International Journal of Information Technology and Knowledge Management
July-December 2009, Volume 2, No. 2, pp. 419-423

A NOVEL SESSION BASED TEXT ENCRYPTION & HIDING TECHNIQUE USING BIT LEVEL CROSS FOLD TRANSPOSITION & GENETIC ALGORITHM

Tanmay Bhattacharya[*], Manas Paul[**] & Arindam Dasgupta[***]

This paper provides a novel approach towards document hiding technique within a color bitmap image using cross fold transposition and genetic algorithm. The document can be a text, word or any other text file. The secret text is firstly represented in its equivalent binary form, upon which cross fold transposition is applied. Next this binary form is perturbed by genetically generated session-key. In the next step this perturbed information is embedded within the Host-image. During extraction the stego-image and the original image along with the session key is used. At first the extraction of the secrete text is done, and then reverse engineering with the session-key is done to obtain the hidden information.

Keywords: Bit Level, Cross Fold, Transposition, Session Key, Genetic Algorithm, Steganography.

1. INTRODUCTION

Today the most important thing in communication via computer networks–is the need for securing information–and this need increasing rapidly. Our prime concern is to protect data from unauthorized access. This work gives a new steganographic [2, 7, 9, 10, 11] approach of hiding text or document information within a color bitmap host image in such a way that the changes are not noticed by normal human eye even comparing the stego-image with the original image. More over the "stego–image" size remains same as that of the original image. Secret text is perturbed in two different steps. In the first step-text is perturbed by "cross fold transposition" [1, 3, 4, 5, and 8]. The encrypted output file is once again perturbed by a "session-key" which is Genetically [12] obtained by applying different permutation techniques [3, 4, 5, 6]. Finally this information is stored in the host image by either changing the pixel values or by keeping the pixel values intact. Diagrammatic explanation is given below:

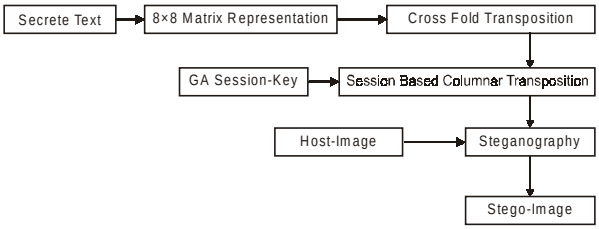


Figure 1: Building the Stego-Image Containing the Hidden Information

JIS College of Engineering, Kalyani, West Bengal, INDIA
[*] Information Technology Dept.,
Email: tanmay_bhattacharya@yahoo.co.in
[**] Computer Application Dept., *Email: manaspaul@rediffmail.com*
[***] *Email: adg7sat0071@yahoo.co.in*

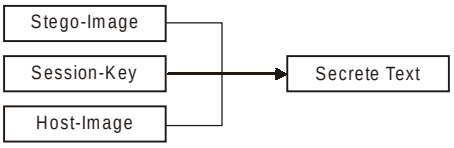


Figure 2: Retrieving Hidden Information from the Stego-Image

2. STEGANOGRAPHY

The word steganography is of **Greek** origin and means "concealed writing". Greek word Steganos, means covered or secret and graphy means writing or drawing. This is a technique of hiding of information in such way that its presence cannot be detected. Steganography and Cryptography are completely two different things.

Cryptography—the science of writing in secret codes—addresses all of the elements necessary for secure communication over an insecure channel, namely privacy, confidentiality, key exchange, authentication, and non-repudiation. But cryptography does not always provide safe communication.

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**.

The following formula provides a very generic description of the pieces of the steganographic process:

cover_medium + hidden_data + stego_key = stego_medium

3. TEXT ENCRYPTION

Algorithm for 1st stage Perturbation of Text is as follows:

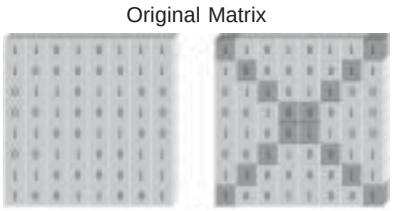
- Enter 8 bit session-key. The session-key is genetically generated, eight digit number,

containing digits 0 to 7 without any repetition of digits within a key. Store the key in an array; say "key".

