ICIC Express Letters
Part B: Applications
Volume **9**, Number **11**, November **2018**

ICIC International ©2018 ISSN 2185-2766
pp. **1091–1099**

SECURING SECRET DATA BY ENHANCING THE CONTROLLING EXPANSION METHOD

Tohari Ahmad[1] and Siprianus Septian Manek[2]

[1]Department of Informatics
Institut Teknologi Sepuluh Nopember
Kampus ITS Surabaya, Jawa Timur 60111, Indonesia
tohari@if.its.ac.id

[2]Department of Informatics
Universitas Timor
Kefamenanu, Timor Tengah Utara, Nusa Tenggara Timur 85616, Indonesia
epimanek18@gmail.com

Received May 2018; accepted August 2018

Abstract. *In order to protect digital data, various methods have been applied. This includes steganography, which has been popular to use. Here, secret data are embedded in a cover medium. Some methods which have been introduced still have problems in certain factors, for example, the quality of the stego data. In this paper, we enhance the present controlling expansion method, which is variant of difference expansion and reduced difference expansion. This improvement is implemented by using an image as the cover. It is done by controlling the new pixel values, such that the difference from their original value is not greater than 2 or 4, for 0 or 1 bit secret message, respectively. In addition, the possible underflow and overflow values are eliminated. Also, different from other existing methods, in the extraction process to obtain the secret message from the stego data, we do not need a location map. The experimental result, which is performed in grayscale images, shows that the proposed method is better than existing ones. It has been able to increase around 3-4 dB of PSNR (Peak Signal to Noise Ratio) on average, for an equivalent size of secret messages.*

Keywords: Data hiding, Data protection, Information security, Network security, Steganography

1. **Introduction.** In this information and communication world, transferring data through a computer network has been a need for computer users. Data, which can be text, image, audio or video, sometimes are sensitive because they contain confidential information, for example, bank account, military data and medical record. This information must be protected, either in the transmission or storing process.

There are two ways of how the information can be protected, which are: cryptography and steganography. The basic difference between those is: cryptography does not require any medium (called the cover or carrier), while steganography does [1, 2, 3]. Data which are encrypted by any cryptographic algorithm have a very different format, while steganography leads to similar one, called stego data. It is slightly different from watermarking [4]. An advantage of steganography is that it may not attract attention of third party users because the data before and after being embedded are similar. In further research, steganography can be classified into two categories: reversible and irreversible. In both methods, the secret data are successfully extracted. Additionally, the former happens when the method is able to recover the stego data back to the original, while the latter is not. The reversible method considers that both the secret and cover are important. So, those must be generated the same as their original form. This is applied to

DOI: 10.24507/icicelb.09.11.1091

various types of data, for example, medical and military data. Contrarily, the irreversible method focuses only on the secret data. Once the original secret data have been obtained, the stego data are ignored. A simple example of this algorithm is by modifying the LSB (Least Significant Bit) of each pixel based on the secret bit, such as in [5, 6]. Because of its simplicity, this may not be secure enough, in some cases.

Some steganographic methods have been introduced, for example, Difference Expansion (DE) and its improvement, called Reduced Difference Expansion (RDE), which are proposed by Tian [7] and Liu et al. [8], respectively. In further research, Alattar proposes Difference Expansion of Quad (QDE) [9] and difference expansion of generalized integer transform [10]. In the case of medical data, Al-Dmour and Al-Ani [11] propose the use of data coding and edge detection. This is followed by optimized general smoothness which is provided by Holil and Ahmad [12] and fuzzy logic-based image steganography by Karakiş et al. [13]. While this previous research explores image as the medium, we design an RDE-variant algorithm for audio [14]. Recently, Angreni and Ahmad [15] enhance the DE-based method by controlling the expansion. Overall, these existing algorithms require a location map for extracting the secret which has been embedded in the cover. This may cause overhead in the stego data.

In this paper, we propose a method which is able to minimize the dependency of this map. That is, a location map is not needed for extracting the secret. It is only required if the users really want to reconstruct the cover. Furthermore, the proposed method, which is developed based on the Controlling Expansion (CE) [15], intends to increase the quality of the stego data, and removes the possibility of underflow and overflow values in the stego data. Indirectly, this affects the capacity of the secret which can be embedded in the cover.

