Steganographic Methods Based on Digital Logic

PARVINDER SINGH[1], SUDHIR BATRA[2], HR SHARMA[3]
[1]Department of Computer Science & Engineering
Guru Jambheshwar University, Hisar(India)
http://www.parvinder.50megs.com
[2]Department of Mathematics
Technological Institute of Textile & Sciences, Bhiwani (India)
[3]Department of Computer Science & Engineering
CSIT, Durg (India)

Abstract:- We are proposing steganographic methods based on digital logic [1]. These methods select the images depending upon the information to carry. So transmission of huge information with minimal effect on carrier images is possible. Bit replacements made in host images are negligible compared to existing embedding techniques [2-9]. We have used digital operations based on logic gates and shift operators to derive the hidden information from image data.

Keywords- Steganography, Hiding Information, Digital Logic, Image Selection

1 Introduction

Steganograhy is name given to techniques used for hiding information in digital objects such as image, video or sound files etc. Hiding information in digital images open wide spectrum of applications such as – Secure communication, copy- right protection, and data authentication. Research in steganography has shown that bit replacement or bit substitution is inherently insecure with safe capacities far smaller than previously thought. For secure communication, hidden information is strictly restricted to fraction of cover image file [10-12]. An upper bound of 0.005 bits/pixel was experimentally determined for safe LSB embedding by Jessica et all [13-14]. Proposed methods select the images based on digital logic operations such that image is not significantly degraded after embedding and embedded information is immune as possible to modifications from intelligent attacks or manipulations.

2 State of Art Methods

We are here proposing three methods –

1. Logic Gate Method
2. Shift Operator Method
3. Combined Method

2.1 Logic Gate Method

Logic gates AND, OR, XOR, and NOT are used on image matrix to derive the information matrix in this method.

2.1.1 Insertion Method

The information to be hidden and image file are read as rows of bits matrices such that no of columns in both matrices are equal. Logic gate operations such as AND, OR, XOR and NOT are used to get the rows of information matrix from image matrix rows. We are using following op-codes for logic operations-

Logic Operation 'Op'	**Op-Code**
AND	00
OR	01
XOR	10
NOT	11

These logic operations are applied one by one to all possible combinations of image rows to get each row of information matrix. Then embedded data in image for each row of information matrix will be *op-code, row address, and row address*. This technique can be well understood by following example. Suppose size of image matrix is 256 * 256 bits and size of information to be hidden is 1Kb. We can write this in information in matrix of 4 * 256, so that number of columns is equal in both image and information matrix. Further suppose first row of information matrix is derived by applying AND operation on 7^{th} and 13^{th} row

of image matrix. Then embedded data for first row (first 256 bits) of information matrix will be **00**0000011**00001101**. Here first two bits 00 specify the op-code of AND operation, next two bytes 00000111 and 00001101 specify the row addresses of image matrix on which AND operation is applied (7 and 13 in this example). We can see only 18 bits are require to embed in place of 256 bits and in total 72 bits are require to hide 1Kb of information in this example. If it is not possible to derive all the rows of information matrix from image matrix, we will select another image for embedding. It should also be noted that-

1. In case of NOT operation, only one row address of image matrix is needed, which will make the embedded data further minimal. (10 bits in place of 256 bits in above example)
2. If we will choose an image matrix of having comparatively more columns then embedded information will decrease but selection of image will become more difficult. Similarly if we will choose an image matrix of having comparatively more rows then selection of image will be easy but embedded information will increase. In above example, if we will choose an image matrix of 64*1024 bits for embed information of 1024 bits then only 14 bits (2 bits for up-code and 6 bits each for two row addresses) are required. However selection of image will become difficult.
3. If K^{th} row of information matrix is same as J^{th} row of image matrix, then either AND or OR operation can be applied on J^{th} row itself to derive embed information.
4. If K^{th} row of information matrix contains all zeros then it can be derived by applying XOR operation on any row of image matrix with itself.