- Encode text file into its corresponding bit format and store in a file and pad zero's to this bit format for eight-cross-eight array representation.
- Repeat the following steps until end of the file (storing the bits) is reached.
  - Open a file "*F*" to store the encrypted form.
  - Read the file (storing bit format) to form a eight-cross-eight matrix.
  - Apply cross fold transposition upon this matrix. Firstly the data of r^{th} row and c^{th} column is swapped with the data of ***(7–r)th row and cth column***. Secondly the data of r^{th} row and c^{th} column is swapped with the data of ***rth row and (7–c)th column.***
  - Use the session-key for columnar transposition, such that the data from the matrix starting with r^{th} row and (session-key [c^{th}]) column is written in the file "*F*".
- Close the "*F*" file.

Example:

The matrix is cross-fold in the following way:



[Dark portions depicts diagonals of the matrix along which Cross-fold transposition is performed]

The transpositions are as follows:

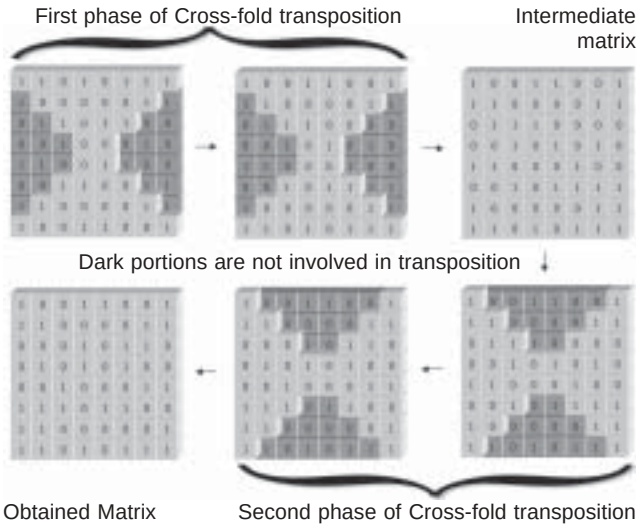


Figure 3

4. GENETIC ALGORITHM

Here the genetic algorithm is used to generate the session-key. The technique applied is as follows:

a. We first find out the all possible combination that can be formed by arranging numbers ranging from 0 to 7; without any reoccurrence of any digits within a number/sequence and without any reoccurrence of any number previously generated and then store the numbers in a two dimensional array.

b. Next we take the sum of difference between each consecutive digit within a number/sequence. If this sum is greater than a predefined value then the number is stored as a specimen. This process is repeated for until all possible combination of sequences is checked.

c. From the specimens obtained by employing above process we take specimen and perform mutation in them by swapping digits of any two positions. This process is repeated for all the specimens.

d. Now from the specimens obtained by the immediate above process we form groups–containing two specimens only. Now in each group the specimen / chromosome(s) are crossover at a particular position. A check is kept for avoiding reoccurrence of digits in a number.

e. Next again sum of difference between each consecutive digits / genes within a sequence / chromosome. If this sum is greater than a predefined value then the sequence /chromosome is stored as a specimen for next generation. This process is repeated for until all possible combination of sequences / chromosome is checked.

f. The above mentioned process is repeated for a pre defined number of times to obtain the ***near maximum disordered sequence***.

After cross fold transposition we apply columnar transposition on the matrix in following way:

Let Session–key generated by using the above mentioned genetic algorithm be '73201564'.

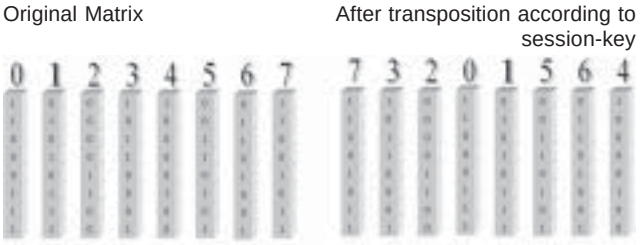


Figure 4

5. HIDING TEXT WITHIN IMAGE

Algorithm to hide the text within the host image and to form the stego-image:

- Suppose, for example, that two adjacent image pixels are (252, 253, 254, 0) and (252, 253, 254, 0) and we wish to hide the data "*A* = 65".
- Convert "*A*" to its bit format. Here we obtain (10000010).
- For first pixel i.e. (252, 253, 254, 0)

  (252, 253, 254, 0) – [{(252, 253, 254, 0) %2} XOR (1, 0, 0, 0)]

  (252, 253, 254, 0) – [(0, 1, 0, 0) XOR (1, 0, 0, 0)]

  (252, 253, 254, 0) – (1, 1, 0, 0)

  (251, 252, 254, 0)
- For second pixel i.e. (252, 253, 254, 0)

  (252, 253, 254, 0) – [{(252, 253, 254, 0) %2} XOR (0, 0, 1, 0)]

  (252, 253, 254, 0) – [(0, 1, 0, 0) XOR (0, 0, 1, 0)]

  (252, 253, 254, 0) – (0, 1, 1, 0)

  (252, 252, 253, 0)
- The new pixels are written in the new picture file i.e. (251, 252, 254, 0) and (252, 252, 253, 0).
- This means that 1 byte of data is hidden per 8 bytes of image data. (0.5 bytes per pixel).
- The merit of this algorithm is that, the alpha channel is also included.

