Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Computer Science and Computational Intelligence 2018

Implementation of Riversible Data Hiding Using Difference Histogram Shifting in Encrypted Medical Image

Muhammad Fadhlan Putranto[a], Second Author[b], Third Author[a,b,*]

[a] *First affiliation, Address, City and Postcode, Country*
[b] *Second affiliation, Address, City and Postcode, Country*

Abstract

In this paper, we implemented reversible data hiding (RDH) in encypted domain (medical image) directly. Specifically, our algoritma is based on the histogram shifting technique. RDH can be accomplished in ecrypted domain directly, since the correlation between the neighboring pixel preserved. Therefore, the RDH scheme have been designed according to the encryption algorithm utilized. In this paper, plain image encrypted using specific encryption algorithm that consists of two process (stream encryption algorithm and permutation blok). Since the correlation between the neighboring pixel can be preserved in encrypted image, RDH schemes can be applied to the encrypted image directly.

Keywords: reversible Data Hiding; Difference Histogram Shifting; encrypted Image;

1. Introduction

Rapid development of information technology as it is today, especially the Internet, bring a variety of convenience. Internet can connect almost all of the computers in the world. It can make all the computers can easily exchange data quickly and efficiently. no exception exchange digital images or other multimedia information. This has made digital image security more important and attracts a lot of attention. Encryption, steganography is the techniques which can be used to provide security.

Steganography is the art of hiding classified information inside any media file such as images, audio or video to produce a secret that integrates with a cover image called a stego image, so that its secrets can not be recognized or recovered by unauthorized recipients [1]. Data hiding algorithms can be classified into two categories: irreversible data hiding and reversible data hiding. In irreversible data hiding algorithm the cover image cannot be completely recovered, so these algorithms are not suitable for medical image, but in the reversible data hiding, cover image

* Corresponding author. Tel.: +62-813-1218-5690 ; fax: +0-000-000-0000.
E-mail address: fadhlanputranto

can be completely recovered[2]. Reversible data hiding will have benefits when precision is taken care of, such as medical images. changes can be deliberately dangerous or inadvertently affect the content interpretation. For example, unintentional changes in X-ray images may cause misdiagnosis, which may result in criminal and legal offenses that could harm various parties. Therefore RDH is designed to solve the problem.

The classical RDH schemes have been proposed based on three fundamental strategies: lossless compression[3], difference expansion (DE)[4], and histogram shifting (HS)[5]. In this paper, we used difference histogram shifting (DHS)[6] algorithm for data hiding. we used DHS because of their large embedding capacity and high fidelity. the main idea is to explore the correlation between the neighboring pixels in a host image. Then the additional message can be reversibly embedded into the host image via modifying the difference histogram. However, using this method cannot be embeding additional message into the encrypted image directly because correlation between the neighboring pixel does not exist anymore.

In this paper, we focus on preserved correlation between neighboring pixel and implemented the RDH method proposed in the encrypted domain directly.

2. Previous Method

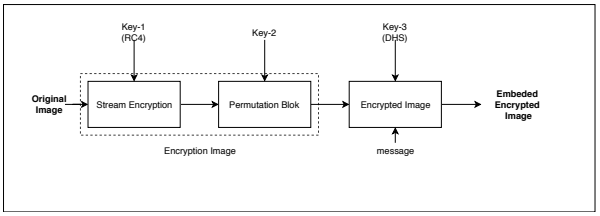

Fig. 1. Block Diagram Encryption and Data Hiding

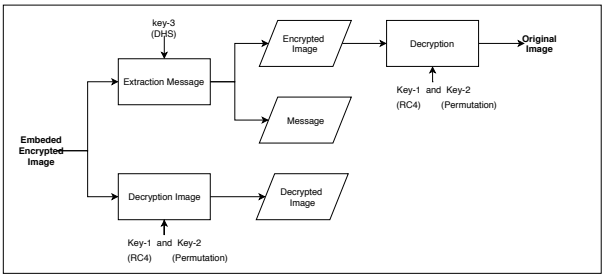

Fig. 2. Block Diagram Extraction and Decryption Image

3. Proposed Method

In this section, we explain our design system from block diagram in detail. Our system consist of two part. One part image encryption and data hiding process,as shown in Fig.1; and the other part is data extraction and image recovery process, as shown in Fig.2.

