J Inf Process Syst, Vol.15, No.2, pp.261~270, April 2019
https://doi.org/10.3745/JIPS.03.0111

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Block-Based Adaptive Data Hiding Approach Using Pixel Value Difference and LSB Substitution to Secure E-Governance Documents

Tanmoy Halder*, Sunil Karforma**, and Rupali Mandal***

Abstract

In order to protect secret digital documents against vulnerabilities while communicating, steganography algorithms are applied. It protects a digital file from unauthorized access by hiding the entire content. Pixel-value-difference being a method from spatial domain steganography utilizes the difference gap between neighbor pixels to fulfill the same. The proposed approach is a block-wise embedding process where blocks of variable size are chosen from the cover image, therefore, a stream of secret digital contents is hidden. Least significant bit (LSB) substitution method is applied as an adaptive mechanism and optimal pixel adjustment process (OPAP) is used to minimize the error rate. The proposed application succeeds to maintain good hiding capacity and better signal-to-noise ratio when compared against other existing methods. Any means of digital communication specially e-Governance applications could be highly benefited from this approach.

Keywords

Block, E-Governance, Pixel Value Difference, Steganography

1. Introduction

The world is digitally growing with rapid implementation of government to consumer (G2C) e-Commerce. Countries use it as a common communicating medium while connecting with people. A number of e-Governance projects are running successfully to fulfill the need of communication. Steganography algorithms are easier to implement and common potential methods which prevent unauthorized access to shared documents from unknown parties thus, protecting the originality.

Steganography is the process of hiding any valuable secret information, in digital form, within a digital medium. It is a practice which is being used over a few decades. Cryptography is another mode of secret data communication. However, it differs from steganography in a single sense. For cryptography if a secret message is hidden, detection is easily followed by mathematical comparison. But, steganography hides the encryption as well as the process of encryption. Steganography uses several digital documents as a cover like image files, video files, music files etc.

The least significant bit (LSB) replacement approach is one of the most widely used approaches in

Manuscript received April 13, 2015; first revision June 18, 2015; accepted July 7, 2015.
Corresponding Author: Tanmoy Halder (tanmoyhalder@gmail.com)
* Dept. of Computer Application, Dr. B. C. Roy Engineering College, Durgapur, India (tanmoyhalder@gmail.com)
** Dept. of Computer Science, The University of Burdwan, Golapbag, Burdwan, India (sunilkarforma@yahoo.com)
*** Dept. of Computer Application, Bengal College of Engineering and Technology, Durgapur, India (rupa.mandal23@gmail.com)

steganography due to its nature of simplicity. Any binary bit stream holds the most negligible bits in its LSB position. Altering or replacing that bit, doesn't affect the actual value too much [1-3]. Bailey and Curran [4] in 2003 analyzed seven different LSB-based steganography methods. Methods are Stego1bit, Stego2bits, etc. It covered almost all basic types of LSB steganography techniques including grayscale and color images. Chan and Cheng [3] in 2004, proposed substitution of an equal number of bits from LSB side of each pixel for hiding secret message. Chang et al. [5] in 2002 and Thien and Lin [6] in 2003 established the fact that LSB substitution is a very common steganography process which directly hides a secret message in the LSB bit of a pixel.

Halder and Karforma [7,8] introduced the concept of indexing in the existing LSB matching process. Their proposed method doesn't hide secret bits directly within the LSB positions. Rather, the secret data stream is scattered within the bits other than LSB positions and LSB bits are used to maintain an index. The index is to remember the positions.

In 2003, Wu and Tsai [9] proposed a new way of hiding secret bits within the intensity gap between the neighboring pixels and named it pixel value difference (PVD). In PVD, a block pairing two consecutive pixels is created. The pixel value is known as the intensity of that pixel. The intensity gap is used for embedding new messages. If the intensity difference is high, the bigger secret message stream could be embedded, whereas smaller gaps are occupied by a message having less size in terms of volume. This adaptive adjustment process successfully hides number of bits engaging almost the same amount of pixels when compared against LSB methods. It is more practical to engage three consecutive pixels at a time rather than two pixels when selecting blocks. The introductory idea was suggested by Chang et al. [10]. This high capacity hiding mechanism was named as "try-way pixel value differencing" (TPVD). In TPVD, a non-overlapping block of four pixels is chosen. From those four pixels, three different pairs consisting of two pixels are selected at a time. Secret bits are embedded within those pairs with the help of PVD. For embedding purpose hybrid approach is applied, which combines LSB and PVD.