6. EXTRACTION OF HIDDEN INFORMATION

To extract the hidden information % both the stego-image and the original image along with the session-key is send to the receiver. Next two pictures are compared pixel by pixel. Where there is a mismatch, the pixel from the stego-image is taken. Next upon this pixel value we apply modulo 2 division operation and the out come of the operation is stored in an eight-cross-eight array format. Next when the array is full the session-key is applied for columnar transposition and to retrieve the original information or matrix. This process is repeated until entire picture files are compared.

7. IMPLEMENTATION

This entire work is implemented by using JAVA 1.6.

The image used here has the following properties:

- Type of file: BMP File.
- Width: 732 pixel.
- Height: 486 pixel.
- Horizontal Resolution: 96 dpi.
- Vertical Resolution: 96 dpi.
- Bit Depth: 24.
- Frame Count: 1.

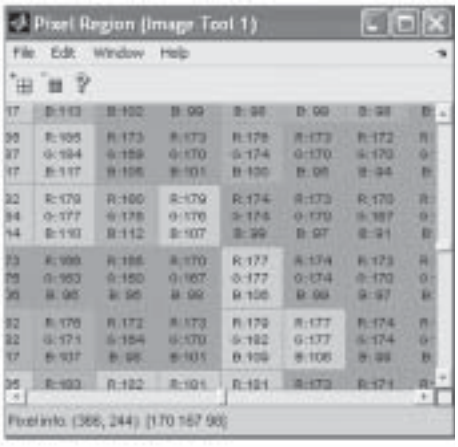
The text used here has the following properties:

- Type of file: Text Document.
- Size: 105 KB (108,242 bytes).

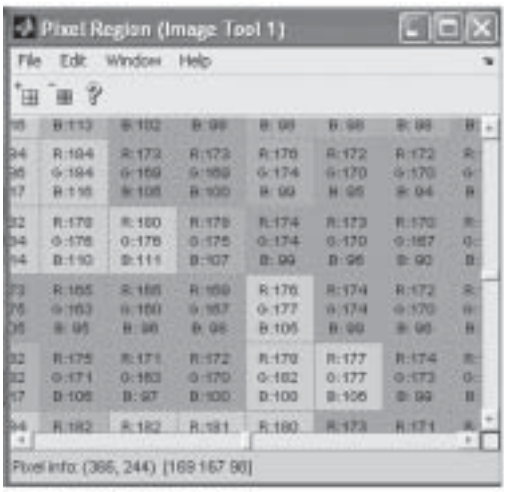
Host Image [1.01 MB.]

Stego-image [1.01 MB.]

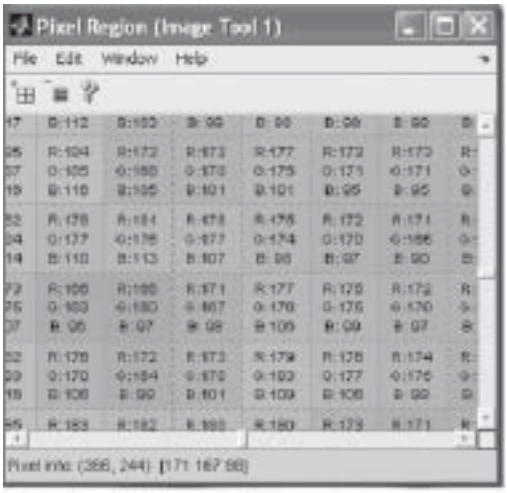
Portion of Secret Text File [105 KB]



A Portion of Host-Image Matrix

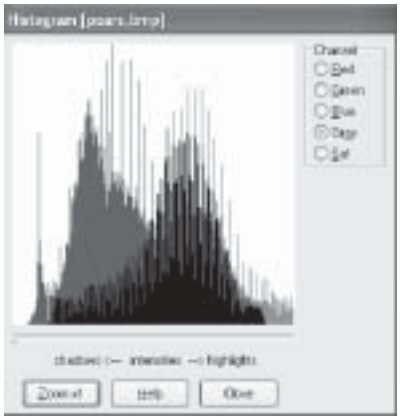


A Portion of Stego-Image [using S-Tool]

