Journal of Theoretical and Applied Information Technology
20th February 2015. Vol.72 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECRET DATA HIDING BY OPTIMIZING GENERAL SMOOTHNESS DIFFERENCE EXPANSION-BASED METHOD

[1]MUHAMMAD HOLIL, [2]TOHARI AHMAD
[1,2] Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia
E-mail: [1]holil12@mhs.if.its.ac.id , [2]tohari@if.its.ac.id

ABSTRACT

Computer network has made it easy to exchange data; however, there is a security challenge raising from it. This is because the transmitting data can be intercepted by an attacker. Steganography can be used to prevent an attacker from exploring the information in the intercepted data. There are three common problems in the steganography, those are: obtaining both secret data and cover data from stego data, capacity of the secret data and the quality of the stego data. In this paper we propose a method which works on those problems based on the smoothness difference expansion. The experimental result shows that the proposed method is able to extract both secret and cover data, to increase both capacity and quality of the stego data.

Keywords: *Difference Expansion, Block Smoothness, Steganography, Data Hiding, Data Security*

1. INTRODUCTION

For the last decades, medical image technology has been growing rapidly. In order to make it easy for medical staff to diagnose a patient, the medical data – for example, identity of patient, medical image and medical record – can be transmitted from one location to another. This ease of transmission may make a patient can be treated fast and accurately, which leads to save the patient life.

Nevertheless, transmitting data over a network is vulnerable to attack. It is possible that an attacker intercepts the data and exploits it for later uses. On the other hand, the data transmission must be done securely. In particular, confidentiality and integrity of the data should be maintained. In order to overcome this security problem, some methods have been introduced, for example steganography. Different from cryptography [13], steganography produces readable data. In the case of images, the stego image may have similar form to the cover image.

In steganography, the secret is embedded to the cover data, such as an image. In the medical environment, the medical data (the secret) is embedded to the respective medical image of the patient (cover data). So, in case an attacker intercepts the stego data, he/she just finds this medical image without knowing who this image belongs to, or any other related information.

Due to its characteristics, any distortion in medical images is not acceptable. Small noises occurred in the medical image may affect the accuracy of the diagnose. Therefore, the extraction process must be able to produce both the original secret and cover data. It means that the process must be reversible [1]. There are two other problems raising from common methods. First, the capacity of the secret can be embedded to the cover data. This problem determines how much data can be hidden in the cover, where larger capacity is better. Second, the similarity between the stego and the cover data, where higher similarity level means better, too.

There have been some research work on those problems. The performance, however, still a challenge. In this paper, we propose a method based on the Quad Smoothness [2] and Generalized Different Expansion (GDE) [3] along with refinement of [4] [5] to increase the capacity and the similarity level while still make the process reversible.

The rest of the paper is structured as follows. Section 2 describes the research relates to the proposed method. Section 3 presents the proposed method whose result is provided in section 4. Finally, the conclusion is drawn in section 5.

2. RELATED WORKS

Journal of Theoretical and Applied Information Technology
20th February 2015. Vol.72 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The DE method [6] has some advantages. Mainly, it is able to provide a large embedding capacity with low complexity [7]. This method works by inserting bits of data on the difference between pairs of pixels while retaining its average value, as depicted in (1), where u_1 and u_2 are adjacent pixels in a 8-bit grayscale image, v and m are the pixel difference and the average of a pair of pixels, respectively.

$$m = \left\lfloor \frac{u_1 + u_2}{2} \right\rfloor, v = u_1 - u_2 \quad (1)$$

The embedding process is performed by expanding the difference pixel v. For secret data $b \in \{0,1\}$, the expansion is done by using (2). Difference pixels $\tilde{v}$ that contains the secret data is used to generate new pixels u'_1 and u'_2 as in (3).