Based on the experimental results, we find that this proposed method achieves higher PSNR values than existing methods, for an equivalent payload. Moreover, its standard deviation is lower than others. It is shown that the proposed method is superior.

The rest of the paper is as follows. Section 2 describes the previous research. Section 3 presents the proposed method whose experimental results are provided in Section 4. Finally, the conclusion is drawn in Section 5.

2. Difference Expansion and Its Variations.

Difference Expansion (DE) [7] employs difference between a pair of pixel values in the 8-bit grayscale image. Let u_1 and u_2 be a pair of pixels in an image, the difference v and the average m can be represented in (1).

$$\left.\begin{aligned} m &= \left\lfloor \frac{u_1 + u_2}{2} \right\rfloor \\ v &= u_2 - u_1 \end{aligned}\right\} \tag{1}$$

The embedding is done by expanding the difference v. If the secret bit is depicted as b, then the embedding process follows (2). Here, $\overline{v}$ represents the difference after being embedded.

$$\overline{v} = 2v + b \tag{2}$$

In order to obtain the new pixel value, (3) is implemented. This process needs m which does not experience any change in the previous steps. The new pixels, u_1' and u_2', must not have value more than 255 (called overflow) or less than 0 (called underflow).

$$\left.\begin{aligned} u_1' &= m + \left\lfloor \frac{\overline{v}+1}{2} \right\rfloor \\ u_2' &= m - \left\lfloor \frac{\overline{v}}{2} \right\rfloor \end{aligned}\right\} \tag{3}$$

Reduced Difference Expansion (RDE) [8] is developed based on the assumption that the difference which is generated by DE [7] is relatively high. This causes the quality of the stego data relatively low. Based on this, Liu et al. [8] propose a method to solve the problem. Here, the difference is reduced before being embedded by the secret. If the difference is greater than or equal to 2, then it is reduced; otherwise the difference does not change as in (4).

$$\overline{v} = \begin{cases} v, & \text{if } v < 2 \\ v - 2^{\lfloor \log_2 v \rfloor - 1}, & \text{if } v \geq 2 \end{cases} \tag{4}$$

In this process, complement data, which are the location map, are needed to store $\overline{v}$. The extraction process is performed by firstly checking this location map, based on the pixel of the stego data. According to the experimental results, RDE [8] has been able to improve the quality of the stego data, which is represented by a PSNR value. This has made the stego data more similar to the cover than that of DE [7]. In terms of the capacity (size of the secret data), however, RDE is almost the same as that of DE. That is, it needs 2 pixels to hide 1 bit data. By referring to this fact, Alattar [9] implements a method, called Difference Expansion of Quad (QDE), which is able to rise the capacity. In this algorithm, he employs blocks whose dimension is 2×2. So, each block consists of 4 pixels which may be used to hide 3 bits.

In 2016, Angreni and Ahmad [15] proposed the Controlling Expansion (CE) method. It has been able to increase both the capacity of the secret message and the quality of the stego data. They only use a pixel to embed 1 bit secret b. In this method, a random value R is implemented for calculating the difference h and average l, as depicted in (5). If the difference is neither 0 nor 1, then this difference is reduced by using (6) for further decreasing the difference and having h'.

$$\begin{aligned} h &= x - R \\ l &= \begin{cases} \left\lfloor \dfrac{x+R}{2} \right\rfloor, & \text{if } h \leq 1 \\ \left\lceil \dfrac{x+R}{2} \right\rceil, & \text{if } h > 1 \end{cases} \end{aligned} \tag{5}$$

$$h' = \begin{cases} |h|, & \text{if } h = 0 \text{ or } h = 1 \\ \left\lceil \dfrac{|h|-1}{2} \right\rceil, & \text{if otherwise} \end{cases} \tag{6}$$

After reducing the difference, the next step is to get the new one. The embedding process considers the initial difference h as in (7), and the new pixel value x' is obtained by improving (3). This is also to prevent new pixels from both underflow and overflow conditions.

$$h'' = \begin{cases} 2 \times h' - b, & \text{if } h < 0 \\ 2 \times h' + b, & \text{if } h \geq 0 \end{cases} \tag{7}$$

3. **Enhancing the Controlling Expansion.** This proposed method is an improvement of the existing ones, especially the controlling expansion [15]. Furthermore, this scheme is able to embed 1 bit secret data in each pixel. In most previous algorithms, both extracting the secret and reconstructing the cover processes rely on the existence of the location map. In this research, we have removed the need of the location map for extracting the secret bit. This prevents attackers from intercepting the location map itself, and attacking the location map does not work at all since it does not exist. Therefore, in the case that the original cover is not necessary, the proposed method can be a solution.