It is possible to hide information, which is high in size then image itself as the size of hidden information is much less than original information to be hide. A recursive approach of the same can be applied on embedded information to make it of desired size.

The data flow diagram of the Logic Gate Method is shown in figure 1. The various blocks in the diagram are intended to represent the logic steps involved in the method.

The image file and information to be hidden are represented in matrix format of having equal number of columns and sent to Logic Function Deriver block which derives the rows of information matrix from rows of image matrix after applying logic gate operation. After all the rows of information matrix are derived, Embedded Information Converter generates the Boolean functions in form of op-code, row address, and row address. The Boolean function is checked in Size check comparator for further applying Logic Gate Method recursively to minimize the embedded information to desired size.

Once the desired size embedded information is achieved, it is appended with number of recursions “R” of the method to give Master Bit Pattern (MBP). If this Master Bit Pattern is brought down to “m” bits, we embed just “m” bits instead of the M bits of information matrix (M>>m).

The Master Bit Pattern thus obtained has to be safely guarded since any bit change in the MBP would lead to distorted retrieval of the hidden information. Hence obtained bit pattern is coded using Turbo Code [15-17].

The MBP is copied to encoder1 and encoder2. Before entering the encoder2, MBP is scrambled by the interleaver. Each encoder generates a string of error correction bits (parity bits) by performing a series of calculation on the data bits it receives. The original data and the two strings of parity bits are combined in to a single block are embedded on to the image in embedding area. This final turbo encoded message is embedded using an existing efficient embedding techniques gives the “pseudem cover file” as output.

This Logic Gate Method leads to two important aspects -

1. Image File is seldom distorted.
2. For maximum utility multiple information files can be pseudo-embedded till the maximum safe limit.

2.1.2 Extraction Method

The extraction method is reasonably simple. From the agreed bit locations (both at the transmitting and receiving ends) in the image, the turbo-encoded message is extracted, which is decoded to get the Master Bit Pattern. The decoded operation is as follows:

The received analog signal corresponding to the secure MBP is sampled and assigns integers indicating how likely it is a '0' or a '1'. For example, -7 mean certainly a zero and +7 mean a certain '1'. Note that an error occurred in the 5^{th} bit in the block as highlighted in figure 3. Originally a '1', it now has a negative value, which suggests it is a logical zero which is the error that has occurred.

Each Decoder takes noisy data and respective parity information and computes how confidence it is about each decoded bit. The two decoders exchange this confidence information repeatedly and after a number of iterations typically around 4 to 10; they begin to agree on the decoded bits. The decoded data is sum of noise and data plus the two strings of confidence value. The output is converted back to binary digits.

Once the Master Bit Pattern has been got by after decoding, the Boolean functions are framed with op-codes, row addresses and size of information. The Boolean functions are evaluated and hence information is retrieved.

The pseudem cover file at receiver end is passed in to the Master Bit Pattern extractor (a combination of encoded message extractor and turbo decoder) that receives the MBP. This pattern enters the Bits to function converter, which checks the value of recursion "R" from end of the MBP. If no recursion is used at transmission end then it is null and function converter directly converts the MBP to Boolean functions. Depending upon the value of "R" output of function converter is again sent as input to the Bits to Function converter and after similar procedures we get next higher level bit pattern and recursion continue until the final Boolean functions are obtained. This recursion ends when recursion null comparator returns a null (This occurs when the decrement operation on the recursion counter results a null value). The final Boolean function along with the image file is derived from output, which in turns returns the information hidden in image.

2.2 Shift Operator Method

Shift operators are generally used in serial transfer of data in digital systems. During the shift operations, if serial input transfers a bit into left most position, it is called shift left operation. If the serial input transfers a bit into right most position, operation is called shift right operation. The information transferred through serial input determines the type of shift. There are three types of shifts: logical, circular, and arithmetic.

A logic shift is one that transfers 0 through the serial input. We will adopt the symbol SHL and SHR for logical shift-left and shift right operations.