$$\tilde{v} = 2 \text{ x } v + b \quad (2)$$

$$u'_1 = m + \left\lfloor \frac{\tilde{v}+1}{2} \right\rfloor, u'_2 = m - \left\lfloor \frac{\tilde{v}}{2} \right\rfloor \quad (3)$$

Practically, new pixels u'_1 and u'_2 must not be overflow (greater than 255), or underflow (smaller than 0). This is because, those overflow and underflow values result in unrecoverable data. This means that the original data can lose. In order to avoid this condition, the difference pixels $\tilde{v}$ has to fulfill (4).

$$\begin{cases} |\tilde{v}| \leq 2 \text{ x } (255 - m), \text{if } 128 \leq m \leq 255 \\ |\tilde{v}| \leq 2 \text{ x } m + 1, \quad \text{if } 0 \leq m \leq 127 \end{cases} \quad (4)$$

In [8], Alattar proposes triplet DE-based method. It is able to increase the number of bits can be hidden in the cover by replacing the number of pixels in a block, from two (pair) to three (triplet). In further development, he improves the capacity of the previous method by replacing the number of pixel in the block into quad [9] and vector [3].

Furthermore, the size of the block pixels can be more dynamic [3]. This is determined based on the set of neighboring pixels whose size is axb as in Figure 1. Those pixels are employed to construt a vector $u = u_1, u_2, \ldots, u_{N-1}$, where N is the number of pixels in a block.

In [3], the average of the first block of pixels is defined in (5). This value is used for determining $v_1, v_2, \ldots, v_{N-1}$ (see 6) where the embedding process is applied.

$$v_0 = \left\lfloor \frac{\sum_{i=0}^{N-1} u_i}{N} \right\rfloor \quad (5)$$

$$v_{N-1} = u_{N-1} - u_0 \quad (6)$$

Figure 1: A Block of Pixels (*adapted from* [3])

The embedding process itself can be performed in two ways: difference expansion (7) or LSB substitution (8), depending on the requirements specified in (9).

$$\left.\begin{aligned} \tilde{v}_1 &= 2 \text{ x } \left\lfloor \frac{v_1}{2} \right\rfloor + b_1 \\ &\ldots \\ &\ldots \\ \tilde{v}_{N-1} &= 2 \text{ x } \left\lfloor \frac{v_{N-1}}{2} \right\rfloor + b_{N-1} \end{aligned}\right\} (7)$$

$$\left.\begin{aligned} \tilde{v}_1 &= 2 \text{ x } v_1 + b_1 \\ &\ldots \\ &\ldots \\ \tilde{v}_{N-1} &= 2 \text{ x } v_{N-1} + b_{N-1} \end{aligned}\right\} (8)$$

$$\left.\begin{aligned} 0 \;&\leq\; \mathrm{v}_0 - \left\lfloor \frac{\sum_{i=0}^{N-1} \tilde{v}_i}{\mathrm{N}} \right\rfloor \leq 255 \\ 0 \;&\leq\; \tilde{v}_1 + \mathrm{v}_0 - \left\lfloor \frac{\sum_{i=0}^{N-1} \tilde{v}_i}{\mathrm{N}} \right\rfloor \leq 255 \\ &\ldots \\ &\ldots \\ 0 \;&\leq\; \tilde{v}_{N-1} + \mathrm{v}_0 - \left\lfloor \frac{\sum_{i=0}^{N-1} \tilde{v}_i}{\mathrm{N}} \right\rfloor \leq 255 \end{aligned}\right\} (9)$$

At the end of embedding process, the values of $\tilde{v}_1$ until $\tilde{v}_{N-1}$ are obtained. The new pixels are calculated using (10) which is the inverse of (5) and (6).

$$\left.\begin{aligned} u'_0 &= v_0 + \left\lfloor \frac{\sum_{i=0}^{N-1} \tilde{v}_i}{N} \right\rfloor \\ u'_1 &= \tilde{v}_1 + u_0 \\ &\ldots \\ u'_{N-1} &= \tilde{v}_{N-1} + u_0 \end{aligned}\right\} (10)$$

Journal of Theoretical and Applied Information Technology
20th February 2015. Vol.72 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

According to the capability of blocks in carrying an image, Hsiao et al [10] categorize blocks of image into 3 groups: complex, normal and smooth. Complex blocks are not embedded in order to maintain the quality of the image. Smooth blocks are able to collect two times bigger than that of normal.

