

Data Hiding in Images using Majority Vote Parity Check

Jyoti Rao

Computer Engineering
D.Y.Patil Institute of
Engineering and technology,
Pimpri, Pune

Kalpna More

Computer Engineering
D.Y.Patil Institute of
Engineering and technology,
Pimpri, Pune

Swati Nikam

Computer Engineering
D.Y.Patil Institute of
Engineering and technology,
Pimpri, Pune

ABSTRACT

The Internet as a complete does not use safe links, thus data in transit may be vulnerable to interception as well. The significant of reducing a chance of the data being detected during the conduction is being a problem now days. Some way out to be discussed is how to pass data in a manner that the very existence of the message is unknown in order to prevent interest of the possible attacker. This approach of data hiding can be extended to copyright protection for digital media. This paper present a comparative study of majority vote parity check method and tree-based parity check. Result shows that Majority vote strategy results in least distortion for finding a stego object.

General Terms

Security, Algorithms, Data Hiding.

Keywords

Majority vote, Steganography, Image processing, Least significant bit (LSB).

1. INTRODUCTION

In this digital world, with the recent advances in computing technology and its intrusion in our everyday life, the necessity for confidential and personal communication has increased. This is the reason steganography is introduced for secured communication. This paper introduces the idea of steganography using a new Majority vote parity check algorithm, which produces negligent distortion as compared to the Tree based parity check.

In Tree Based parity Check, embedding algorithm obtains the stego tree by performing XOR between the master tree and the toggle tree. The TBPC extraction algorithm is simple but not efficient because it extracts the message by performing parity check on each root-leaf path of the stego tree from left to right. To reduce the distortion on cover objects and better efficiency for embedding and extraction Majority Vote Parity check is used. Also Encryption algorithm is used to encrypt secret information for enhancing the security.

A commonly used Method for steganography is to embed the message by slightly distorting the cover object into the target stego object. If the distortion is acceptably small, the stego object will be impractical to differentiate from the noisy input image. Thus, minimizing alteration is a vital issue for steganography methods. This paper proposes an efficient embedding scheme that uses the least number of changes over the tree-based parity check model. Strategy of majority vote is used to relax the toggle criteria of a node. MPC Method creates a stego object with least alteration under the tree based parity check model. The time complexity of embedding and extraction algorithm is best, because, it is linearly enclosed by the hidden message length. The embedding effectiveness is defined as the amount of hidden message bits per embedding alteration. For better Undetectability of steganography methods higher embedding efficiency requires. The lower embedding efficiency can be defined as the ratio of the number of hidden message bits

to the maximum embedding modifications. The lower embedding efficiency indicates that Undetectability in the worst case. Thus, for a steganography system we can say that the lower embedding effectiveness is an important safety factor. In this technique, it is $2 - \Theta(1/L)$, where L is the hidden message length and $\Theta(1/L)$ is a set of functions enclosed both above and below by $1/L[1]$.

2. RELATED WORK

J. Fridrich et. al. showed that the communication channel recognized as writing in memory with defective cells is a relevant information-theoretical model for a specific case of passive warden steganography when the sender embeds a secret message into a subset C of the cover object X without sharing the selection channel C with the recipient. "Writing on wet paper" is nothing but in this method the set C could be arbitrary, found by the sender from the cover object using a deterministic, pseudo-random, and truly random process. Realizes it using low-density random linear codes with the encoding step based on the LT process. J. Fridrich et. al. discussed about the importance of writing on wet paper for covert communication within the context of adaptive steganography and a new approach for digital media called perturbed quantization. However, writing on wet paper leaves most of the flippable pixels unused and only a fraction of the embedding capacity is used [2].

For producing steganographic schemes with an arbitrary selection channel that means not shared between the sender and the recipient J. Fridrich et. al. proposed Wet paper codes as a tool . These codes improved embedding efficiency which is nothing but number of message bits per embedding change and improved for low computational complexity. J. Fridrich et. al. described a new approach(the matrix LT process) to wet paper codes using the apparatus developed for irregular low-density parity check erasure codes called LT codes. This approach offers greatly simplified implementation and a substantially decreased computational complexity. J. Fridrich et. al. also presented a few simple modifications of the matrix LT process to improve the embedding efficiency while preserving its low computational complexity. Additionally, J. Fridrich et. al. introduced another different approach to wet paper codes called Block Minimal embedding that provides significantly improved embedding efficiency and also enjoys low computational complexity suitable for steganographic applications and data embedding[3].