Secret data hiding following the adaptive mechanism could be implemented applying a combination of LSB substitution along with PVD. Khodaei and Faez [11] have implemented this adaptability mainly over grayscale images where color intensity varies from 0 to 255. Selection of blocks engaging three pixels instead of two pixels is a wise decision to utilize more gaps. Three LSB bits of the middle pixel are substituted/embedded applying the normal substitution method. Therefore, two adjusting pixels within the block are embedded using PVD. While applying PVD, the middle one is taken as a common pixel and participates in both the adjustments. Optimal pixel adjustment process (OPAP) is applied to minimize the percentage of error.

Khodaei and Faez's proposal was further modified to implement flexibility by Tsai et al. [12]. Here, in the proposed approach block size is variable and depends on the volume of the secret message stream. In this high capacity adaptive embedding technique, non-overlapping blocks of varying size may be identified for embedding. The size is denoted by M and N. Value of M and N ranges from 1 to 7. The proposed method induced flexibility against a selection of blocks, depending on the type of cover images. It also shows better results for smaller size blocks, i.e. when the value of M and N is low. On the other hand, with increased block size, capacity and peak signal-to-noise ratio (PSNR) degrades for a noticeable amount. Especially for block size 7×7 and 6×6. When capacity is too high (more than 900000 bits) PSNR degrades below 30 dB, which tends to a very low quality stego image.

In our proposal, the implemented method overcomes the problem arising in Chan's proposal [12]. We have restricted the value of *M* and *N* between 1 and 3 to create small size blocks. We also modified the sub-division of the range allocation system. The entire range is segmented into two different sub parts of the basic of embedding strategy. While embedding, due to modification in pixels, out of range values is modified again. The existing PVD adjustment process is also reconstructed for embedding purpose. Obtained result experimented with standard images reflects that the proposal succeeds to achieve better capacity and PSNR for all blocks. Each of the blocks in the proposal reaches the same or sometimes better capacity and better signal to noise ratio when compared to Chan's method [12].

The rest of this paper is organized as follows: Section 2 discusses the Chan's method [12] in brief; Section 3 is about the proposed method; Section 4 is about experimented results and discussion followed by conclusions and future scope in Section 5.

2. Review of Wu & Tsai's Method and Chan's Method

2.1 Wu & Tsai's Method

In Wu & Tsai's method [9], the entire range of the pixel is segregated in 5 sub-ranges. The range division is as follows (0.. 7), (8.. 15), (16.. 31), (32.. 63), (64.. 255). For each subrange lower value is denoted by *Li* and the upper limit is denoted by *Ui*. *D* defines the gap between any two consecutive pixels. It is calculated by taking the difference gap between any two pixels in absolute form. The width of the range is defined by *Wi* and always is a power of 2. Value of *Wi* is to be calculated using Eq. (1).

$$Wi = (Ui - Li + 1) \tag{1}$$

The restriction in width range is applied to implement a smoother embedding process. If the value of *D* lies in the area where the difference gap between pixels is less (smoother area), comparatively fewer data could be hidden in that region. Oppositely, if *D* lies within the area where the difference gap is larger (sharper area), the particular block is tolerable to adopt changes due to alteration in bits. Following this phenomenon, hiding capacity (*Ti*) is calculated using Eq. (2).

$$Ti = \lfloor \log Wi \rfloor \tag{2}$$

Ti denotes the number of bits to be taken from the secret bitstream and to be embedded for hiding purpose within that area. The decimal number corresponding to those bits is denoted by *Ti´*. New difference between adjacent pixels after embedding the secret message is to be calculated using Eq. (3).

$$D' = Li + Ti \tag{3}$$

Bits from secret data stream could be embedded by altering *P* and *Q*. New state of *P* and *Q* after the alteration is to be denoted by *P´* and *Q´*. The adjustment is explained in Eq. (4).