The circular shift (also known as rotate operation) circulates the input bits around the two ends without loss of information. We will adopt the symbols CIL and CIR for the circular shift left and right, respectively.

An arithmetic shift operator shifts a signed binary number to the left or right. It is not used in our technique.

The information to be hidden and image file are read as rows of bits matrices such that number of columns in both matrices are equal. Shift operators such as SHL, SHR, CIL and CIR are used to get the rows of information matrix from image matrix rows. We are using following op-codes for shift operations-

Shift Operation 'Op'	**Op-Code**
SHL	00
SHR	01
CIL	10
CIR	11

In this method, rows of image matrix are shifted to derive the rows of information matrix. Then embedded data in image for each row of information matrix will be *op-code, row address, and numbers of shifts.* Let us take the same example of embedding 4*256 bits (1Kb) of information matrix in to image of size 256*256. If shifting 15 bits of 32nd row of image derives any row of

information matrix and shift is circular left, then embedding for that information row will be **10**00100000**00001111**. Here first two bits specify the CIL operation, next eight bits specify the row address of image matrix (32 in this example) and next eight bits (15 in this example) specify the number of bits shifted. We can see only 18 bits are require to embed in place of 256 bits and in total 72 bits are require to hide 1Kb of information in this example. If it is not possible to derive all the rows of information matrix from image matrix, we will select another image for embedding. The embedded information can be turbo encoded at transmission end and turbo decoded at receiving end for safe transmission as described in logic gate method.

2.3 Combined Method

This method is combination of logic gate and shift operator method. In this scheme, we can use either logic gates or shift operators on image matrix to derive information matrix. Due to option of applying more operations on image matrix, there is more chance of selecting an image for embedding information. The op-codes used in this scheme for various operations are-

Operation 'Op'	**Op-Code**
OR Gate	000
AND Gate	001
XOR Gate	010
NOT Gate	011
SHL	100
SHR	101
CIL	110
CIR	111

The cost of using this scheme is one extra bit in Op-code to accommodate more operations on image matrix. This scheme can also be applied in conjunction with turbo encoding and turbo decoding as used in Logic Gate Method.

3 Conclusion

We can hide very huge information in image with very less embedding. Experiments shows that, change in image (bits/pixel) remain near zero with our methods, when it crosses the upper limit of safe embedding (0.005 bits/pixel) in LSB method. Moreover the embedding can be more than image data and effect of hiding is very less distortion to image. Another advantage of these methods is – Images, which are more suitable for particular embedding, can be selected. These methods use turbo encoding for error correcting, which is best technique for this propose in digital communication.

References

[1] M Morris Mano, *Computer System Architecture, 3rd Edition*, Printence Hall, 1998.

[2] Neil F. Johnson, Sushil Jajodia, "Exploring Steganography: Seeing the Unseen", *IEEE Computer*, Feb 1998, pp 26-34.

[3] Neil F. Johnson, Sushil Jajodia, "Steganalysis of Images Created Using Current Steganography Software", *Lecture Notes in Computer Science*, vol 1525, 1998, Springer-Verlag.

[4] JJ Eggers, R Bauml, Bernd Grid, "A Communication Approach to Image Steganography", Proceedings of SPIE vol 4675, Jan 2002, *Security and Watermarking of Multimedia Contents IV*, San Jose, Callifornia.

[5] Parvinder Singh, Sudhir Batra, HR Sharma, "Evaluating the Performance of Message Hidden in 1st and 2nd Bit Plane", *WSEAS Transactions on Information Science and Applications*, issue 8, vol 2, Aug 2005, pp 1220-1227.

[6] Parvinder Singh, Sudhir Batra HR Sharma, "Hiding Credentials in Biological Images", *A and B Research*, Vol 22 no 1, Jan 2006, pp 22-25.

[7] SN Sivanandan, CK Gokulnath, K Prasanna, S Rajeev, "NFD Techniques for Efficient and Secured Information Hiding in Low Resolution Images", *Lecture Notes in Computer Sciences*, vol 3347, Springer Verlag, 2004, pp 458-467.