J. Fridrich and D. Soukal proposed Matrix embedding is a general coding method that can be applied to most steganographic schemes to improve their embedding efficiency which is nothing but the number of message bits embedded per one embedding change. As smaller number of embedding changes is less likely to disrupt statistic properties of the cover object. Matrix embedding generally has better steganographic security. J. Fridrich and D. Soukal proposed two new approaches to matrix embedding that are suitable when the embedded message length is close to the embedding capacity.

First approach is depending on random linear codes of small dimension. Random linear codes provide good embedding efficiency. Also, it covers the range of large payloads, which makes these codes suitable for practical applications. The second approach proposed for large payloads. Structured codes are more efficient and it is used even for relative payloads more than 0.9. Their performance is however not as good as for the random codes for small payloads [4].

W. Zhang et.al. Proposed a new method to create stego-codes, for showing that not just one but a family of stego-codes can be generated from one covering code by combining Hamming codes and wet paper codes. Performance of stego-code families of structured codes and random codes are analyzed by using the stego-code families of LDGM codes. W. Zhang et.al. Obtained a family of near optimal embedding schemes, which can approach the upper bound of embedding efficiency for various chosen embedding rate. Zhang proposed a new technique for applications in steganography to create embedding schemes, which can produce a family of stego-codes from one covering code. After combining this steganography method with random codes such as LDGM codes, we can get a family of near optimal stego-codes for arbitrarily chosen embedding rates. To oppose discovery, the sender can always reduce changes to the cover by embedding fewer messages into an image, i.e., use low embedding rate. However, recent improvements in steganalysis have made LSB steganography with small embedding rates visible. LSB steganography is less secure than ± 1 steganography because ± 1 embedding can avoid the statistical difference introduced by LSB exchange. Larger embedding efficiency can be obtained with the modified SCFs, so ± 1 steganography plus the modified SCFs will provide even better security [5].

J. Fridrich analyzed asymptotic behavior of the embedding construction for steganography proposed by Zhang[5]. The embedding efficiency of codes from the ZZW embedding construction [5] follows the upper bound on embedding efficiency. J. Fridrich noted that the embedding construction for ± 1 embedding also proposed in [5] approaches the bound on embedding efficiency of ternary codes with the same limit, because the ternary bound increases by 1 compared to the binary bound. The embedding effectiveness of ± 1 ZZW code families is also larger by 1 [6].

R.Y. M. Li et.al. Proposed a new algorithm to achieve improvement in visual quality that can be applied to most of the existing data hiding algorithms called as tree based parity check (TBPC). In data hiding process, original image is modified then distortion is created. Almost existing data hiding algorithms try to minimize the visual artifacts introduced by the modifications. The probability of modifying the original host image is reduced by the proposed algorithm. R.Y.M.Li et.al. Proved that an increase in visual quality is achieved in the watermarked image [7].

Based on Tree based Parity Check Chung-Li Hou et.al. Proposed a Majority vote strategy that results in least alteration for finding a stego object. MPC technique shows improved result for lower embedding efficiency than that of previous works when the hidden message length is relatively large. Majority vote strategy effectively constructs the stego object with least distortion under the tree structure model. In comparison with the TBPC method, Majority vote strategy method significantly reduces the number of modifications on average [1].

3. TREE BASED PARITY CHECK

Location finding method is used to determine a sequence of locations that point to elements in the cover object. To hide the message the embedding algorithm modifies the elements in these locations and the extraction algorithm can recover the message by searching the same series of locations. The Tree Based Parity Check method is the Least significant bit (LSB) steganographic technique. In this method Only the LSBs of the elements pointed by the determined locations are used for embedding and extraction. The TBPC method produces a complete N-ary tree, called as master tree, to represent the LSBs of the input image that is nothing but cover object. Then we fills the nodes of the master tree with LSBs of the cover object level by level, from left to right and top to bottom. L is the number of leaves of the master tree (e.g. see Figure 1). The TBPC embedding algorithm derives master string which is nothing but an L-bit binary string, by performing parity check on the master tree from the root to the leaves.