3.1. **Embedding process.** The embedding process is the step for hiding the secret in the cover. As in this research the cover is an image, the value of pixels is manipulated according to the value of both pixels and secret bits. Inspired by our previous research [15], we present an improvement whose embedding process is depicted in Figure 1.

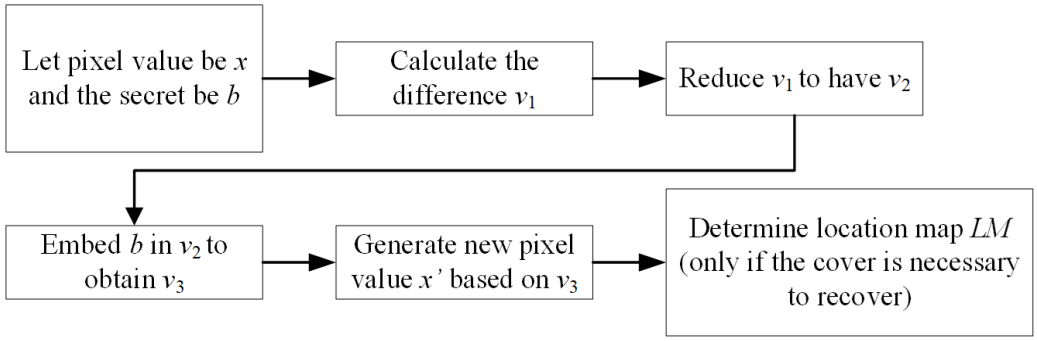

FIGURE 1. Embedding process

It is assumed that the cover is a gray level image; so, the range of pixel value is 0-255. Firstly, the secret message is converted to bits, each of which is processed one by one. In the previous method [15], in order to obtain a difference, a random value is needed. Differently, in this paper we propose to have the difference v_1 by adding the pixel value with either 6 or 2 as depicted in (8).

$$v_1 = \begin{cases} \left\lfloor \dfrac{x+6}{2} \right\rfloor, & \text{if } x \leq 251 \\ \left\lfloor \dfrac{x+2}{2} \right\rfloor, & \text{if } x > 251 \end{cases} \tag{8}$$

Here, there are 2 conditions which should be met by the pixel value x. That is, when x is less than or equal to 251, it must be added by 6 before being divided by 2; otherwise, 2 is used instead of 6. These two values are the best choice according to the experiment for preventing the stego pixels from both underflow and overflow. For example, if $x = 0$ is added by 2, then underflow occurs in the next step; equivalently, if $x = 252$ is added by 6, then it results to overflow.

The next step is to get the reduced difference v_2 whose calculation is provided in (9). Since the secret bit needs a space for embedding, v_1 must be subtracted by 1. The resulted value is divided by 2 in order to minimize the effect of the embedding process to the pixel value.

$$v_2 = \left\lfloor \frac{v_1 - 1}{2} \right\rfloor \tag{9}$$

The secret bit is embedded to v_2 by using (10), similar to [7]. In the case that the secret message is the only focus, the next step is generating new pixels. Furthermore, the information of the embedding properties is not needed. This is because the extraction of the secret does not need this information. If the original cover must be obtained, however, then a location map (LM) must be generated. This contains information of which pixels that have been processed. In the previous methods, the information is compulsory because data extraction and recovery rely on it. Here, LM is designed as simple as possible in order to reduce the complexity. It is just the difference between the new and old pixel values, which also means the difference between the stego and the cover. In both extraction and recovery cases, generating new pixels is necessary, whose calculation is presented in (11); while LM is produced by using (12).

$$v_3 = 2 \times v_2 + b \tag{10}$$

$$x' = 2 \times (v_3 - 1) \tag{11}$$

$$LM = x' - x \tag{12}$$

Overall, the proposed embedding process is an improvement to [15]. In more details, this process can be explained in the following example.

