Tree Based Parity Check Scheme for Data Hiding

[1] Sampath Kumar Dara, [2] Harshavardhan Awari
*1 Assoc. Professor, Department of Electronics and Communication Engineering, Vaageswari College of Engg, Karimnagar, A.P
*2 Assoc.Professor, Department of Computer Science Engineering, Vaageswari College of Engg, Karimnagar, A.P.

Abstract
The information hiding deals with distortion reduction using steganography and security enhancement using cryptography. Distortion reduction is done using Tree Based Parity Check which uses Majority vote strategy. The Tree Based Parity Check is very optimal for cloaking a message on image. The proposed majority vote strategy results in least distortion. The SHA-1 algorithm is implemented for security enhancement. The result obtained in proposed method works effectively even with large payload.

Key Words— Image coding, information security, Stenography.

I. I NTRODUCTION

Stenography studies the scheme to hide secrets into the communication between the sender and the receive r such that no other people can detect the existence of the secrets. A steganographic method consists of an embedding algorithm and an extraction algorithm. The embedding algorithm describes how t o hide a message into the cover object and the extraction algorithm illustrates how to extract the message from the stego object. A commonly used strategy for steganography is to embed the message by slightly distorting the cove r object into the target stego object. If the distortion is sufficiently small, the stego object will be indistinguishable from the noisy cove r object. Therefore, reducing distortion is a crucial issue for steganographic methods. In this paper, we propose an efficient embedding scheme that uses the least number of changes over the tree-based parity check model.

The ideas of matrix embedding and defined the codes with the matrix as steganographic codes. For matrix embedding, finding the stego object with least distortion is difficult in general. In some special cases, there exist constructive and fast methods. LT Codes to improve the computational complexity of wet paper codes derive d a hash function to efficiently obtain the stego object. Proposed a scheme called tree-based parity check.

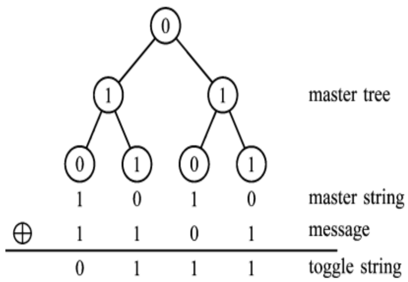

Fig. 1. Master and toggle strings of a master tree with for LSBs 0, 1, 1, 0, 1, 0, 1 o f the cove r object. (TBPC) to reduce distortion on a cover object based on a tree structure.

The embedding efficiency is defined to be the number of hidden message bits per embedding modification. Higher embedding efficiency implies better undetectability for steganographic methods. The lower embedding efficiency is defined to be the ratio of the number of hidden message bits to the maximum embedding modifications. The lower embedding efficiency is related to undetectability in the worst case. It implies steganographic security in the worst case.

II. TBPC METHOD

Before embedding and extraction, a location finding method determines a sequence of locations that point to elements in the cover object. The embedding algorithm modifies the elements in these locations to hide the message and the extraction algorithm can recover the message by inspecting the same sequence of locations. The TBPC method is a least significant bit (LSB) steganographic method. Only the LSBs of the elements pointed by the determined locations are used for embedding and extraction. The TBPC method constructs a complete *N*-ary tree, called the master tree, to represent the LSBs of the cover object. Then it fills the nodes of the master tree with the LSBs of the cover object level by level, from top to bottom and left to right. Every node of the tree corresponds to an LSB in the cover object. Denote the number of leaves of the master tree by *L*. The TBPC embedding algorithm derives an *L*-bit binary string, called the *master string*, by performing parity check on the master tree from the root to the leaves (e.g., see Fig. 1.). The embedding algorithm hides the message by modifying the bit values of some nodes in the master tree. Assume that the length of the message is also L. Performing the bitwise exclusive-or (XOR) operation between the message and the master string, we obtain a toggle string (e.g., see Fig. 1). Then, the embedding algorithm constructs a new complete N-ary tree, called the toggle tree in the bottom-up order and fills the leaves with the bit values of the toggle string and the other nodes with 0. Then, level by level, from the bottom to the root, each nonleaf node together with its child nodes are flipped if all its child nodes have bits 1 (e.g., see Fig. 2). The embedding algorithm obtains the stego tree by performing XOR between the master tree and the toggle tree (e.g., see Fig. 3). The TBPC extraction algorithm is simple. We can extract the message by performing parity check on each root-leaf path of the stego tree from left to right.

III. Majority Vote Strategy

Two critical issues for a steganographic method are:
1) Reducing distortion on cover objects

2) Better efficiency for embedding and extraction. We give a majority vote strategy on building the toggle tree. It uses the least number of 1's under the TBPC model. Since the number of 1's in the toggle tree is the number of modifications on the master tree (i.e., the cover object), the majority vote strategy can produce a stego tree with least distortion on the master tree. The proposed method uses a set of standard measures, to capture image properties before or after the embedding process, which effect the performance of steganalysis techniques. These measures are divided into two categories. First cover image properties, and second cover-stego based distortion measures.

