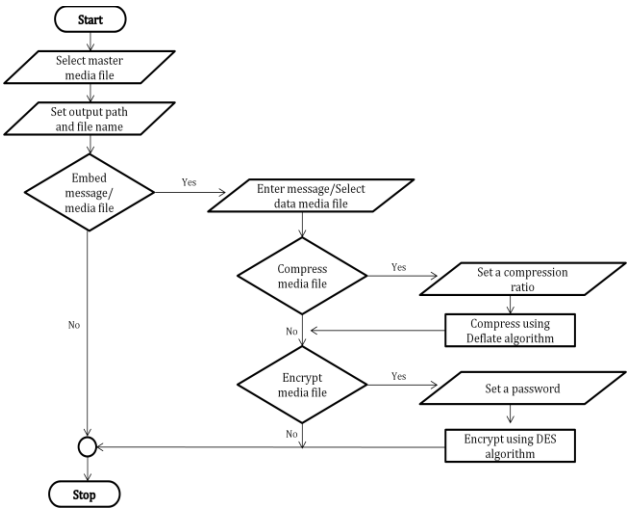

Fig. 2 Embedding a data message or media file

3.3 Embedding a Message

Data file can be of any size and of any format. Here the datafile is decomposed into its structural parts. For example the structural parts for image are small dots, curves, edges etc. These structural parts are replaced with the master file structural parts with help of dictionary by changing decomposition coefficients.

It is necessary that the transition from the pixel domain to the MP domain and then back to the pixel domain does not introduce approximation errors that could prevent the correct decoding of the stego-message. The easiest way of achieving this result consists of requiring that all the operations are performed in integer arithmetic with no need to quantize the stego image when the transformation from the MP to the pixel domain is performed. The second requirement stems from the very goal of all our work, that is to embed the stego-message at as high a semantic level as possible, hence the dictionary is as semantically meaningful as possible.

3.4 MP Selection and Update

Here we try to embed a data file which is of different format from the master file and also it is not always sure that the structural path of both data file and master file matches. In this both case we use MP selection and update rule. This rule avoids unnecessary quantization errors and coefficient instability.

If errors are large then new part is created and the output file is stored in the path specified as in first module with the same specified name.

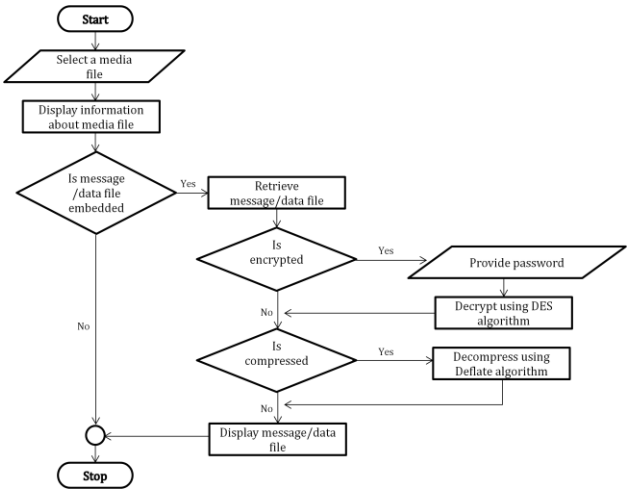

Fig. 3 Retrieving a data message or media file

3.5 Retrieving a Message

We first select the file from which message or data file has to be extracted and then check whether the file actually contains the embedded message or not, steganalyzers is used for this purpose. If the file contains any hidden message then the details about the file is displayed or else an error message is shown. The details are about file size, compression ratio if used and whether it is password protected or not. Then the permission to retrieve file is asked and if it is given the embedded file is extracted in the output path selected by user.

4. COMPARITIVE STUDY

A comparative study was performed between Deflate compression algorithms to that of Lempel-Ziv-Welch (LZW) data compression algorithm and the results showed that the deflate algorithm is efficient in both compression rate as well as the speed at which the compression is done.

Table 1: Compression techniques comparison

Data File Size (bytes)	*Compression using*		*Compression Ratio using*	
	Deflate (bytes)	*LZW (bytes)*	*Deflate (%)*	*LZW (%)*
17768	723	14740	95.93	17.04
53298	1135	29416	97.87	44.81
94038	24771	89956	73.66	4.34
99734	1637	43064	98.36	56.82
120054	1756	4780	98.54	96.02
186658	47403	156336	74.60	16.24

Below shown graph illustrates the comparison of compression ratio of Deflate algorithm to that of LZW algorithm.

