A Novel Approach for Colour Image Steganography using Hash Algorithm

Ms Minu Mathew
Federal Institute of Science and Technology,Kerala,India

Mr Anoop E G
Federal Institute of Science and Technology,Kerala,India

Abstract

Steganography has been proposed as a methodology for transmitting messages through innocuous covers to conceal their existence. The word Steganography is originally composed of two Greek words steganos and graphia, which means "covered writing". Any digital file such as image, video, audio, text or IP packets can be used to hide secret message. Generally the file used to hide data is referred to as cover object, and the term stego-object is used for the file containing secret message. Among all digital file formats available nowadays image files are the most popular cover objects because they are easy to find and have higher degree of distortion tolerance over other types of files with high hiding capacity. In this paper a hash-based approach for colour image steganography is proposed. The proposed system can be used for embedding data in any type of colour image such as bmp, jpg, tiff, gif.

1. Introduction

Steganography is an art to hide the important data from the unauthorised user during transferring or communication through any medium. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts. Data used to hide data in steganography can be text or image. A few algorithms for image based steganography such as, Chaos [8],International Data Encryption Algorithm (IDEA) [2],Advanced Encryption Standard (AES) [6], Data Encryption Standard (DES) [2], Message Digest 5(MD5) [7], Bit Stream Ciphers (BSC) [11], Secure Hash Algorithm (SHA) [4], [10], etc have been proposed in recent times. Some of these algorithms such as AES, DES, and IDEA are good for little amount of data but not good for massive data sets as these algorithms involve intensive computation and required super fast processing machines [4], [5]. Similarly, other algorithms such as MD5, SHA are based on cryptographic hash functions that use a hash key of 16 byte. But these algorithms are vulnerable in terms of providing security due to inherent flaws caused by used checksum approach. [3], [4].Here we propose a novel approach that uses robustness of perfect hash-function algorithm that is more secure and more efficient.

2. Problem statement

Steganography generally includes the selection of a steganography carrier, and also the two complementary processes design: coding and decoding. The steganography is the process of embedding the information by a slight alteration of some carrier property. The decoding process of stegnography extracts the hidden information. Some of the existing algorithms like International Data Encryption Algorithm (IDEA) [7], Advanced Encryption Standard (AES) [3], Data Encryption Standard (DES) [7], are efficient and good for small amount of data but not sufficient for massive information sets as these algorithms involves more complex computation and required much fast processing machines [4], [5]. Some other algorithms such as BSC, MD5, SHA are in-efficient these algorithms are inefficient in terms of providing flawless security [3], [4], [5]. Due to such problems with the available approaches, the field of steganography still remains un-adopted and ineffective.

There is no doubt about the significance and criticalness of steganography in the field of cryptic and secure data transmission [1] on internet but there is need of a better and improved algorithm that is not only efficient but also secure. To achieve this goal we present a novel hash-based algorithm that uses a hashing technique to code and decode data in a colour image. Our presented approach is based on perfect hash function [12].

3. Proposed system

The presented approach allows the user to embed their secret textual information in images in a way that can be invisible and doesn't degrade or affect the quality of the original image. The target users of the presented system are those who want to make their information secure or protect their work form other or illegal use. This system provides an efficient way for secure transfer of information. The presented approach is able to manipulate with different file formats e.g. Bitmap, jpeg/jpg, GIF, and TIFF. Our system is able to load the above mentioned image file formats and code the given piece of text into the image.

3.1. Input image

A user can input a (bmp, jpg, gif or tiff) image in which he/she wants to hide his/her personal data for privacy purposes. It is recommended that input image should be medium sized to get better results, as very large sized image will use more bandwidth on internet and very small sized images may lost their bit of quality for bulky size of input text data.

3.2. Input textual data

Input the text file containing the textual data which the user wants to code in the image. The input text file is read by our system and the chunks of 3 characters (including space characters) are made. One chunk of data is stored in place of one pixel in the image. For example a string "My name is Sradha Ann Jobin." is processed as"[My][nam] [e i] [s S] [rad] [ha] [Ann] [Jo] [bin] [04]"To point the end of the message, at the end of the message an End-Of-File (EOF) character is stored using ASCII value of EOF that is 04.

3.3. Coding data in image

For coding textual data in the image, a hash-based algorithm (described in section 4) is used. Basic purpose of using the hash-based algorithm is to pick pixels randomly to store chunks of input data. The used algorithm randomly generates a hash-key that is afterwards used by the algorithm to generate a pattern of pixels, where the data will be stored. The data chunks are stored in red, blue and green bytes of the selected pixel. The benefit of using this approach is that each time data is coded the data is coded on a new pattern that makes the coding of data very efficient.