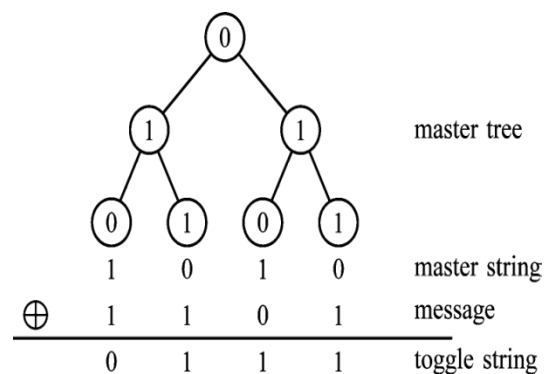


Figure 1. Master and toggle strings of a master tree with L=4 for LSBs 0, 1, 1, 0, 1, 0, 1 of the cover object.

We can hide the message by modifying the bit values of some nodes in the master tree using the embedding algorithm. Assume that L is the length of the message. Toggle string created by performing the bitwise exclusive-or (XOR) operation between the message and the master string (e.g. see Figure 1).

Then we create a complete N-ary tree which is nothing but toggle tree using embedding algorithm in the bottom-up order then use bit values of the toggle string to fills the leaves and put 0 to the other nodes. For each nonleaf node together with its child nodes we check that if all its child nodes have bits 1 then flipped the value of its parent and child from the bottom to the root, level by level. (e.g. see Figure2). By performing XOR between the master tree and the toggle tree we can create stego tree (e.g. see Figure3). The Tree Based Parity Check extraction algorithm is used to extract the message by performing parity check on each root to leaf path of the stego tree from left to right.[1].

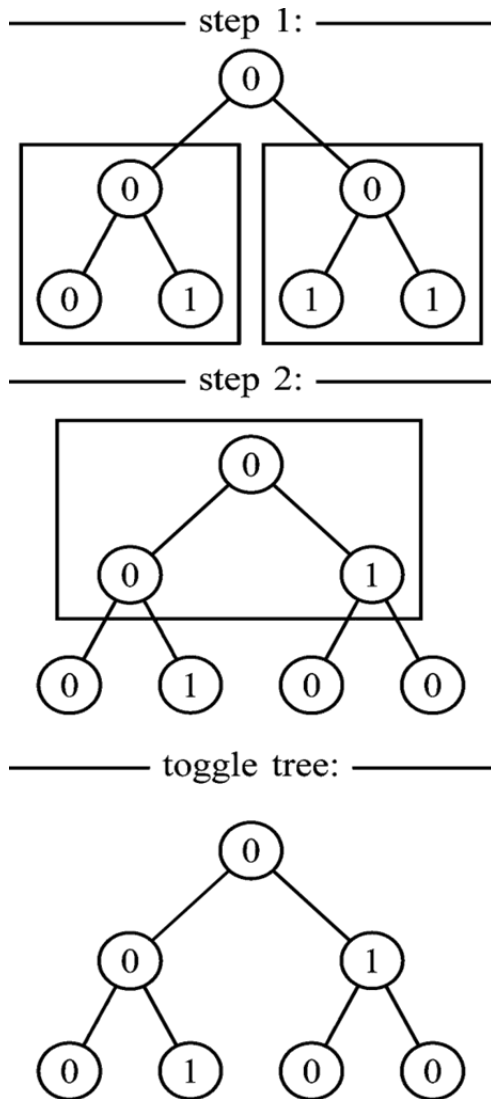


Figure 2. Construction of a toggle tree with $L=4$ for toggle string 0, 1, 1, 1.