1) Let the pixel and the secret bit values be 250 and 0, respectively.

$$x = 250, \qquad b = 0$$

2) Calculate the different value.

$$v_1 = \left\lfloor \frac{x+6}{2} \right\rfloor \Rightarrow v_1 = \left\lfloor \frac{250+6}{2} \right\rfloor = 128$$

3) Obtain the reduced difference value.

$$v_2 = \left\lfloor \frac{v_1 - 1}{2} \right\rfloor \Rightarrow v_2 = \left\lfloor \frac{128-1}{2} \right\rfloor = 63$$

4) Embed the secret data bit b in v_2 to obtain v_3.

$$v_3 = 2 \times v_2 + b \Rightarrow v_3 = 2 \times 63 + 0 = 126$$

5) Calculate the new pixel value x' (and location map LM, if it is needed).

$$x' = 2 \times (v_3 - 1) \Rightarrow x' = 2 \times (126 - 1) = 250$$

$$LM = x' - x \Rightarrow LM = 250 - 250 = 0$$

The new pixel is 250 and the location map is 0. It is worth noting that the location map is only used to recover the original image (cover/carrier), and it is not required for extracting the secret message.

3.2. **Extraction and recovery processes.** Extraction and recovery are the process of getting back the secret and the original cover data, respectively. Generally, both processes are the inverse of the embedding one. What we need to do is just to start from the latest step of embedding. In order to get the secret, the last bit of each pixel (LSB) is taken. The collected bits are then combined for constructing the message. The overall steps of this method are presented in Figure 2.

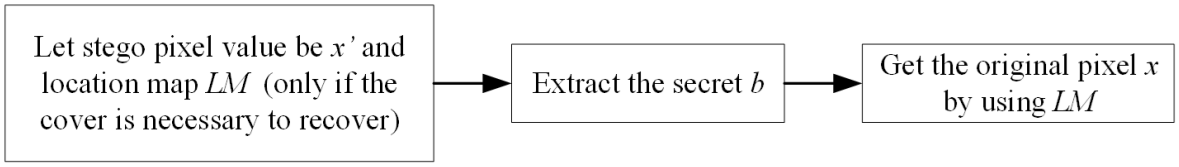

FIGURE 2. Extracting and recovering processes

The extraction process is carried out directly to each pixel by using (13), where $LSB(x)$ is the function to take the LSB value of x. In recovering the cover, the stego pixels are simply subtracted by the value of LM. This is because, as described in the previous section, LM is the difference between the stego and the cover. This recovery step can be denoted in (14).

$$b = LSB\left(\left(\frac{x'}{2}\right) + 1\right) \tag{13}$$

$$x = x' - LM \tag{14}$$

An example of these extraction and recovery steps is illustrated as follows, which relates to the previous embedding step. It is shown that this proposed method is reversible and is able to overcome the overflow and underflow problems.

1) Take the pixel values of stego image and location map.

$$x' = 250, \qquad LM = 0$$

2) Extract the value of the bit b to get the secret data.

$$b = LSB\left(\left(\frac{250}{2}\right) + 1\right) = LSB(126) = 0$$

3) Recover the original image pixel value by subtracting the pixel value of stego images with location map value.

$$x = x' - LM = 250 - 0 = 250$$

4) Finally, we have the original pixel value $x = 250$ and the secret bit $b = 0$.

4. **Experimental Result.** Similar to [15], in this research we evaluate the performance of the proposed method by measuring the Peak Signal to Noise Ratio (PSNR) in order to find the similarity between the cover and the stego images with the respective number of secret bits which can be hidden in it. Additionally, for measuring the amount of secret data, we use bit per pixel (bpp). This value is calculated by dividing the number of embedded bits by the number of pixels in the cover. For this evaluation purpose, we use a public image database [16] and a text generator [17] for the cover and payload, respectively. In addition, other related methods [7, 8, 15] are also implemented for comparison.

The capacity of the proposed method, similar to [15], is 1 bpp. It is by assuming that all pixels are able to accommodate the secret bit. It is also found that in terms of the capacity of the secret data, DE [7] and RDE [8] have the same value, which is 0.5 bpp. This is because those two methods need 2 pixels for hiding 1 bit data, while CE [15] and our proposed method need 1 pixel for the same purpose. In [15], however, this bpp value is achieved by separating the location map from the stego data. Different from this, we are able to obtain this bpp value because the location map is not required, in case that the secret message is the only necessary data.