3.4. Decoding data from image

For decoding textual data in the image, the hash key is used that was generated to during coding. By using the hash-key the used algorithm (described in section 4) generates the exactly same pattern that was used at the time of coding. The pixels (red, green, blue byte) values of each position are read one by one and generated characters concatenated to form a complete message.

4. Hash based algorithm

The most important part of the proposed algorithm is the used hashing technique. We have used the perfect hashing as hash-function (H) algorithm [13]. A function for perfect hashing is defined for set N to map distinct elements in N to distinct integers, without any collisions. A perfect hash function supports efficient lookups by placing hash-keys from N to a hash-table. [14] A few implementations for perfect hash functions are available. There are number of advantages of using perfect hashing over other hashing techniques. Some of those are following:

- Perfect hashing is fast than other techniques as it avoids any hash collision [13], [14]. Hence, there is no need to use any collision resolution techniques (such as linear probing or quadratic probing) as collision resolution is an overhead and involves intensive computation.
- Perfect hashing technique supports very large key sets [16] so we can use it for very bulky data sets and it is

equally effective and efficient for large data sets as for small data sets.

The algorithm steps for coding and decoding text into an image is described separately. The major steps of the used algorithm are given below.

4.1. Algorithm for coding

Following steps of the algorithm were used to code data into the image.

(i)Input a text (.txt) file containing textual data and input (Jpg, .gif .bmp or .tiff) image.

(ii) Read text file, tokenize the text and make chunks of the text of 3 characters each and store each chunk of data in an array-list (A_e).Total count of data chunks are represented as n.

(iii) Generate n random number using Linear Congruential Generator that is used as a hash key and hash-key is represented using h

(iv) The hash-function H [12] uses the hashkey (h) and total number of chunks (n) to generate a pattern i.e. sequence of numbers (hash-values) those are position of the pixels where data will be stored.

(v)The generated pattern (containing sequence of numbers) is stored in an array-list (A_p).

(vi) First chunk from A_e and A_p are read. The ASCII value of 1st character of chunk $A_c[i]$ is replaced with the red-byte of the $A_p[i]$, similarly, ASCII value of 2nd character of chunk $A_c[i]$ is replaced with the green-byte of the $A_p[i]$, and the ASCII value of 3rd character of chunk $A_c[i]$ is replaced with the blue-byte of the $A_p[i]$.

(vii) The output is the image containing coded data.

4.2. Algorithm for decoding

Following steps of the algorithm were used to decode data from the image.

(i) Input the (.jpg, .gif .bmp or .tiff) image that contains that coded information and the hash key(h) that was actually used to code data.

(ii) The input hash-key (h) is used with the hash function (H) to generate sequence of numbers as an array-list (A_p) and these numbers are actually position of the pixels where data was stored. Here the hash-function (H) generates specifically same pattern of random numbers for a hash-key (H) those were generated at the time of coding.

(iii) Each value from the generated patterns represent index of a pixel where the data is saved. Values of read, green and blue byte are read at $A_p[i]$ are read. As each byte contains an ACSII value of a character, the read ASCII value is converted to a character and each character is written to a text file in sequence it is read from the image.

(iv) The output is a text file that contains the decoded data from image.

5. Result and discussions

The proposed method is applied on different colour images with different size and format. The system is implemented in MATLAB.The implemented system is used to embed secret data in the form of textual file or image onto a colour image of format .bmp, .jpg, .gif, .png. The system proved to be compatible to all the image file format and yielded good results. A screen shot of the original image used to hide data is shown in Figure.1 and image with hidden data is shown in Figure.2. From the simulation results it was very clear that there is no degradation in the image quality even after embedding a data and human eye can never detect any difference between the original image and image with hidden data. We experimented with different examples to validate the performance of our system the length of smallest data set was 10 words, while the length of largest used data set for experimentation was 1200 words.