$$P',Q' = \begin{Bmatrix} \left(P+\left\lceil\frac{m}{2}\right\rceil, Q-\left\lfloor\frac{m}{2}\right\rfloor\right) \text{ if P} \geq \text{Q and D'>D} \\ \left(P-\left\lfloor\frac{m}{2}\right\rfloor, Q+\left\lceil\frac{m}{2}\right\rceil\right) \text{ if P<Q and D'>D} \\ \left(P-\left\lceil\frac{m}{2}\right\rceil, Q+\left\lfloor\frac{m}{2}\right\rfloor\right) \text{ if P} \geq \text{Q and D'} \leq \text{D} \\ \left(P+\left\lceil\frac{m}{2}\right\rceil, Q-\left\lfloor\frac{m}{2}\right\rfloor\right) \text{ if P < Q and D'} \leq \text{D} \end{Bmatrix} \quad (4)$$

where $m = |D' - D|$.

2.2 Chan's Method

In Chan's proposal [12] the authors modified the rigidity in the selection of block size. At first, blocks in a non-overlapping manner are selected from the cover image. Here, the size is *M* by *N*.

The range of *M* and *N* is flexible and varies from 1 to 7. In each block, the pixel is located at the central position, surrounded by eight neighbors, and is engaged for embedding using the substitution process. The number of bits of the pixel to be covered is denoted by *k*. As mentioned in this approach the value of *k* is 3. Here, *pic* denotes the pixel at the center position of a block and *p'ic* denotes the same pixel when 3 rightmost bits are substituted by replacement/adjustment. *p''ic* points out the state of the same pixel after applying the error reduction process, popularly known as OPAP [132]. The embedding error due to modification is calculated as $\delta i = p'ic\text{-}pic$. Three different adjustments are suggested to calculate *p''ic*. They are formulated in Eq. (5).

$$\begin{aligned} &Case1\text{: } (2^{k-1} < \delta < 2^k)\text{: } if\ pic \geq 2^k \\ &Then\ p''ic = p'ic - 2^k\ \ Otherwise\ p''ic = p'ic; \\ &Case2 : (-2^{k-1} \leq \delta i \leq 2^{k-1}) : if\ p''ic = p'ic; \\ &Case3 : (-2^k < \delta i < 2^{k-1}) : if\ pic < 256 - 2^k, then \\ &p''ic = p'ic + 2^k; Otherwise\ p''ic = p'ic; \end{aligned} \quad (5)$$

In the next phase of the embedding procedure, the surrounding pixels are numbered sequentially. Consequently, the difference gap is calculated between the pixel at middle and other surrounding pixels. The process continues for a block thereafter, repeated for all blocks. *Pik* denotes surrounding pixels in a block. After embedding the secret message, modified state of the pixel is denoted by *P'ik*. Calculation of *P'ik* from *Pik*, after modifying the pixel is done using Eq. (6). $\delta'ik$ denotes the difference gap in pixel intensity observed before embedding message and after the embedding is complete. Here *k*=1,2….N.

$$p'ik = \begin{Bmatrix} p''ik\ if\ |pik - p''ik| <= |pik - p'''ik|\ and\ 0 \leq p''ik \leq 255 \\ p''ik\ else\ if\ p'''ik > 255 \\ p'''ik\ otherwise \end{Bmatrix} \quad (6)$$

$$Where \begin{Bmatrix} p''ik = p'ic - \delta'ik \\ p'''ik = p'ic + \delta'ik \quad k = 1,2,.....M*N \end{Bmatrix}$$

While extracting the secret message, the steganographic image containing secret data is partitioned into *M*×*N* blocks. For each block, 3-bits secret message is extracted from the central pixel. Thereafter, the block wise gap between the pixel at the center and pixels surrounding it is calculated. The difference in

the gap is checked and mapped against the same range divisions. The final secret message is obtained by calculating the lower value and the difference value.

3. The Proposed Approach

The entire method is divided into three phases: the phase of distribution of ranges, the embedding process, and the phase of extraction.

In the first phase, pixels are categorized in 6 different ranges (0..7), (8..15), (16..31), (32..63), (64..123), and (124..255). For each range, there is an upper limit and lower limit. The first digit within brackets indicates the lower limit and the second digit indicates the upper limit. The mathematical procedure behind calculating the number of bits to be embedded in each range remains the same like Chan's method using equation 1 and 2, i.e., the method proposed by Wu and Tsai [9] is applied here.