In the encryption and data hiding process, the plain image is firstly divided into sub-blocks. Then, via a specific stream cipher, the divided image is encrypted with the encryption key key-1. After that, the sub-blocks of the stream encrypted image are permutated with the permutation key key-2, and the encrypted image is obtained. Additional data are reversibly embedded into the encrypted image data hiding key key-3. in this paper, the RDH scheme selected for data hiding is previously proposed DHS based approaches.

In the extraction and decryption process, we can do two scenario. In the first scenario, the receiver will decrypt the image with the decryption keys (i.e., key-2 and key-1) directly, and the decrypted image is similar to the original host image. In the second scenario, firstly the additional hidden data is extracted and meanwhile the encrypted image is reversibly recovered with data hiding key-3. Then the restored encrypted image is decrypted to obtain the original host image with the decryption keys (i.e., key-2 and key-1).

3.1. Encryption Image Algorithm

The encryption algorithm used in this paper includes two steps; specific stream encryption and permutation. Huang *et al.*[7] proposed new method for stream encryption to preserved correlation between neighboring pixel after encryption image.

- Specific Stream Encryption

  Before encryption, the plain image I is divided into N non-overlapping sub-blocks $B1, B2, \ldots, BN$. The sub-blocks are with the size of $m \; x \; n$ and scanned in the order from left to right and then top to bottom. Let $P_{i,j}(1 \leq i \leq N, 1 \leq j \leq m \; x \; n)$ denotes one of the pixels in sub-block B_i, where i represents the index of a sub-block, and j represents the index of the pixel in the sub-block B_i . In each sub-block, the pixels are also scanned from left to right and then top to bottom. The pixel value can presented in Fig.3.

  According to the encryption key-1, we used RC4 algorithm to generate the key stream with the length of N bytes. We represented the key with $K_i(1 \leq i \leq \mathrm{N})$, where i is the index of the generated key from RC4 algorithm.

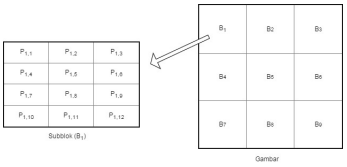

Fig. 3. Represented Pixel Value in each Block

In encryption process, the bitwise exclusive-or (XOR) operation is performed between $P_{i,j}$ and K_i , as shown in Eq.1.

$$E_{i,j} = P_{i,j} \oplus K_i (1 \leq j \leq m \; x \; n) \tag{1}$$

- Permutation Block

  In this phases, we permute all sub-block with key-2. We can get permutation key from Eq.2. Where P is prime number which bigger than N (sub-block) and G is number between 1 and P-1 and x is the index of the sub-block ($1 \leq x \leq N$). let $Y_i (1 \leq i \leq N)$ denotes generated number from Eq.2. Then we swap all of pixel in the sub-block i with sub-block Y_i. In this step, we only permute sub-block and the pixel within each sub-block still preserved.

$$Y_i = G^x \; mod \; P \tag{2}$$

3.2. Data Hiding

Since encrypted image have the higher points of difference histogram still exist and correlation between the neighboring pixel still preserved, we can easily accomplished RDH algorithm in the ecrypted doamin. We devide the encrypted image into sub-block and the pixel in the encrypted image is represented by $C_{i,j} (i \leq N_E, j \leq m_E x n_E)$,where N_E represents the number of sub-block and $m_E x n_E$ represented the size of the sub-block.

To avoid the saturation (i.e., the overflow or underflow) during the embedding process, we modifying the pixel with pixel value 0 or 255 and noted in a location map L (initialized to be empty) as that in[8]. To do this, visit pixels sequentially and append a bit 0 to L when $C_{i,j} 1, 254$. If $C_{i,j} 1, 255$, append a bit 1 to L and modify $C_{i,j}$ to $C'_{i,j}$ using Eq.3.

$$C'_{i,j} = \begin{cases} 254 & if \; C_{i,j} = 255 \\ 1, & if \; C_{i,j} = 0 \\ C_{i,j}, & otherwise \end{cases} \tag{3}$$

After preprocessed pixel, we embedded additional message into encrypted domain via modifying the difference histigram. difference value in each block is compute using Eq.4.

$$D_{ij} = C_{i,j} - C_{i,1} \tag{4}$$

And then the embedding algorithm of the RDH scheme using Eq.5

$$C''_{i,j} = \begin{cases} C'_{i,j} - 1, & jika\ D_{i,j} < -1 \\ C'_{i,j} - b, & jika\ D_{i,j} = -1 \\ C'_{i,j} + b, & jika\ D_{i,j} = 0 \\ C'_{i,j} + 1, & jika\ D_{i,j} > 0 \end{cases} \tag{5}$$

where $b \subset [0,1]$ is a message bit to embedded and message will embedded if difference value equal -1 or 0.