Next evaluation is to embed various sizes of payload: 7 Kb, 35 Kb, 105 Kb and 210 Kb whose results are depicted in Tables 1, 2, 3 and 4, respectively. These sizes represent different types of data according to their common sizes, such as password and description which usually have relatively small and large number of bits. It is worth noting that as shown in Table 4, DE [7] and RDE [8] are not able to hold the given amount of payload (210 Kb). They only accommodate 131072 bits for each gray scale image. This is because those two methods need 2 pixels for carrying 1 bit, different from [15] and the proposed method.

According to Tables 1, 2, 3 and 4, as predicted, increasing the secret size leads to decreasing the PSNR value for all methods. This is because the amount of pixels whose value changes are going up, which leads to more noises. It is also shown in those tables that the proposed method has relatively similar PSNR values for various cover images with the same payload size. Furthermore, the average standard deviation values of DE

Table 1. Quality of stego data with 7 Kb of secret data

Cover Image	**PSNR (dB)**			
	DE [7]	**RDE [8]**	**CE [15]**	**Proposed Method**
Lena	55.66	64.28	70.86	73.07
Baboon	42.22	46.84	70.88	73.16
Boat	51.13	57.66	69.72	72.87
Pepper	51.09	58.12	68.56	73.16
Plane	55.66	61.33	66.39	73.36

[7], RDE [8], CE [15] and the proposed methods are 3.78, 4.16, 1.86 and 0.18, respectively. Based on these values, it can be inferred that the proposed method is more stable than others.

TABLE 2. Quality of stego data with 35 Kb of secret

Cover Image	PSNR (dB)			
	DE [7]	RDE [8]	CE [15]	Proposed Method
Lena	46.43	54.11	64.00	66.14
Baboon	36.55	42.08	64.86	66.15
Boat	42.12	48.28	61.42	65.73
Pepper	43.73	50.37	61.90	66.23
Plane	49.92	54.93	58.85	66.30

TABLE 3. Quality of stego data with 105 Kb of secret

Cover Image	PSNR (dB)			
	DE [7]	RDE [8]	CE [15]	Proposed Method
Lena	38.64	45.69	57.89	61.36
Baboon	33.59	39.95	58.13	61.42
Boat	36.17	42.06	56.47	61.06
Pepper	38.76	45.27	57.73	61.47
Plane	38.99	44.75	53.85	61.43

TABLE 4. Quality of stego data with 210 Kb of secret

Cover Image	PSNR (dB)			
	DE [7]	RDE [8]	CE [15]	Proposed Method
Lena	–	–	55.24	58.36
Baboon	–	–	55.29	58.41
Boat	–	–	54.63	58.08
Pepper	–	–	54.93	58.45
Plane	–	–	51.65	58.47

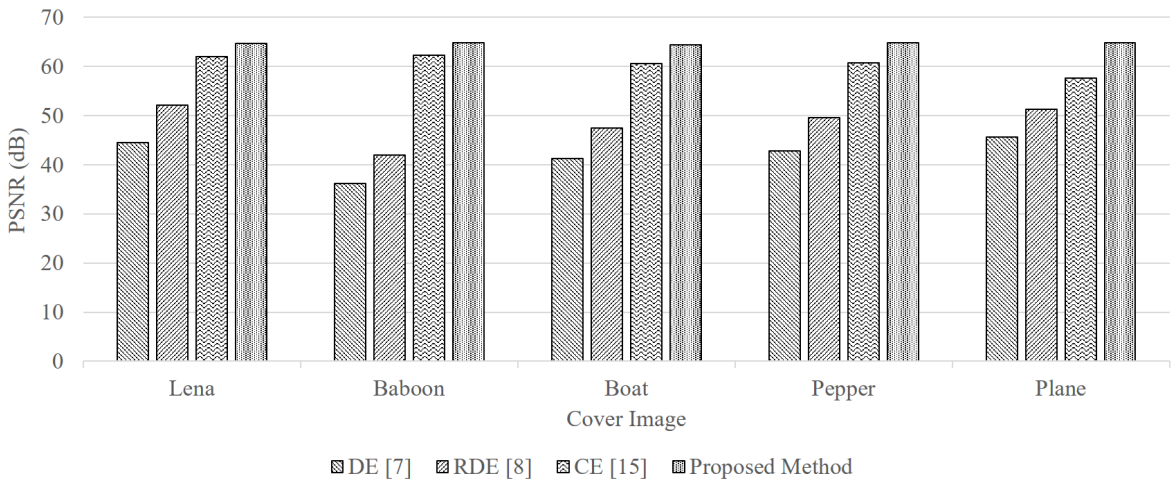

FIGURE 3. Average of PSNR values of methods with various payloads

Additionally, the average of PSNR values of various payload sizes for each method is presented in Figure 3. It depicts that overall, for all secret sizes, the proposed method is better than others. Furthermore, this method restricts the difference of pixel value between before and after the embedding process. That is, the difference is 2 and 4 for the secret bit of 0 and 1, respectively. So, this difference value can be kept as low as possible to make the stego pixel relatively close to its respective original value.