[8] S Katzenbeisser, FAP Petitcolas, "*Information Hiding Techniques for Steganography and Digital Watermarking*", Artech House, 2000.

[9] NF Johnson, S Katzenbeisser, "A Survey of Steganographic Techniques", *Information Hiding*, Artech House, pp 43-78, 2000.

[10] R Chandramouli, Nasir Memmon, "Analysis of LSB based Image Steganography Techniques", *Proccedings of ICIP 2001*, Greece, Oct 2001, pp 1019-1022.

[11] R Chanramouli, "A Mathematical Framework for Active Steganalysis", *ACM Multimedia Systems Journal*, 2003.

[12] RJ Anderson, FAP Petitcolas, "On the Limits of Steganography", *IEEE Journal of Selected Areas in Communication*, Special issue 16 no 4, 1998, pp 474-481.

[13] J Fridrich, MGoljan, R Du, "Reliable Detection of LSB Steganography in Grayscale and Color Images", *Proccedings of ACM Workshop on Multimedia and Security, Canada*, Oct 200, pp 27-30.

[14] J Fridrich, MGoljan, R Du, "Detecting LSB Steganography in Color and Grayscale Images", *IEEE Multimedia, Nov 2001*, pp 22-28.

[15] MR Titchener, "Digital Encoding by Means of New T Codes to provide Improved Data Synchronization and Message Integrity", *IEE Proceedings, Computer Digital Technology*, 1984, pp 151-153.

[16] C Berrou, Glavieux, P Thitimajshima, "Near Shannon Limit Error Correcting Coding: Turbo Codes", Proceedings of IEEE Conf. On Communication, Geneva, Switzerland, 1993, pp 1064-1070.

[17] Anil Kumar, Navin Rajpal, "Application of T-Code, Turbo Codes and Pseudo-Random Sequence for Steganography", *Journal of Computer Science 2(2)*, 2006, pp 148-153.

List of Figures-

1. Process Flow Cycle of Logic Gate Method
2. Turbo Encoder Block with Sample MBP
3. Turbo Decoder Block for Extraction of the Master Bit Pattern
4. Data Flow Diagram for Extraction Technique
5. Comparison of Logic Gate Method with LSB and Upper Limit for Safe Embedding in Different Image Sizes