In the embedding phase, the cover image is segmented into $M \times N$ blocks where the value of M and N is 3×3, 2×2 or 1×2. Now, within the central pixel of a block, 4-bit secret data is embedded by LSB and OPAP using equation 5, having k=4 (Fig. 1).

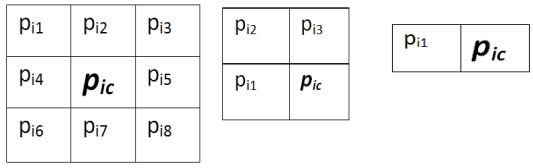

Fig. 1. M×N blocks: 3×3, 2×2, and 1×2. *Pic* indicates the central pixel.

Thereafter, differences between the central pixel and other pixels are calculated. If the difference value is between 0 and 31, embed secret data within the gap between *Pic* and *Pik*. After embedding secret data *Pik* is adjusted to $P\hat{i}k$ according to following cases.

$$Case1: p'ik = pik - |d' - d| \; if \; d' > d \text{ and } pik \le p"ic$$
$$Case2: p'ik = pik + |d' - d| \; if \; d' > d \text{ and } pik > p"ic$$
$$Case3: p'ik = pik + |d' - d| \; if \; d' < d \text{ and } pik < p"ic$$
$$Case4: p'ik = pik - |d' - d| \; if \; d' < d \text{ and } pik > p"ic$$
$$Where \; d = |pik - pic| \; and \; d' = |pik - p"ic|$$

Case1 to Case 4 follows a different approach for adjusting pixels according to the pixel values in the original image. If the gap is greater than 31, 4-bit secret data is embedded by LSB and OPAP using equation 3 having a value of k=4. To avoid overflowing of data $P\hat{i}k$ is calculated once more. If the value of $P\hat{i}k$ is <0 or the same is greater than 255, the calculation is as follows: if $P\hat{i}k$ <0 then $Pik = Pik$ + absolute ($P\hat{i}k$). Otherwise, if $P\hat{i}k$ >255 then $Pik = Pik - (P\hat{i}k - 255)$ and the embedding process is repeated considering the new value of *pik*.

In the extraction phase first, the image with a secret message is segmented into equal size blocks as it was done at the time of embedding. For each block initially, four bits from secret message are extracted, then the difference between $P\acute{i}c$ and *Pik* is checked. The range of the difference value is checked from the same range division and finally, the secret message is obtained by calculating the lower value and the difference value. Otherwise, four bits of data directly extracted from *Pik*.

4. Experimented Results

After successfully implementing the proposal over sample images, experimented results have been tabulated in this section. Here, in the proposal, three types of blocks have been used for the experiment. Sizes of those blocks are 3×3 block, 2×2 block, and 1×2 block. Results obtained from each of the blocks are compared with the results obtained from equivalent blocks of Chan's method [12] and tabulated in Tables 1 and 2. The proposal has been implemented in an environment configured with Intel Core 2 Duo processor supported by a primary memory of 2 GB. MATLAB programming language has been used here to implement the algorithm. The said experiment has been applied to some standard, commonly used grayscale images. Statistical distortion is obvious in such applications. To calculate the percentage of change between the original image and embedded copy, PSNR is used. Prior to calculating PSNR, calculation of mean square error (MSE) is required. The process of calculating MSE and PSNR has been discussed in the second half of this section.

Table 1. Capacity or payload (in bits) and the PSNR (in dB) experimented using the proposal and Chan's method for 3×3 and 2×2 block

Image	3×3 block				2×2 block			
	Chan's method [12]		Proposed method		Chan's method [12]		Proposed method	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	811342	34.74	837140	34.20	810192	35.61	868587	37.55
Baboon	934574	27.03	936091	27.71	944226	28.72	951656	35.48
Peppers	813628	31.27	843170	32.92	809896	32.21	874003	37.7
Tiffany	812573	33.77	843089	34.58	803649	33.75	873270	37.04
Aerial	873936	28.46	856230	32.56	863742	29.58	875516	37.44
Goldhill	834264	34.22	835133	33.91	817359	35.16	880120	37.34
Boat	842506	32.19	855442	32.62	829630	33.37	883362	37.13
Brian	821005	34.63	832975	35.67	810386	35.18	863125	37.71
Avg.	842978	32.03	854908	33.02	836135	32.94	883705	37.17