In contrast to the previous methods, Lou et al [11] propose Reduced Difference Expansion (RDE). Here, the difference in the original DE method [6] is reduced. As a result, the stego image has higher similarity to the respective cover image. It means that the quality of the stego image is better than before. Other research relates to the stego image quality is performed by Lin et al [2]. In this method, the embedding process is similar to the Quad DE [9]; however, the order of the embedding process is determined based on the smoothness level of each block. The calculation of smoothness level itself is depicted in (10), where avg_i is the average of block i^{th}, and $\overline{avg}$ is the average of them (see (11)). The embedding process starts from the smallest to the largest va. This method is capable of providing better quality of images, especially in the small data embedding.

$$va = \frac{\sum_{i=0}^{3}(avg_i - \overline{avg})^2}{4} \quad (10)$$

$$\overline{avg} = \frac{(avg_1 + avg_2 + avg_3 + avg_4)}{4} \quad (11)$$

3. PROPOSED METHOD

As in other data hiding methods, this proposed algorithm consists of two parts, data embedding and data extraction. This is developed based on the Quad Smoothness [2] and Generalized Different Expansion (GDE) [3] and the refinement of our previous research [4] [5].

3.1 Data embedding

Like [2], we calculate the smoothness level before the embedding process begins. In this proposed method, we enhance the block size to quad of vector instead of quad of quad. In addition, the calculation of the smoothness level is taken from the first pixel u_0 of each block instead of the average pixel v_0. This smoothness level is calculated based on the variance of the four blocks. In this proposed method, we use the median value instead of v_0 as in [2]. This is because the value of median does not change for both before and after the embedding process.

Suppose U_{m0}, U_{m1}, U_{m2}, and U_{m3} are the first pixel of block A, B, C and D, respectively. The smoothness level is calculated using (12) and (13).

$$va = \frac{\sum_{i=0}^{3}(U_{mi} - \overline{avg})^2}{4} \quad (12)$$

$$\overline{avg} = \frac{\sum_{i=0}^{3} U_{mi}}{4} \quad (13)$$

According to the quad smoothness [2], the embedding process is performed according to the smoothness level of each block, where the block with the smallest va is embedded firstly. The data embedding for the concerned block is performed like in generalized DE [3], where the process begins with calculating the difference of pixels in the respective block. Different from that, v_0 in (5) is removed to make it reversible. Furthermore, we use the median pixel of each block as the base point. This is intended to have a smaller difference value, so that the capacity and the quality of the data is better. For blocks whose pixel number is odd, their median is to be the u_m; and for that with even, u_m is the closest pixel whose value is smaller than the median itself. In this proposed method, the difference of pixels is specified in (14).

$$\begin{cases} v_1 = u_1 - u_m \\ \ldots \\ v_{N-1} = u_{N-1} - u_m \end{cases} \quad (14)$$