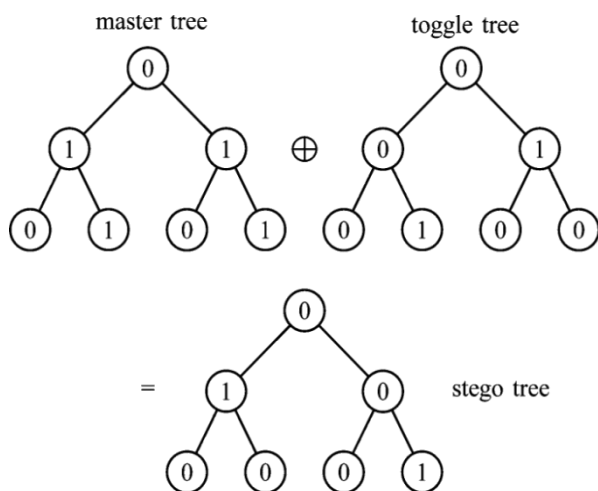


Figure 3. Modify the master tree into the stego tree .

4. MAJORITY VOTE PARITY CHECK

Two critical issues for a steganographic method are: 1) reducing distortion on cover objects and 2) better efficiency for embedding and extraction. A majority vote strategy is used on building the toggle tree for resolving these two issues. MPC method uses the least number of 1's as compare to the TBPC model. As the number of 1's in the toggle tree is nothing but the number of modifications on the master tree (i.e., on the input image), the MPC can produce a stego tree with least distortion on the master tree. Majority vote parity check [MPC] is used due to its use of majority vote in calculating the parity bit. MPC construct the toggle tree with the least amount of 1's level by level in the bottom-up order using the following steps [1].

Algorithm MPC:

Input: a toggle string of length L ;

1. Index the nodes of the initial toggle tree;
2. Set the leaves of the toggle tree from left to right & bit by bit with the toggle string & the other nodes 0;
3. for $i=1$ to h
for each internal node on level i do
if the majority of its unmarked child nodes holds 1
then flip the bit values of this node & its child nodes;
else if the numbers of 0 & 1 in its unmarked child nodes are the same
then mark this internal node;
4. if N is even then
for $i=h-1$ for 1
for each marked internal node holding 1 on level i do
flip the bit values of this node & its child nodes;

5. EXPERIMENTAL RESULT

MPC and TBPC methods are implemented for a comparison between their pToggle values. MPC and TBPC are implement for different type of images with different number of child. Result show that MPC is always better than TBPC for $N > 4$. When $N=2$ and 3 they are the same. To make it clear, we define the percentage of reduced modifications as follows:

$$pReduce = \frac{Rt}{Dt}$$

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.

5.1 Result of MPC and TBPC for BMP

Image:

The pReduce values and pToggle values of both methods for BMP image are shown in Table 1 and Table 2 respectively.

Table 1. Experimental Results of pReduce

pReduce Results		
NodeValue	TBPC	MPC
2	46.45	41.99
3	12.88	71.3
4	7.73	45.87
5	4.58	66.97
6	2.75	45.43
7	1.13	64.51
8	1.61	48.92
9	0.69	62.19
10	0.47	47.13
11	0.4	62.22
12	0.27	45.46
13	0.28	61.61
14	0	47.27
15	0	59.83

Table 2. Experimental Results Of pToggle

pToggle Results		
NodeValue	TBPC	MPC
2	0.0189	0.0194
3	0.0271	0.0271
4	0.0351	0.0338
5	0.0434	0.0387
6	0.055	0.0505
7	0.0443	0.0385
8	0.0177	0.0147
9	0.0625	0.0543
10	0.0572	0.0489
11	0.05	0.0433
12	0.0358	0.0343
13	0.0276	0.0238
14	0.0116	0.0096
15	0.0683	0.0591

The pReduce and pToggle comparison of MPC and TBPC for BMP image are Shown in Figure 4 and Figure 5 respectively.