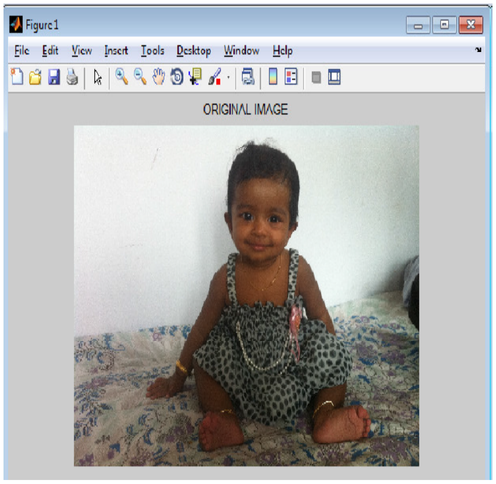

Figure 1 Original image

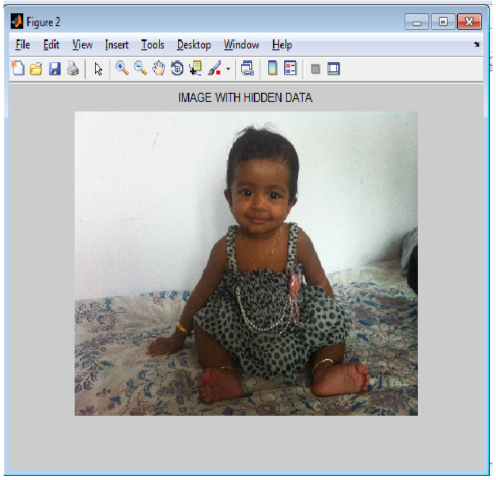

Figure 2 Image with hidden data

6. Conclusion

The designed system has robust ability to read spatial information from a (bmp, git: jpeg, and tif) image and hide image and hide data in the image without losing the quality of the image. This system specifically works for efficient and secure data hiding in images to make possible large-sized data encryption and transmission over internet. The proposed approach is fully automated. We have presented the initial experiments with the perfect hashing based algorithm based approach for colour image steganography. However, the used algorithm can be improved to get better and accurate results.

7. References

[1] F. Shih, Digital watermarking and steganography, fundamentals and techniques. USA: CRC Press, 2008.
[2] K. Usman, H. Juzoji, I. NakajiIm, S. Soegidjoko, M. Ramdhani, T. Hori, and S. Igi, "Medical image encryption based on pixel arrangement and random permutation for transmission security," in Proceedings of IEEE 9th International Conference on e-Health Networking,Application and Services, Taipei, Taiwan, 2007, 19-22 June.pp.244-247.
[3] U. Gopinathan, D.S. Monaghan, T.I Naughton, IT. Sheridan, and B. Javidi, "Strengths and weaknesses of optical encryption algorithms," in Proc. 18th Annual Meeting of the IEEE Lasers and Electro-Optics Society, 2005, 22-28 Oct pp.951-952.
[4] Cheddad, Abbas; Condell, Joan; Curran, Kevin; McKevitt, Paul, AHash-based Image Encryption Algorithm, Optics
Communications, Volume 283, Issue 6, p. 879-893. March, 2010
[5] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," IEEETransactions on Image Processing, 15(7X2006)2061-2075.
[6] M. Zeghid, M Machhout, L Khriji, A Baganne, and R Tourki,"A modified AES based algorithm for image encryption,"International Journal of Computer Science and Engineering,1(IX2006) 70-75.
[7] Y. Wang, X Liao, D. Xiao, and K.W. Wong, "One-way hash function construction based on 2D coupled map lattices,"Information Sciences, 178(5X2008)1391-1406.
[8] V. Patidar, N.K Pareek, and K.K Sud, "A new substitutiondiffusion based image cipher using chaotic standard and
logistic maps," Communications in Nonlinear Science and Numerical Simulation, 14(7)(2009) 3056-3075.
[9] D.C. Lou and C.R Stmg, "A steganographic scheme for secure comnnmications based on the chaos and Euler theorem," IEEE Transactions on Multimedia, 6(3X2004)501-509.
[10] I. Ahmad and AS. Das, "Hardware implementation analysis of SHA -256 and SHA -512 algorithms on FPGAs,"Computers & Electrical Engineering, 31(6X2005)345-360.
[11] I Wen, M. Severa, and W. Zeng, "A format-compliant configurable encryption framework for access control of video," IEEE Trans. Circuits Syst. Video Technol, 12(6X2002)545-557.
[12] Thomas R Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein Introduction to Algorithms, Second Edition MIT Press and McGraw-Hill, 2001. Section 11.5: Perfect hashing, pp. 245-249.
[13] W. P. Yang and M. W. On, "A Dynamic Perfect Hash Function Defined by an Extended Hash Indicator Table", in proceedings of I (/' International Coriference of Very Large Databases, VLDB'84, Singapore, August, 1984, pp: 245-254
[14] David Talbot, Thorsten Brants, "Randomized Language Models via Perfect Hash Functions", in proceedings of ACLHLT 2008, Ohio, USA, June 2008, pp:505-513