3.3. Extraction and Decryption Image

As we describe before, In this process , we can do two scenario. In the first scenario, we decrypt the image image with decryption key (i.e., key-2 and key-1) directly. First, using Eq.2 we permutate sub-block from sub-block N until sub-block 1. Then, we generate random number (RC4) and perform the bitwise exclusive-or (XOR) operation to each sub-block in the image following Eq.1. In the second scenario, the additional hidden data is extracted following Eq.6.

$$b^* = \begin{cases} 0, & jika\ C''_{i,j} - C''_{i,1} = 0, -1 \\ 1, & jika\ C''_{i,j} - C''_{i,j} = 1, -2 \end{cases} \tag{6}$$

where b^* represent the extracted message bit. In the same time, restore the image using Eq.7.

$$C'^{\,*}_{i,j} = \begin{cases} C''_{i,j} - 1, & jika\ C''_{i,j} - C''_{i,1} > 0 \\ C''_{i,j} + 1, & Jika\ C''_{i,j} - C''_{i,j} < -1 \\ C''_{i,j}, & lainnya \end{cases} \tag{7}$$

where $C'^{\,*}_{i,j}$ represent the restore pixel value. After image restoration, the original encrypted image can be recovered via using the extracted location map following Eq.8.

$$C^*_{i,j} = \begin{cases} 255, & jikaC'^{\,*}_{i,j} = 254 \\ 0, & jikaC'^{\,*}_{i,j} = 1 \\ C'^{\,*}_{i,j}, & otherwise \end{cases} \tag{8}$$

4. Experiment Result and Analysis

4.1. Experiment on Typical image

To denote that the proposed method can be implemented in medical image,in this paper we used twelve test image as shown in Fig.4. The test image have size 1024 x 1024 pixel. RDH in test image aims at developing a method that preserved the correlation between neighboring pixel and increases the embedding capacity as high as possible. Most

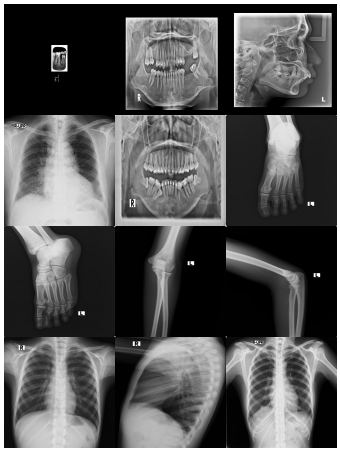

Fig. 4. Test Image from left to right, up to down a) Example 1. b) Example 2. c) Example 3. d) Example 4. e) Example 5. f) Example 6. g) Example 7. h) Example 8. i) Example 9. j) Example 10. k) Example 11. l) Example 12.

importantly, since the encryption process and the embedding process can be done independently. That is, even if the people who does not know the encryption algorithm that has been used by the content owner, he/she can apply RDH algorithm to encrypted image directly.

4.2. Experiment Result

In this Section, we describe our experiment using proposed method introduced in section 3. First,we encrypted (stream encryption and permutation) a series of stego-image. In this phase, it will produce stream image (SI) and the permutated stream image (PSI). we used the sub-blok size 3 x 3 for the encryption. In the next, some test result about the distribution of pixel value and the difference pixel value are illustrated. for ease of explanation, the representative image ”Example 1” is selected for a demosntration. To denote that the correlation between neighboring pixel still preserved after encryption, in each sub-block the difference value are compute according to Eq.4 and it is observed from Fig.5(a)-(c). The difference value for all pixel in plain imaga (PI), as shown in Fig.5, most of the values are with small magnitudes (*e.g.,* 0,±1,±2,±3,...) and after stream encryption permutation the difference value are randomly distributed in the whole range of the image but still with small magnitudes,as shown in Fig.5. Thats means, the distribution of the plain image, stream image and permutation stream image may have the similar or even the same distribution.