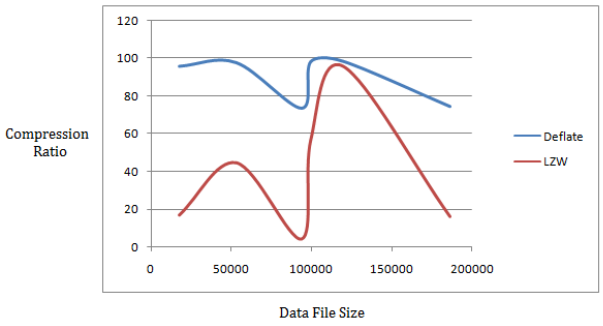

Fig 5: Comparison Graph

X-axis -> Data files size
Y-axis -> Compression ratio

5. RESULT

A data file or message will be embedded within a media file with additional feature of compression and password protection. The data file will be an image file, where as the media file will be an Image, Audio or Video file. This will be the More stable system proposed until now.

6. CONCLUSION & FUTURE WORK

An algorithm for embedding a stego message into color images represented by means of high redundant basis decomposition has been presented. The problems of previous schemes proposed in this sense will be solved, with particular attention to security and compression. The Data Encryption Standard algorithm will be used for password protection and the Deflate algorithm will be used for compression.

In addition to preventing the above security pitfall, the new scheme proposed in this paper slightly increases the admissible payload, by adding a decomposition refinement step.

In future this system can be re-structured to support cloud computing.

REFERENCES

[1] Giacomo Cancelli and Mauro Barni. "MPSteg-Color: Data Hiding Through Redundant Basis Decomposition", IEEE Transactions on Information Forensics and Security, vol. 4, no. 3, September 2009.
[2] http://en.wikipedia.org/wiki/Matching_pursuit
[3] Applied cryptography: protocols, algorithms, and source code in C, by Bruice Schneier, Second Edition, John Wiley & Sons, 1996.
[4] http://en.wikipedia.org/wiki/DEFLATE
[5] Giacomo Cancelli, Gwenael Doerr, Ingemar J. Cox and Mauro Barn, "Detection of ±1 LSB steganography based on the amplitude of histogram local extrema" in Image Processing, ICIP 2008.
[6] http://en.wikipedia.org/wiki/LZ77_and_LZ78
[7] http://en.wikipedia.org/wiki/Huffman_coding
[8] G. Cancelli and M. Barni, "MPSteg-color: A new steganographic technique for color images," in information Hiding: 9th Int. Workshop (IH 2007), Saint Malo, France, Jun. 11–13, 2007, vol. 4567, pp. 1–15.
[9] P. Vandergheynst and P. Frossard, "Image coding using redundant dictionaries," in Document Image Compression, M. Barni, Ed. Boca Raton, FL: CRC Press, May 2006.
[10] Cryptography for dummies, by Chey Cobb, Wiley Publishing Inc, January 2004.

[1]**Mrs. S. Smyrna Grace** has received B.E. Instrumentation and Control Engineering from the Jayamatha Engineering College affiliated to Anna University, Chennai in May 2006. She is currently pursuing M.Tech. Computer Science and Engineering (Final Year) in Bharath University. She is a member of ACM.

[2]**Dr. T.Nalini** received Ph.D in computer science and Engineering in 2011 from the Bharath University. She was completed M.Sc from the Karanataka university, M.Tech from the Bharath University in 2000,2007 respectively. She is working in the teaching field from 2000 to till date. Currently She is an Associate Professor in the Department of CSE. She has published more than 7 research papers in international journals. She also presented more than 25 papers in national conferences and 5 papers in international conferences. She is a life member of many professional bodies like ISTE, CSI, IEEE, IAENG.

[3]**Mr. A. Pravin Kumar** has received B.E. Computer Science and Engineering from the Bharath Institute of Science and Technology affiliated to Anna University, Chennai in May 2005. He has also done his MBA in eBusiness from Annamalai University in May 2007. He is a Sun Certified Java Programmer, Sun Certified Web Component Developer and Six Sigma Green Belt holder. He is currently working with a reputed software company.