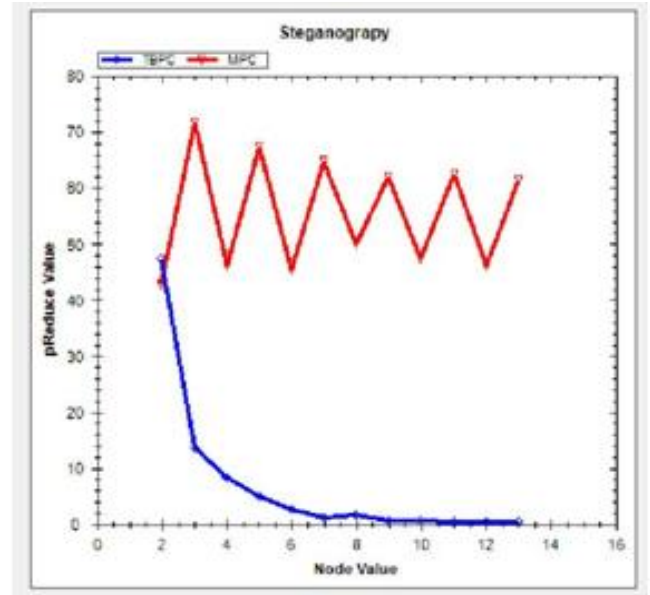


Figure 4. . pReduce comparison of MPC and TBPC

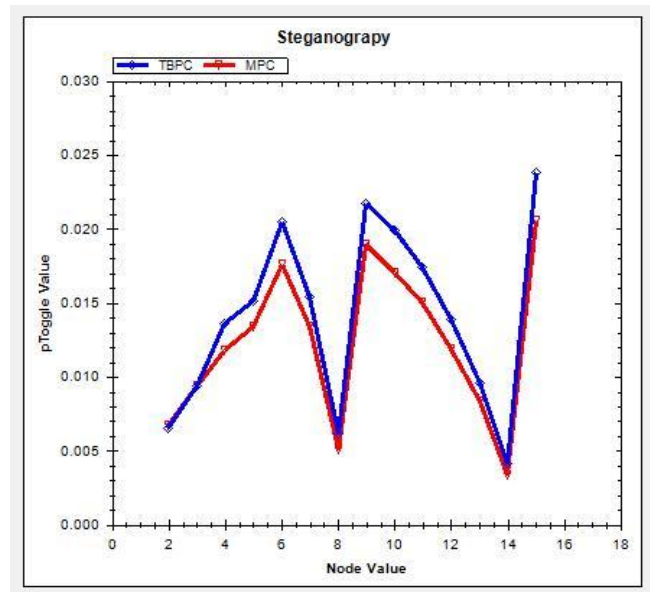


Figure 5. pToggle comparison of MPC and TBPC

It is easy to construct a method that achieves the expected embedding modifications per hidden bit of 0.5. In other words, if we try to embed an L -bit message into the cover object, $0.5 L$ modifications will occur on average.

$$pToggle = \frac{Da}{L}$$

We use to indicate the probable embedding modifications per hidden bit, where Da is the average number of embedding modifications for an L -bit message.

5.2 Result of MPC and TBPC for PNG Image:

The pReduce values and pToggle values of both methods for PNG image are shown in Table 3 and Table 4 respectively.

Table 3. Experimental Results of pReduce

pReduce Results		
NodeValue	TBPC	MPC
2	66.33	66.33
3	45.61	81.12
4	27.24	78.96
5	20.6	80.24
6	21.31	75.15
7	19.26	77.34
8	26.64	73.45
9	17.97	78.7
10	16.28	75.94
11	16.42	75.48
12	17.06	73.17
13	20.8	77.75
14	19.65	72.74
15	14.05	74.9

Table 4. Experimental Results of pToggle

pToggle Results		
NodeValue	TBPC	MPC
2	0.0006	0.0006
3	0.0007	0.0007
4	0.0012	0.0009
5	0.0013	0.001
6	0.0017	0.0012
7	0.0013	0.0009
8	0.0005	0.0004
9	0.0018	0.0012
10	0.0017	0.0011
11	0.0014	0.0009
12	0.0012	0.0008
13	0.0008	0.0006
14	0.0004	0.0003
15	0.002	0.0013