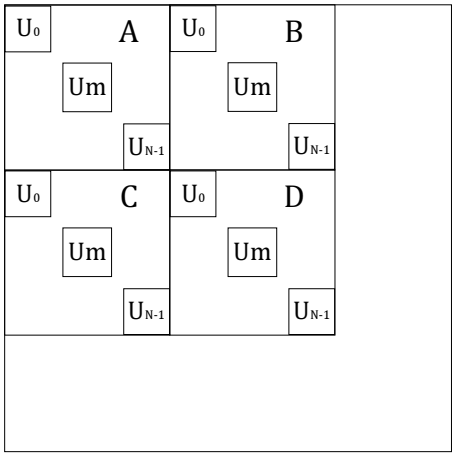

Figure 2 A quad block in the proposed method

Before the data is embedded, the difference of pixels is reduced using RDE [11] as in our previous research [4]. Difference pixels reduction is performed according to (15). Unlike the RDE, however, the difference pixels in this proposed method can be positive or negative. It is intended to make the sequence of pixels in each block unchanged. Furthermore, the expandable block is divided into two subcategories according to the result of (15), that is expandable RDE (if $\overline{v_i}$ is not equal to v_i) and expandable non-RDE (if $\overline{v_i}$ is equal to v_i).

Journal of Theoretical and Applied Information Technology
20th February 2015. Vol.72 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\bar{v} = \begin{cases} v, & \text{if } -2 < v < 2 \\ v + 2^{\lfloor \log_2 |v| \rfloor - 1}, & \text{if } -2 \geq v \\ v - 2^{\lfloor \log_2 v \rfloor - 1}, & \text{if } v \geq 2 \end{cases} \quad (15)$$

Difference pixels $\bar{v}$ that have been reduced will be embedded with the secret data. It is done as in generalized DE [3]. In order to improve the message capacity, we embed two bits data into expandable blocks (expandable RDE and expandable non-RDE) whose variance is less than or equatl to 1; and one bit for that with variance is more than 1. For this purpose, we use (16), where $b_1, b_2, \ldots, b_{N-1} \in \{0,1,2,3\}$ and (8), for those first and second cases, respectively.

$$\begin{cases} \tilde{v}_1 = 4 \text{ x } v_1 + b_1 \\ \ldots \\ \tilde{v}_{N-1} = 4 \text{ x } v_{N-1} + b_{N-1} \end{cases} \quad (16)$$

Once the embedding process is complete, the new pixels can be obtained. In this proposed method, the new pixel is constructed by using (20) which must satisfy the requirements in (21) to prevent those values from overflow or underflow.

$$\begin{cases} u'_0 = u_0 \\ u'_1 = \tilde{v}_1 + u_0 \\ \ldots \\ u'_{N-1} = \tilde{v}_{N-1} + u_0 \end{cases} \quad (20)$$

$$\begin{cases} 0 \leq u'_1 \leq 255 \\ \ldots \\ 0 \leq u'_{N-1} \leq 255 \end{cases} \quad (21)$$

As described before, we divide blocks into four categories: expandable RDE, expandable non-RDE, changeable, and unchangeable whose identity is according to the value in the location map (*LM*). It has been defined in our previous research [5] that the first two bits of *LM* are 11, 10, 01 and 00 for expandable RDE, expandable non-RDE, changeable and unchangable, respectively. The size of the *LM* is $p(l+2)$, where p and l are the number of blocks and the number of difference of pixels that contains data in each block.

For expandable RDE, the value of rest bit in *LM* is determined according to (22). For changeable block, the value of the remaining bits is determined according to LSB of each difference of pixels. Expandable non-RDE and unchangeable blocks do not need more additional bits, so, the value of rest bits is 0. As in our previous method [4], LM will not be embedded to stego image.

$$LM = \begin{cases} 0, \text{if} \bar{v}_i \pm 2^{\log_2 |\bar{v}_i| - 1} = v_i \\ 1, \text{if } \bar{v}_i \pm 2^{\log_2 |\bar{v}_i| - 1} \neq v_i \end{cases} \quad (22)$$

3.2 Extraction and recovery process

The data extraction process begins from the smallest to the highest va. Then $\tilde{v}_i$ is calculated according to (12). The expandable blocks (expandable RDE and expandable non-RDE blocks) are extracted according to the order of the smoothness level. Those are, two bits LSB for $v_a \leq 1$, and one bit LSB for $v_a > 1$ as depicted in (23).