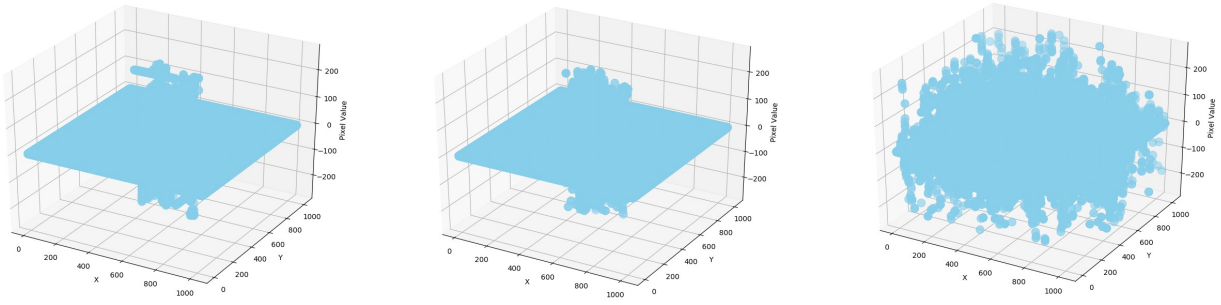

Fig. 5. The difference value for all pixel, from left to right a) Plain Image. b) Stream Image. c) Pemutated Image.

For the test, we embedded different length of the message. for ease explanation, DHS1,DHS2 and DHS2 represent length of the message are 16000 bit, 360000 bit and 640000 bit. The embedding capacity (EC) are illustrated in

Table 1. The Embedding Capacity and Visual Quality

	DHS1		DHS2		DHS3	
	EC	PSNR	EC	PSNR	EC	PSNR
Example 1	911997	31.95	911997	28.57165	911997	26.40822
Example 2	375320	26.52	375320	25.51501	375320	25.40996
Example 3	381175	26.4	381175	25.50724	381175	25.42543
Example 4	176035	25.22	176035	25.19485	176035	
Example 5	99713	25.98	99713	25.98612	99713	25.98992
Example 6	686576	28.83	686576	27.96458	686576	26.85564
Example 7	614096	28.12	614096	27.35289	614096	26.48029
Example 8	712379	28.33	712379	26.7831	712379	25.23355
Example 9	695552	28.2	695552	26.62274	695552	25.09933
Example 10	289499	25.91	289499	25.61252	289499	
Example 11	299350	26.14	299350	25.78227	299350	
Example 12	346925	25.54	346925	25.00996	346925	

Table.1. In Table 1. Considering about that in saparable RDH in encrypted image[9], the receiver can decrypt the embedded encrypted image to get an approximated original image, the peak signal-to-noise ratio (PSNR) values between original image and the directly decrypted image are given in this experiment.

4.3. Security Analysis

in implementation of encryption process, there are use stream encryption and permutation algorithm. in stream encryption the pseudorandom generator should be unpredictable and the key should never be reused. However,in our implementation, the generate key bytes have been reused for $m\ x\ n$ times, where $m\ x\ n$ is the the block size. So, The encryption process may leak the difference information between the neighboring pixel in the same sub-block.

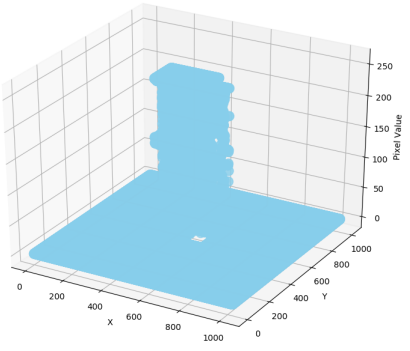

Fig. 6. The distribution of the pixel values of the plain image

For example[7], two pixels ($P_{i,x}$, $P_{i,y}$) in the same sub-block are encrypted with the same key stream byte R_i. The encrypted of the pixels represented :

$$E(P_{i,x}) = P_{i,x} \oplus R_i$$
$$E(P_{i,y}) = P_{i,y} \oplus R_i \quad (9)$$

Since xor is commutative and has the property that (X xor X) = 0 (self-inverse), an adversary has intercepted E(Pi,x) and E(Pi,y) can easily compute:

$$\begin{aligned} E(P_{i,x}) \oplus E(P_{i,y}) &= (P_{i,x} \oplus R_i) \oplus (P_{i,y} \oplus R_i) \\ &= P_{i,x} \oplus P_{i,y} \oplus R_i \oplus R_i \\ &= P_{i,x} \oplus P_{i,y} \end{aligned} \tag{10}$$

According the analysis, we know that if anyone intercepts two pixels encrypted with the same key , he/she can recover the difference value between the two pixels. To secure our proposed framework, a permutation algorithm is conducted following Eq.2. After permutation process, the difference signal from plain images may have the similar or even the same distribution. Using random permutation process make at most N! different permutated patterns that can be generated. Where N is the number of devided sub-block. Because of N! is a very large number, it makes no one to one mapping that can be created between the difference signal and permutated pattern.