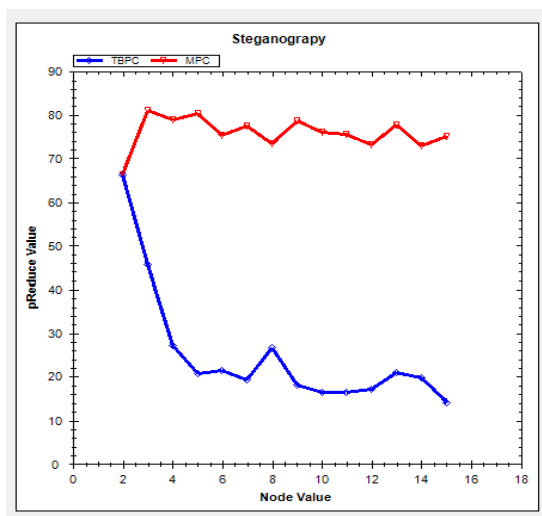


Figure 6. . pReduce comparison of MPC and TBPC

The pReduce and pToggle comparison of MPC and TBPC for PNG image are Shown in Figure 6 and Figure 7 respectively.

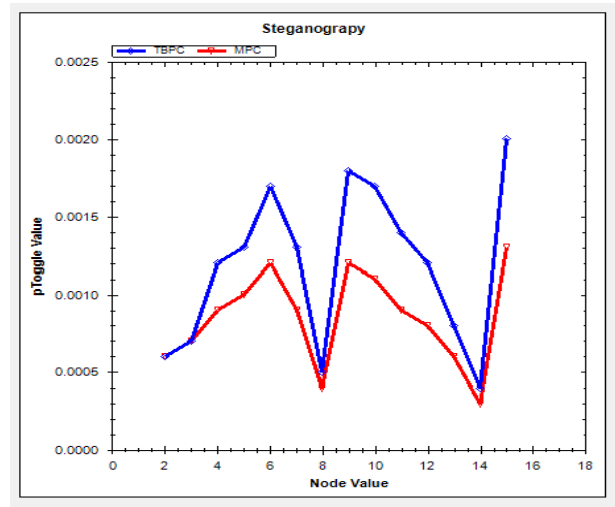


Figure 7. pToggle comparison of MPC and TBPC

5.3 Result of MPC and TBPC for JPEG image

The pReduce values and pToggle values of both methods for JPEG image are shown in Table 5 and Table 6 respectively.

Table 5. Experimental Results of pReduce

pReduce Results		
NodeValue	TBPC	MPC
2	45.52	45.52
3	24.25	74.84
4	12.4	49.84
5	8.3	68.08
6	5.65	49.24
7	3.11	65.46
8	2.12	49.77
9	1.6	63.78
10	1.26	49.54
11	0.74	62.28
12	0.54	49.04
13	0.34	61.06
14	0.28	50.32
15	0.15	60.38

Table 6. Experimental Results of pToggle

pToggle Results		
NodeValue	TBPC	MPC
2	0.0181	0.0181
3	0.0606	0.0606
4	0.0664	0.0573
5	0.0577	0.0512
6	0.1042	0.0871
7	0.0164	0.0142
8	0.1277	0.106
9	0.0956	0.0808
10	0.0462	0.0387
11	0.1532	0.1297
12	0.1462	0.1229
13	0.137	0.1165
14	0.126	0.1057
15	0.1112	0.0947

The pReduce and pToggle comparison of MPC and TBPC for

JPEG image are Shown in Figure 8 and Figure 9 respectively.