$$b_i = \begin{cases} \tilde{v}_i - \left(4 \text{ x } \left\lfloor \frac{\tilde{v}_i}{4} \right\rfloor\right), \text{if } va \leq 1 \\ \tilde{v}_i - \left(2 \text{ x } \left\lfloor \frac{\tilde{v}_i}{2} \right\rfloor\right), \text{if } va > 1 \end{cases} \quad (23)$$

The recovery of the pixel difference depends on the category of each block, whether is expandable RDE, expandable non-RDE or changeable. That is, for both expandable RDE and expandable non-RDE, it is done by performing right shift (one bit if $v_a \leq 1$, and two bits if $v_a > 1$).

For expandable RDE blocks, a further step should be done to recover the reduced pixel value. It is applied to the third bit onwards of the location map LM_j, where $j \in \{3,4,..,l+2\}$, and l is the number of pixels in the resepective blocks [5]; or it can also be determined that $j = i + 2$. For this expandable RDE, $\bar{v}_i$ that is reduced in the embedding process is recovered to v_i using (24) for $\bar{v}_i < 0$, or using (25) which has been specified in [11] for $\bar{v}_i > 0$. As for changeable blocks the recovery process is performed as in the [3].

$$v_i = \begin{cases} \bar{v}_i - 2^{\log_2 |\bar{v}_i| - 1}, & \text{if } LM = 0 \\ \bar{v}_i - 2^{\log_2 |\bar{v}_i|}, & \text{if } LM = 1 \end{cases} \quad (24)$$

$$v_i = \begin{cases} \bar{v}_i + 2^{\log_2 |\bar{v}_i|}, & \text{if } LM = 1 \\ \bar{v}_i + 2^{\log_2 |\bar{v}_i| - 1}, & \text{if } LM = 0 \end{cases} \quad (25)$$

4. EXPERIMENTAL RESULT

The evaluation is done by using 5 medical images in idimages [12]. Furthermore, the generalized DE [3] is also implemented whose result is compared to this proposed method. Similar to [5], the experiment is carried out by embeding random data to the cover images which have divided into blocks of size: 2x2, 3x3, 4x4, 5x5, and 6x6. The evaluation is applied for measuring the PSNR of stego data, by embedding 25 kb, 50 kb and 100 kb for all those images and block sizes. In addition, the maximum

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

capacity and its corresponding PSNR are also measured.

Let P0 represents the generalized DE; P1, P2, P3 and P4 are our previous methods provided in [5]; and P5 is the proposed method in this paper. It is found that P5, in general, has higher PSNR values for relatively high capacity whose block size is 6x6. In contrast, P4 is superior to other methods for less capacity of the secret data. In this case, P5 is the second highest. In more details, these experimental results are depicted in Tables 1, 2, and 3. Therefore, P5 is more appropriate to use for relatively high capacity secret data with a large block size. Moreover, the PSNR of P5 is significantly higher than that of P0 for all block sizes and for all specified capacities.

It is also shown that the best PSNR is obtained with block size of 2x2 and the worst is with 6x6. In more details, the number of the PSNR slightly goes down inline with the rising of the size of the blocks. This is because, increasing the size of the block results in increasing the number of unchangeable blocks caused by the rise of the difference value between the median pixel and other pixels in the respective block.

Furthermore, increasing the capacity or block sizes degrades the stego image quality (decreasing the PSNR value). However, in general, the proposed method (P5) is still able to maintain this image quality, where the decrease is less significant than that of P4.

In terms of the maximum secret data can be embedded into the cover data, the result is varied. Even though its maximum capacity is lower than that of certain methods, P5 is able to obtain the highest PSNR for various block sizes, as shown in Table 4.

5. CONCLUSIONS

In this paper, we propose a method to increase the capacity of the secret data and the similarity between cover and stego data while both the original secret data and the cover data can be obtained without any distortion. The proposed method, which is developed based on smoothness DE, has been able to obtain better results than existing methods, particularly for relatively higher secret data size with a large block size.