Table 2. Capacity (in bits) and the PSNR (in dB) of 1×2 block of proposed method 7×7 block of Chan's method,1×3 block using Khodaei and Faez's method

	Proposed method		Chan's method [12]		Khodaei and Faez's method [11] (k=3)	
	1×2 block		7×7 block		1×3 block	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	930024	36.16	849736	30.15	805823	36.81
Baboon	1010681	34.02	1010310	25.71	876021	34.71
Peppers	932809	36.36	850975	27.48	804181	35.42
Tiffany	933147	35.61	824638	31.90	800670	36.23
Aerial	928884	36.22	957550	24.81	839429	34.76
Goldhill	933393	36.12	877493	31.43	810213	36.64
Boat	949594	35.87	886427	29.14	820490	33.80
Brian	924244	36.32	848365	30.21	806028	36.14
Avg.	942847	35.58	888186	28.85	820357	35.55

The procedure of measuring the error is explained here; x and y indicate the arrays of size $M \times N$. Respectively they also represent the reference frame indicating the original image and the frame representing the same image after embedding messages. Calculation of MSE between the two images is observed in Eq. (7). Thereafter, PSNR is calculated following Eq. (8).

$$MSE = \frac{1}{M * N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i,j) - Y(i,j)]^2 \tag{7}$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \tag{8}$$

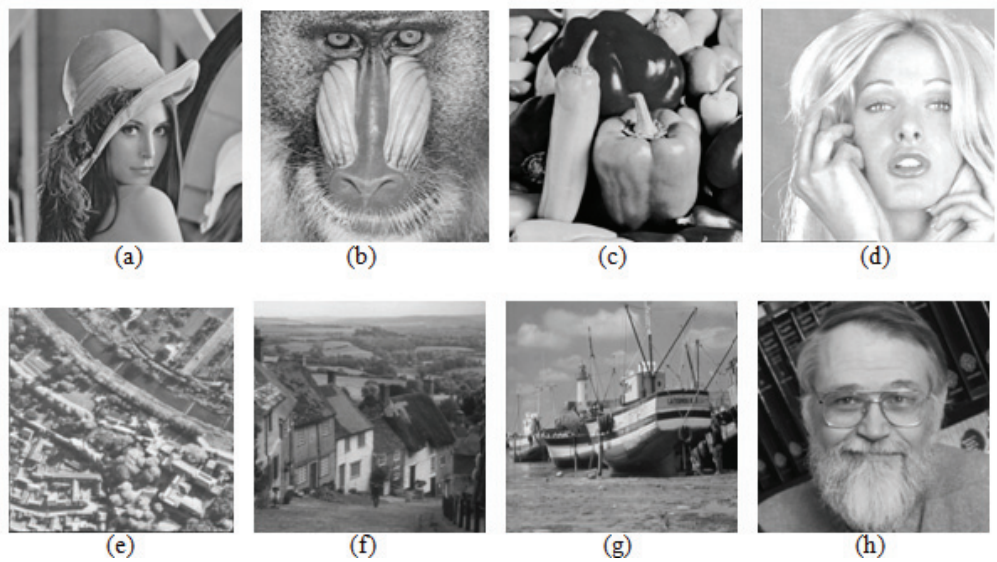

Fig. 2. Original images: (a) Lena, (b) Baboon, (c) Peppers, (d) Tiffany, (e) Aerial, (f) Goldhill, (g) Boat, and (h) Brian.

The range of intensity values for any pixel is represented by L. If, Y is of 8-bit depth, then $L = 2^8 - 1 = 255$. While representing the results decibel (dB) is used as a measurement unit. Table 1 shows PSNR and the capacity of our proposed method and Chan's method for different block size. Images Lena, Baboon, Peppers, Tiffany, Aerial, Goldhill, Boat and Brian are standard 512×512 grayscale images. Secret data embedded in each image are random bits (0/1), generated using a random number generator function. From Table 1, it is observed that the 3×3 block from Chan's method [12] and our proposed method, produces almost the same PSNR for the almost same amount of secret data. For 2×2 blocks, our proposed method produces much better PSNR with better hiding capacity when compared with the Chan's method [12]. In the proposed method the 1×2 block achieves the highest capacity rather than the other two blocks experimented by our algorithm. The 7×7 block of the Chan's method [12] has the highest capacity. In Table 2, we compare the 1×2 block of our method with the 7×7 block of the Chan's method [12] as the 7×7 blocks can hide maximum data according to their algorithm. In this situation, our proposed method gets much better PSNR with an average of 35.58 dB and can store 942847 bits of the secret message on an average, compared to Chan's method with a poor PSNR of 28.85 dB having a