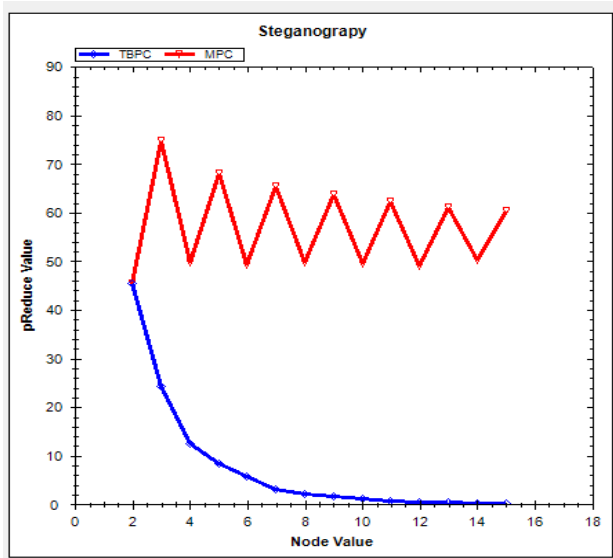


Figure 8. . pReduce comparison of MPC and TBPC

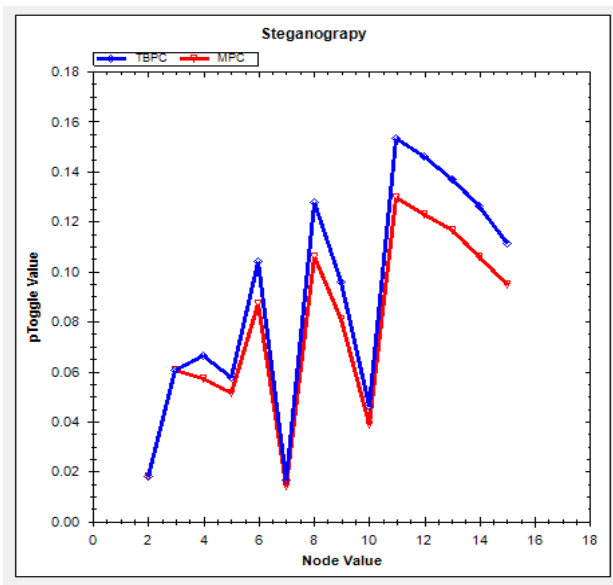


Figure 9. pToggle comparison of MPC and TBPC

6. CONCLUSION

This paper presents comparative study of two steganographic techniques Tree based parity check and Majority vote parity check for different type of images. Result shows that majority vote parity check has least distortion as compare to tree based parity check. Because MPC method reduces the number of modifications on input image. Due to tree structure model Majority vote strategy effectively construct the stego object. The time complexity of embedding and extraction algorithm of Majority vote parity check is $O(L)$.

7. REFERENCES

- [1] Chung- Li Hou, ChangChun Lu, Shi-Chun Tsai, and wen-Guey Tzeng., " An Optimal Data Hiding Scheme With Tree-Based Parity Check", IEEE Trans.On Image processing, vol.20, No. 3, March 2011.
- [2] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper", IEEE Trans. Signal Process., vol. 53, no. 10, pp. 3923-3935, Oct. 2005.
- [3] J. Fridrich, M. Goljan, and D. Soukal, "Efficient wetpaper codes", in Proc.7th Int.Workshop Inf. Hiding (IHW 05), Lecture Notes in Computer Science 2005, vol. 3727, pp. 204-218.
- [4] J. Fridrich and D. Soukal, "Matrix embedding for large payloads", IEEE Trans. Inf. Forensics Security, vol. 1, no. 3, pp. 390-395, Sep.2006.
- [5] W. Zhang, X. Zhang, and S. Wang, " Maximizing steganographic embedding efficiency by Combining hamming codes and wet paper codes", in Proc. Int. Workshop Inf. Hiding (IH 08), 2008, vol. LNCS 5284.
- [6] J. Fridrich, "Asymptotic behavior of the ZZW embedding construction", IEEE Trans. Inf. Forensics Security, vol. 4, no. 1, pp. 151-154, Mar. 2009.
- [7] R.Y. M. Li, O. C. Au, K. K. Lai, C. K. Yuk, and S.- Y. Lam, "Data hiding with tree based parity check", in Proc. IEEE Int. Conf. Multimedia and Expo (ICME 07), 2007, pp. 635-638.
- [8] A. Westfeldl, "F5: A steganographic algorithm, high capacity despite better steganalysis", in Proc. 4th Int. Workshop Inf. Hiding., 2001, vol. LNCS 2137, pp. 289-302.
- [9] W. Zhang and S. Li, "A coding problem in steganography", Designs, Codes Cryptogr., vol. 46, no. 1, pp. 68-81, 2008.