REFERENCES

[1] J.-B. Feng, I.-C. Lin, C.-S. Tsai and Y.-P. Chu, "Reversible Watermarking: Current Status and Key Issues," *International Journal of Network Security,* pp. 161-171, 2006.

[2] C.-N. Lin, D. J. Buehrer, C.-C. Chang and T.-C. Lu, "Using quad smoothness to efficiently control capacity–distortion of reversible data hiding," *The Journal of Systems and Software,* vol. 8, no. 3, p. 1805–1812, 2010.

[3] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing,* vol. 13, no. 8, pp. 1147-1156, 2004.

[4] T. Ahmad, M. Holil, W. Wibisono and R. M. Ijtihadi, "An Improved Quad and RDE-based Medical Data Hiding Method," in *The IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom)*, 2013.

[5] T. Ahmad and M. Holil, "Increasing the Performance of Difference Expansion-based Steganography when Securing Medical Data," *Smart Computing Review,* vol. 4, no. 4, pp. 307-322, 2014.

[6] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transaction on Circuits and Systems for Video Technology,* vol. 13, no. 8, pp. 890-896, 2003.

[7] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing,* vol. 16, no. 3, pp. 721-730, 2007.

[8] A. M. Alattar, "Reversible watermark using difference expansion of triplets," in *International Conference on Image Processing*, 2003.

[9] A. M. Alattar, "Reversible watermark using difference expansion of quads," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[10] J.-Y. Hsiao, K.-F. Chan and J. M. Chang, "Block-based reversible data embedding," in *Signal Processing 89* , 2009.

[11] D.-C. Lou, M.-C. Hu and J.-L. Liu, "Multiple layer data hiding scheme for medical images," *Computer Standards & Interfaces,* vol. 31, no. 2, p. 329 –335, 2009.

[12] September 2013. [Online]. Available: http://www.idimages.org/images/browse/ImageTechnique/

[13] T. Ahmad, J. Hu and S. Han, "An efficient mobile voting system security scheme based on elliptic curve cryptography," in *NSS 2009 - Network and System Security*, 2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1 PSNR of various methods whose secret data capacity is 25kb

Image	Method	Block size				
		2x2	3x3	4x4	5x5	6x6
Abdominal	P0	40.33	42.30	42.25	42.64	42.38
	P1	42.75	39.11	39.49	36.94	38.07
	P2	42.64	38.16	37.36	35.33	35.18
	P3	42.45	41.61	40.91	39.13	39.42
	P4	60.75	46.18	61.01	42.31	59.91
	P5	53.90	51.30	50.86	50.57	50.53
Chest	P0	45.69	47.78	48.06	47.03	46.14
	P1	45.62	44.00	43.06	43.14	42.99
	P2	45.40	42.61	41.41	39.37	37.93
	P3	44.37	44.47	43.93	43.35	42.85
	P4	61.03	58.77	53.02	54.12	49.89
	P5	53.95	51.29	50.92	50.53	50.51
Hand	P0	38.43	39.87	43.89	45.04	46.30
	P1	41.60	40.74	42.31	41.26	42.04
	P2	42.59	39.79	41.26	41.29	39.41
	P3	40.47	40.35	40.92	41.96	43.24
	P4	59.25	58.23	54.42	56.44	54.25
	P5	52.83	51.03	51.20	50.29	47.34
Head	P0	44.27	46.43	46.29	46.06	47.54
	P1	44.00	38.94	41.22	40.47	39.94
	P2	43.61	37.64	38.18	37.02	36.67
	P3	44.20	43.12	43.09	42.09	43.79
	P4	60.38	55.02	58.96	43.21	53.68
	P5	53.98	51.21	50.40	50.56	50.54
Leg	P0	42.22	42.66	43.60	44.25	44.08
	P1	45.45	41.73	41.89	41.40	38.93
	P2	46.10	41.02	39.70	38.95	36.62
	P3	44.36	42.85	42.79	41.62	40.15
	P4	59.94	59.37	59.08	56.48	57.82
	P5	53.06	51.24	51.04	50.68	50.87