capacity of 888186 bits. Table 2 also shows a comparison of capacity and PSNR which includes the 1×3 block using Khodaei and Faez's method [11] and LSB-3 method. The results of the proposal show that the capacity is higher and PSNR is almost the same when compared with Khodaei and Faez's work [11]. Fig. 2 shows original images used in the approach, and Fig. 3 shows images after embedding.

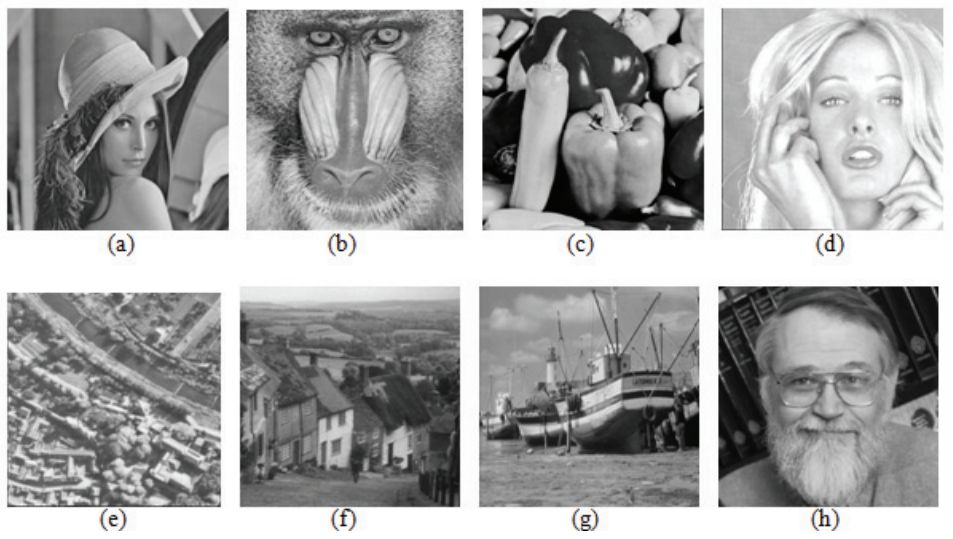

Fig. 3. Images after embedding secret data: (a) Lena and (b) Baboon using 3×3 block; (c) Pepers and (d) Tiffany using 2×2 block; (e) Aerial, (f) Goldhill, (g) Boat, and (h) Brian using 1×3 block.

4.1 Performance Analysis

The block-based adaptive concept proposed by Chan et al. [12] added flexibility on Khodaei and Faez's algorithm [11] but, in their work, it has been observed that for a small size block (2×2, 3×3) capacity is low and PSNR is high. For a large size block (7×7) capacity has been increased, but as a consequence, PSNR degrades in a significant amount. So, a proper ratio between the capacity and the image quality is not available. Our proposed method overcomes that limitation by increasing data hiding capacity without degrading PSNR. To reflect this, we have made changes in the size of blocks and modified the encoding algorithm according to their capacity. In our approach, the block size is ranged from 1 to 3 because it is observed from Chan's result that, for blocks where a value of M and N is more than 3, PSNR and capacity are degraded. The range division phase is divided into two parts. To increase capacity, ranges having a value less than 32 is embedded by PVD method and ranges having value more than 32, use LSB along with OPAP. A flat application of anyone may not properly utilize the gaps between pixels; rather this hybrid approach increases the capacity along with PSNR. Table 1shows that, the proposed algorithm succeeds to achieve better capacity using small size blocks (1×2 of the proposed method) with an increase PSNR of 6.73 dB when compared to large size blocks (7×7) of the Chan's method [12]. Each of our blocks also has better capacity and PSNR when compared with the same size block of the Chan's method. Table 2 shows that the 1×2 block of our proposed method has better PSNR and capacity ratio than the 7×7 block of the Chan's method. It is also better than the 1×3 block using Khodaei and Faez's method [11]. Results experimented using other blocks applying the Chan's method [12] are not mentioned here

because those results are quite same with already mentioned blocks in Table 2. We didn't mention any comparison for the 1×1 block of the Chan's method [12] in this paper, because the application in this block is a simple LSB implementation of 3 bits using 3-LSB and OPAP. At the same time, there is no application of PVD. Our algorithm also ensures that all the pixels are within the boundary region after embedding secret data.