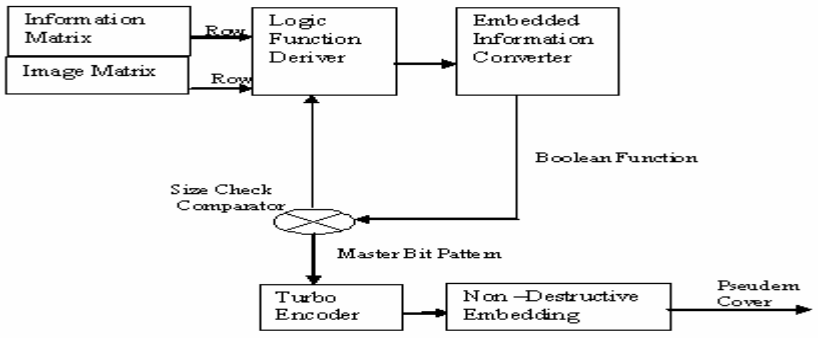

Figure 1 Process Flow Cycle of Logic Gate Method

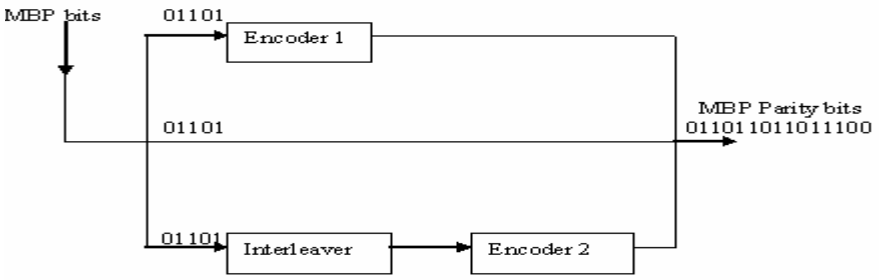

Figure 2 Turbo Encoder block with sample MBP

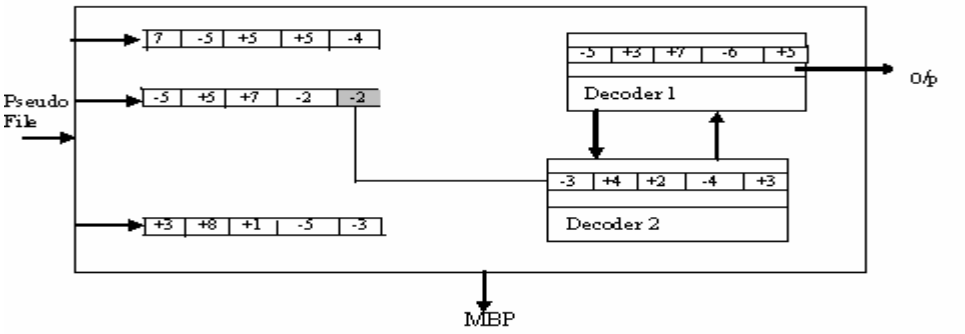

Figure 3 Turbo decoder block for extraction of the Master Bit Pattern

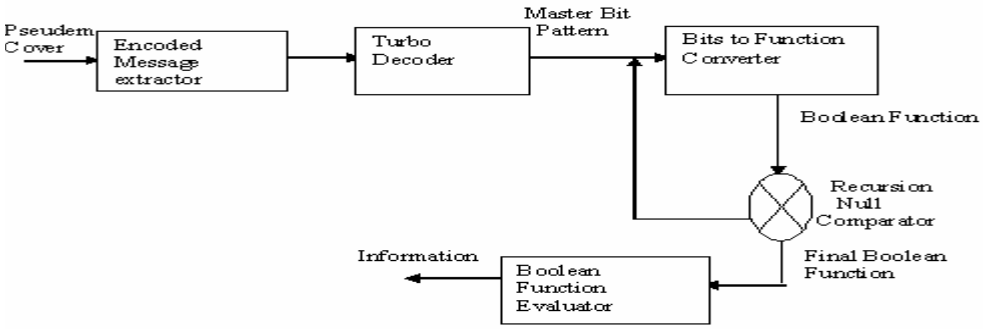

Figure 4 Data Flow Diagram for Extraction Technique

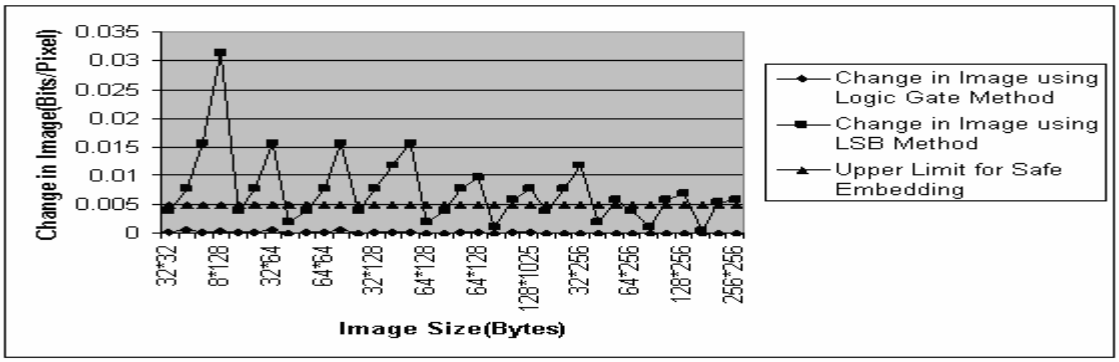

Figure 5 Comparison of Logic Gate Method with LSB and Upper Limit for Safe Embedding in Different Image Sizes