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2 PSNR of various methods whose secret data capacity is 50kb

Image	Method	Block size				
		2x2	3x3	4x4	5x5	6x6
Abdominal	P0	38.76	39.29	39.09	39.26	39.16
	P1	41.15	37.24	36.74	34.51	34.95
	P2	41.05	36.12	34.18	32.14	31.66
	P3	40.47	39.07	38.01	36.69	36.43
	P4	57.93	46.03	57.91	42.15	56.21
	P5	51.01	48.28	47.89	47.53	47.51
Chest	P0	44.18	45.23	44.56	43.25	42.25
	P1	44.35	42.02	40.65	39.64	38.93
	P2	44.21	40.85	38.57	36.00	33.98
	P3	43.29	42.23	40.77	39.75	38.62
	P4	58.05	56.76	52.37	53.11	47.90
	P5	50.97	48.26	47.81	47.56	47.49
Hand	P0	38.31	39.59	43.37	44.30	45.41
	P1	41.33	40.47	41.65	40.61	41.20
	P2	42.18	39.38	39.98	39.14	36.96
	P3	40.35	40.13	40.59	41.51	42.47
	P4	55.50	54.19	52.28	53.01	49.44
	P5	49.58	47.88	48.18	46.85	45.26
Head	P0	41.96	42.98	42.45	42.06	42.31
	P1	42.19	37.38	38.16	37.45	37.10
	P2	42.01	35.96	35.38	33.94	33.59
	P3	41.98	39.87	39.07	38.44	38.12
	P4	57.06	51.54	53.58	41.23	50.12
	P5	50.96	48.24	47.70	47.54	47.49
Leg	P0	41.13	42.01	42.49	42.60	42.15
	P1	42.20	40.02	40.20	39.33	37.19
	P2	42.90	39.69	38.47	36.83	34.62
	P3	43.09	41.41	41.28	40.09	38.37
	P4	55.20	54.12	52.09	51.04	51.09
	P5	50.09	47.97	47.70	47.14	47.64

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3 PSNR of various methods whose secret data capacity is 100kb

Image	Method	Block size				
		2x2	3x3	4x4	5x5	6x6
Abdominal	P0	38.02	38.02	37.49	37.22	37.08
	P1	40.13	36.21	35.18	33.05	33.16
	P2	39.99	34.99	32.63	30.42	29.36
	P3	39.10	37.33	35.95	34.55	34.01
	P4	47.92	41.45	45.23	38.63	42.02
	P5	47.32	42.43	42.47	41.55	41.71
Chest	P0	42.37	41.72	40.39	39.04	37.93
	P1	42.65	39.54	37.51	36.02	34.82
	P2	42.29	38.23	35.04	32.34	30.11
	P3	41.63	39.29	37.06	35.78	34.55
	P4	52.50	53.22	50.50	50.65	43.63
	P5	47.95	45.24	44.82	44.53	44.45
Hand	P0	37.37	38.08	39.64	39.36	39.04
	P1	39.84	38.02	37.64	36.31	35.96
	P2	40.92	37.64	36.19	34.16	32.24
	P3	39.56	38.49	37.69	37.21	36.35
	P4	51.58	50.33	49.22	49.13	46.69
	P5	46.02	44.84	44.69	43.77	42.38
Head	P0	37.11	38.05	37.69	37.46	37.75
	P1	38.24	34.81	34.33	33.51	33.23
	P2	37.89	33.17	31.92	30.28	29.67
	P3	38.54	36.21	35.25	34.06	34.53
	P4	49.87	49.97	48.21	40.50	43.81
	P5	47.78	45.30	44.76	44.54	44.49
Leg	P0	40.58	40.76	40.69	40.35	39.37
	P1	41.34	38.44	37.75	36.70	34.80
	P2	42.18	38.50	36.60	34.66	32.01
	P3	41.41	40.06	39.29	38.00	36.68
	P4	49.48	47.14	46.29	43.47	43.38
	P5	45.89	44.08	44.03	43.56	43.49