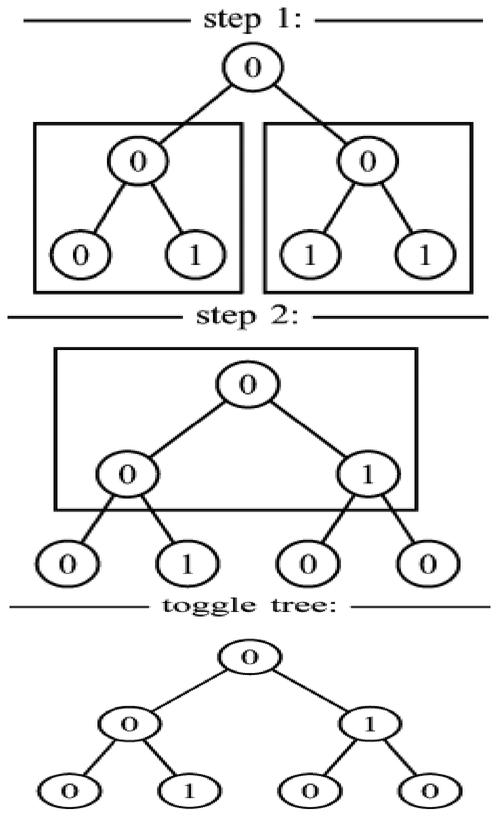

Fig. 2. Construction of a toggle tree with for toggle string 0, 1, 1, 1.

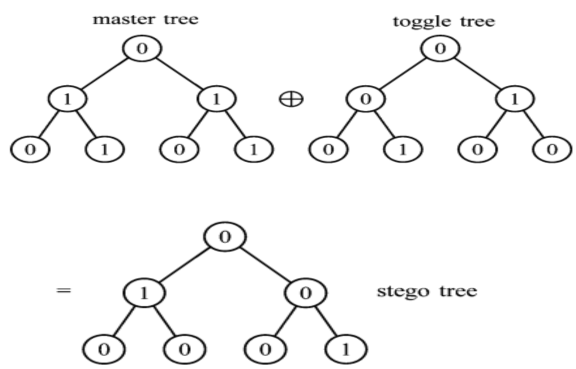

Fig. 3. Modify the master tree into the stego tree by the toggle tree constructed from the toggle string 0, 1, 1, 1.

First, index all nodes of a complete -ary tree with leave s from top to bottom and left to right. Set the bit toggle string bit by bit into the leave s from left to right and the other nodes 0. Assume that the level of the tree is. Traverse all nonleaf nodes from level 1 to. A nonleaf node and its child nodes form a simple complete subtree. For each simple complete subtree, if the majority of the child nodes hold 1, then flip the bit values of all nodes in this subtree. Since the construction is bottom-up, the bit values of the child nodes in every simple complete subtree are set after step 3. Note that marking a node at step 4 applies only for being even. When is even, after step 3, there may exist a two level simple complete subtree with 1's in the child nodes and 1 in its root. In this case, flipping the bit values in this simple complete subtree results in one fewer node holding 1 and keeps the result of related root-leaf path parity check unchanged. Step 4 takes care of this when the condition applies, and it is done level b y level from top to bottom. Also note that for the root of the whole toggle tree, the bit value is always 0 when half of its child nodes hold 1. Thus, after step 4, the bit values of the child nodes in each simple complete subtree are determined.

IV. Experimental Results

To make it clear, we define the percentage of reduced modifications as follows:

p Reduce = Rt/Dt

Where Rt is the reduced number of 1's in the toggle tree and Dt is the number of 1's in the toggle string. The p Reduce values of both methods are shown in Fig. 4.

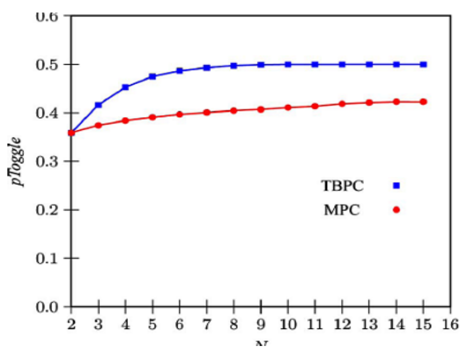

Fig.4. p Toggle comparison of MPC and TBPC.

The results show that the MPC method significantly improves previous TBPC results.

V. Conclusion

By introducing the majority vote strategy, the stego object is constructed with least distortion under the tree structure model. We also show that our method yields a binary linear stego-code and preserves the secrecy of the hidden data. In comparison with the TBPC method, the proposed MPC method significantly reduces the number of modifications on average.

References

[1] S. Chang Chun Lu, Shi-Chun Tsai, and Wen-Guey Tzeng, " An Optimal Data Hiding Scheme With Tree-Based Parity Check" in IEEE Trans. on Image Processing, vol. 20, no. 3, Mar 2011.

[2] J. Fridrich, M. Goljan, and D. Soukal, —Efficient wet paper codes," in Proc. 7th Int. Workshop Information Hiding (IHW 05), vol. 3727, pp. 204–218, Apr 2005.

[3] J. Fridrich and D. Soukal, —Matrix embedding for large payloads," in IEEE Trans. on Information Forensics and Security, vol. 1, no. 3, pp. 390–395, Sep. 2006.

[4] M. Khatirinejad and P. Lisonek, —Linear codes for high payload steganography," in IEEE Trans. on Discrete Applied Mathematics, vol. 157, no. 5, pp. 971–981, 2009.

[5] W. Zhang and X. Wang, "Generalization of the ZZW embedding construction for Steganography," in IEEE Trans. on Information Forensics and Security, pp.564-569, Jan 2009.

[6] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, —Writing on wet paper," in IEEE Trans. on Signal Processing., vol. 53, no. 10, pp. 3923–3935, Oct. 2005.