Also, this proposed method does not specify the average value of the pixels. This is different from CE [15], which employs a random value R, as well as DE [7] and RDE [8] which use another closer pixel (pair of pixels) for the calculation. In this proposed method, therefore, there is no other factor which influences the change of the pixel value. So, the value of a stego pixel is independent which does not rely on its neighbor pixels.

5. **Conclusion.** This paper provides an algorithm for hiding a secret message in an image. According to the experimental results, this method is able to increase the performance, which is measured by the capacity of the secret message that can be embedded, and the quality of the stego data. The possibility of underflow and overflow condition can be minimized by establishing a certain constant at the beginning of the embedding process.

Furthermore, this proposed method does not need any location map at all for obtaining the secret message from the stego data. Only if the cover is necessary to recover, the location map is required. This characteristic has been able to save the storage. In the future, this research may be extended to either RGB or non-RGB images for the hiding medium, so that it is also implementable for other image types and characteristics, such as medical images. This can be done by processing each component of the respective image.

REFERENCES

[1] X. Yan, S. Wang, X. Niu and C.-N. Yang, Halftone visual cryptography with minimum auxiliary black pixels and uniform image quality, *Digital Signal Processing*, vol.38, pp.53-65, 2015.

[2] M. Liśkiewicz, R. Reischuk and U. Wölfel, Security levels in steganography – Insecurity does not imply detectability, *Theoretical Computer Science*, vol.692, pp.25-45, 2017.

[3] M. Y. M. Parvees, J. A. Samath and B. P. Bose, Dual imaging-based reversible hiding technique using protecting large size medical images with logistic map using dynamic parameters and key image matching, *Int. J. of Network Security*, vol.19, no.6, pp.984-994, 2017.

[4] P. Selvam, S. Balachandran, S. P. Iyer and R. Jayabal, Hybrid transform based reversible watermarking technique for medical images in telemedicine applications, *Optik – International J. for Light and Electron Optics*, vol.145, pp.655-671, 2017.

[5] H. Dadgostar and F. Afsari, Image steganography based on interval-valued intuitionistic fuzzy edge detection and modified LSB, *J. of Information Security and Applications*, vol.30, pp.94-104, 2016.

[6] T. Lu, C. Tseng and J. Wu, Dual imaging-based reversible hiding technique using LSB matching, *Signal Processing*, vol.108, pp.77-89, 2015.

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

[8] C.-L. Liu, D.-C. Lou and C.-C. Lee, Reversible data embedding using reduced difference expansion, *The 3rd Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, 2007.

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

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

[11] H. Al-Dmour and A. Al-Ani, Quality optimized medical image information hiding algorithm that employs edge detection and data coding, *Computer Methods and Programs in Biomedicine*, vol.127, pp.24-43, 2016.

[12] M. Holil and T. Ahmad, Secret data hiding by optimizing general smoothness difference expansion based method, *J. of Theoretical and Applied Information Technology*, vol.72, no.2, pp.155-163, 2015.

[13] R. Karakiş, I. Güler, I. Çapraz and E. Bilir, A novel fuzzy logic-based image steganography method to ensure medical data security, *Computers in Biology and Medicine*, vol.67, pp.172-183, 2015.

[14] M. B. Andra, T. Ahmad and T. Usagawa, Medical record protection with improved GRDE data hiding method on audio files, *Engineering Letters*, vol.25, no.2, pp.112-124, 2017.
[15] D. S. Angreni and T. Ahmad, Enhancing DE-based data hiding method by controlling the expansion, *The 4th Int. Conf. on Information Technology for Cyber and IT Service Management*, Bandung, Indonesia, 2016.
[16] *The USC-SIPI Image Database*, University of Southern California, http://sipi.usc.edu/database /database.php, 2018.
[17] *Lorem Ipsum*, http://www.lipsum.com/, 2016.