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4 Maximum capacity (in bits) and its corresponding PSNR

Image	Method	2x2		3x3		4x4		5x5		6x6	
		Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Abdominal	P0	194,922	35.69	227,616	34.49	239,205	33.71	242,808	33.05	244,790	32.84
	P1	195,135	37.81	228,016	33.40	239,235	31.62	242,640	29.88	245,350	29.36
	P2	192,180	37.47	220,632	32.05	230,505	29.24	231,192	27.34	232,225	26.07
	P3	258,592	36.02	255,915	33.48	253,184	31.89	251,825	30.52	249,120	29.85
	P4	195,030	37.81	228,096	33.39	239,040	31.63	242,424	29.87	239,540	29.46
	P5	227,835	36.24	226,560	34.60	231,990	33.87	229,968	33.28	225,330	33.34
Chest	P0	195,705	40.91	229,472	39.20	242,715	37.57	246,504	36.23	249,620	35.13
	P1	195,789	41.07	229,568	37.14	242,910	34.53	246,864	32.89	249,865	31.40
	P2	195,363	40.91	228,552	35.91	241,335	31.93	244,992	29.07	247,485	26.53
	P3	259,820	39.10	258,246	36.45	257,232	34.18	257,125	32.61	257,040	31.09
	P4	195,780	41.07	229,648	37.15	242,970	34.54	246,888	32.90	244,125	31.41
	P5	255,108	37.81	246,008	37.40	252,810	36.39	249,528	35.65	245,455	34.99
Hand	P0	196,599	35.58	231,192	34.46	245,715	33.66	249,672	32.90	252,875	32.06
	P1	196,602	37.74	231,192	34.43	245,715	32.22	249,672	30.89	252,875	29.71
	P2	196,599	38.61	231,184	33.86	245,670	30.68	249,528	28.53	252,630	26.55
	P3	261,116	36.25	260,082	33.97	260,048	32.02	260,025	30.66	260,064	29.52
	P4	196,608	37.74	231,192	34.43	245,745	32.26	249,696	30.89	246,925	29.77
	P5	266,757	36.77	271,888	34.58	278,535	33.95	278,736	33.29	273,385	32.45
Head	P0	192,195	34.12	222,384	33.32	233,070	32.62	234,744	32.23	236,075	32.14
	P1	192,711	35.49	223,200	31.48	233,190	30.01	235,608	29.10	236,390	28.71
	P2	190,791	35.34	217,808	30.19	225,705	27.80	224,976	26.33	223,475	25.59
	P3	255,608	33.54	251,640	31.31	248,288	30.07	246,025	29.12	243,432	28.88
	P4	192,579	35.49	223,336	31.49	233,250	30.00	235,896	29.07	231,105	28.69
	P5	238,080	34.65	222,880	34.12	226,455	33.36	221,784	32.86	215,075	32.73
Leg	P0	196,341	38.62	230,104	37.08	244,650	35.63	248,952	34.68	250,810	33.80
	P1	196,353	39.14	230,192	35.40	244,590	33.32	248,352	31.73	251,020	30.26
	P2	196,215	39.00	230,072	34.38	244,065	31.32	247,080	29.03	250,180	27.09
	P3	260,608	38.55	259,029	35.73	258,912	34.21	258,325	32.48	258,192	31.51
	P4	196,368	39.14	230,184	35.43	244,560	33.32	248,376	31.74	246,820	30.76
	P5	249,636	37.16	257,184	36.01	264,525	35.25	261,936	34.64	258,195	34.42