5. Conclusions

The proposal in this paper is an adaptive data hiding technique which combines LSB substitution, OPAP and PVD. Our original work is motivated by Khodaei and Faez's work [11]. The important point about this algorithm is that, this approach is flexible towards choosing a block of variable size while embedding. Selection is dependent on capacity and image quality, whereas Khodaei and Faez's work were rigid towards the size of the block. Our proposed method also succeeds to maintain a good ratio between capacity and image quality for all type of blocks by applying changes in embedding procedure and block size. Selected blocks achieve better capacity and much better PSNR when compared against Chan's method [12]. Boundary regions are manipulated so that none of the values fall out of range. In the future, our scope would be to embed more bits by engaging the same amount of pixels resuming image quality of the proposed algorithm. This application may be applied to secure e-Governance related attachments against unauthorized attack and could raise high the acceptability of this novel application.

References

[1] T. Sharp, "An implementation of key-based digital signal steganography," in *Information Hiding*. Heidelberg: Springer, 2001, pp. 13-26.

[2] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures*. New York, NY: Springer, 2001.

[3] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469-474, 2004.

[4] K. Bailey and K. Curran, "An evaluation of image based steganography methods," *International Journal of Digital Evidence*, vol. 2, no. 2, pp. 1-40, 2003.

[5] C. C. Chang, M. H. Lin, and Y. C. Hu, "A fast and secure image hiding scheme based on LSB substitution," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 4, pp. 399-416, 2002.

[6] C. C. Thien and J. C. Lin, "A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function," *Pattern Recognition*, vol. 36, no. 12, pp. 2875-2881, 2003.

[7] T. Halder and S. Karforma, "A LSB-indexed steganographic approach to secure e-governance data," in *Proceedings of the 2nd International Conference on Computing and System (ICCS)*, Burdwan, India, 2013, p. 158.

[8] T. Halder and S. Karforma, "A novel e-governance data hiding approach combining LSB-steganography and cryptography," *Indian Science Cruiser*, vol. 28, no. 4, pp. 44-49, 2014.

[9] D. C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1613-1626, 2003.

[10] K. C. Chang, C. P. Chang, P. S. Huang, and T. M. Tu, "A novel image steganographic method using tri-way pixel-value differencing," *Journal of Multimedia*, vol. 3, no. 2, pp. 37-44, 2008.

[11] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Processing*, vol. 6, no. 6, pp. 677-686, 2012.
[12] Y. Y. Tsai, J. T. Chen, and C. S. Chan, "Exploring LSB substitution and pixel-value differencing for block-based adaptive data hiding," *International Journal of Network Security*, vol. 16, no. 5, pp. 363-368, 2014.
[13] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469-474, 2004.

Tanmoy Halder https://orcid.org/0000-0001-6216-683X

He has received MCA (master in computer application) from The University of Burdwan in 2007. Now he is working as Assistant Professor in Dr.B.C.Roy Engineering College, Durgapur, West Bengal, India. His interested area of research is steganography and network security. He has published five research papers in international journals and conferences.

Sunil Karforma https://orcid.org/0000-0001-8371-4743

He has completed B.E. and M.E. degrees in computer science and engineering from Jadavpur University. He has completed Ph.D. in the field of cryptography. He is presently holding the post of Professor and the Head of the Department in the Department of Computer Science, The University of Burdwan. Network security and e-Commerce is his field of interest in the research area. He has published approximately 16 research papers in reputed national and international journals and proceedings.

Rupali Mandal https://orcid.org/0000-0001-7180-5795

She has completed MCA (master in computer application) from The Bangalore University in 2008. Now working as Assistant Professor in Bengal College of Engineering and Technology, Durgapur, West Bengal, India. She has published two research papers on steganography and network security.