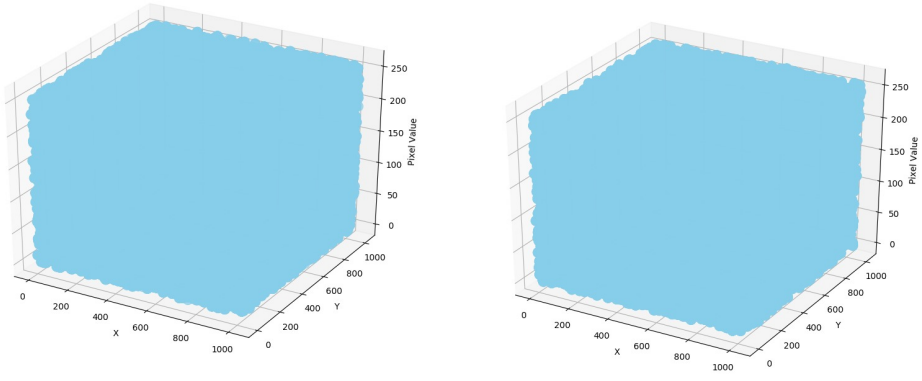

Fig. 7. The distribution of the pixel values of the plain image

In Fig.6,we have illustrated the original pixel value of ”Example 1” image. The X axis , Y axis and Z axis represent row index, column index and pixel value of the image. Observed from Fig.6, the pixel value are continuous in the plain image. However, after stream encryption and permutation, the pixel value of the encrypted image are uniformly distributed in the range [0,255] as shown in Fig.7. thats means without the permutation key, the attacker cannot restore the permutated sub-block according to the encrypted pixel value. So, they cannot restore the sub-block arrangement of the stream encrypted image.

5. Conclusions

In this era, the security of medical image very important. Changing a medical image can be deliberately dangerous or inadvertently affect the content interpretation. To avoid that, RDH scheme conducted. RDH can recover medical image similar to original image after decryption. And most important, the encryption process and embedding process can be done independently each other.So, this is suitable for the separable data-Hiding scheme in encypted domain[9]. After our encryption process, The correlation between neighboring pixel definitely preserved and provide large embedding capacity, we can reversibly embedding large secret message (diagnosis) into encrypted image directly using our RDH scheme.

References

1. Al-Bahadili, H.. A secure block permutation image steganography algorithm. *International Journal on Cryptography and Information Security (IJCIS)* 2013;**3**(3).
2. Ramaswamy, R., Arumugam, V.. Lossless data hiding based on histogram modification. *Int Arab J Inf Technol* 2012;**9**(5):445–451.
3. Barton, J.M.. Method and apparatus for embedding authentication information within digital data. 1997. US Patent 5,646,997.
4. Tian, J.. Reversible data embedding using a difference expansion. *IEEE transactions on circuits and systems for video technology* 2003; **13**(8):890–896.
5. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.. Reversible data hiding. *IEEE Transactions on circuits and systems for video technology* 2006; **16**(3):354–362.
6. Lee, S.K., Suh, Y.H., Ho, Y.S.. Reversiblee image authentication based on watermarking. In: *Multimedia and Expo, 2006 IEEE International Conference on*. IEEE; 2006, p. 1321–1324.
7. Huang, F., Huang, J., Shi, Y.Q.. New framework for reversible data hiding in encrypted domain. *IEEE Transactions on Information Forensics and Security* 2016;**11**(12):2777–2789.
8. Yin, Z., Luo, B., Hong, W.. Separable and error-free reversible data hiding in encrypted image with high payload. *The Scientific World Journal* 2014;**2014**.
9. Zhang, X.. Separable reversible data hiding in encrypted image. *IEEE transactions on information forensics and security* 2012;**7**(2):826–832.