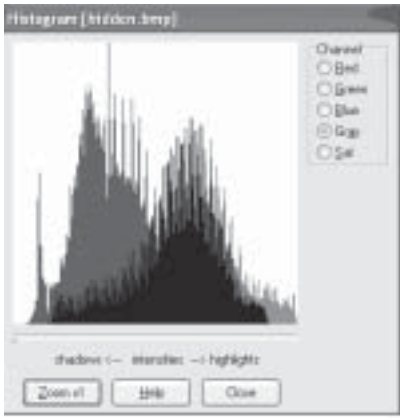


A Portion of Stego-Image [using Proposed Algorithm]

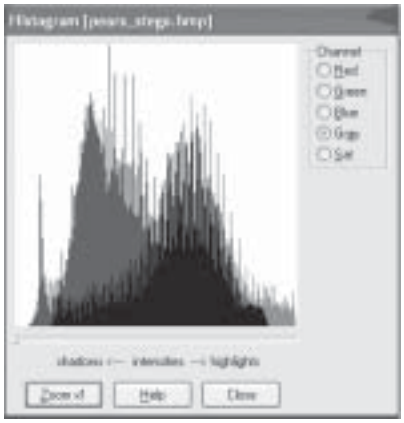
Figure 5



a.
Histogram of Host-Image



b.
Histogram of Stego-Image [using S-Tool]



c.
Histogram of Stego-Image [using Proposed Algorithm]

Figure 6

8. CONCLUSION & FUTURE SCOPE

There is enough scope of betterment of this work but comparing with the popular Steganographic tool, "S-Tool" there is no doubt that Histogram Analysis and Matrix Analysis results of this approach is quite satisfactory. Since the alpha channel is also utilized in this proposed algorithm we can hide more amount of text in an image than compared to "S-Tool". Moreover the "Cross Fold" encryption technique proposed here is satisfactorily effective. There is enough scope of applying advanced Steganographic approach to hide the encrypted text and enhanced security can be achieved.

9. REFERENCES

[1] T. Bhattacharya, T. K. Bhattacharya & S. R. B. Chaudhuri "*A General Bit Level Data Encryption Technique using Helical & Session Based Columnar Transpositions*" Proceedings of an IEEE International Advance Computing Conference (**IACC'09**) (2009) 364-368.

[2] T. Bhattacharya, S. Bhowmik & S. R. B. Chaudhuri *"A Steganographic Approach by Using Session Based Stego-Key, Genetic Algorithm and Variable Bit Replacement Technique"*: Proceedings of International Conference on Computer and Electrical Engineering, 2008. [ICCEE 2008], 51-55.

[3] S. Contini and Y. L. Yin. "*On Differential Properties of Data Dependent Rotations and their use in MARS and RC6*" In Proceedings of Second AEF Conference (2008).

[4] Z. Shi and R. B. Lee. "*Implementation Complexity of Bit Permutation Instructions*". In Proceedings of the Asilomar Conference on Signals, Systems, and Computers, (2003).

[5] Z. Shi, X. Yang, and R. B. Lee. "*Arbitrary Bit Permutations in One or Two Cycles*". In Proceedings of the 14th International Conference on Application-Specific Systems, Architectures and Processors, (2003) 237–247.

[6] R. B. Lee, Z. Shi, and X. Yang. "*Efficient Permutation Instructions for Fast Software Cryptography*". *IEEE Micro*, **21**(6) (2001) 56–69.

[7] Provos N. and Honeyman, P., "*Detecting Steganographic Content on the Internet*", Center for Information Technology Integration, University of Michigan. Technical Report (2001).

[8] Z. Shi and R. B. Lee. "*Bit Permutation Instructions for Accelerating Software Cryptography*". In Proceedings of the 11 International Conference on Application-Specific Systems, Architectures and Processors, (2000) 138–148.

[9] Sellars. D., An Introduction to Steganography", *Ics.uct.ac.za / courses / CS400W / NIS / papers99 / dsellars / stego.html.*

[10] PETITCOLAS, F. A. P., ANDERSON, R. J., and KUHN, M.G. : '*Information Hiding–A survey*', Proc. IEEE, **87**(7) (1999) 1062-1078.

[11] VAN SCHYNDEL, R. G., TIRKEL, A. Z., and OSBORNE, c.F.: "*A Digital Watermark*". Inl. ConT. Image Processing, (1994) 86-90.

[12] D. E. Goldberg, "*The Genetic Algorithms in Search, Optimization, and Machine Learning*", New York: Addison-Wesley (1989).