



**HAL**  
open science

## A New Fast Reversible Method for Image Safe Transfer

William Puech, José Marconi Rodrigues, Jean-Eric Develay-Morice

► **To cite this version:**

William Puech, José Marconi Rodrigues, Jean-Eric Develay-Morice. A New Fast Reversible Method for Image Safe Transfer. *Journal of Real-Time Image Processing*, 2007, 2 (1), pp.55-65. 10.1007/s11554-007-0045-x . lirmm-00192119

**HAL Id: lirmm-00192119**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00192119>**

Submitted on 10 Dec 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

W. Puech · J.M. Rodrigues · J.E. Develay-Morice

# A New Fast Reversible Method for Image Safe Transfer

**Abstract** In this paper a novel reversible method for fast and safe image transfer is proposed. The method combines compression, data hiding and partial encryption of images in a single processing step. The proposed approach can embed data into the image according to the message size and partially encrypt the image and the message without changing the original image content. Moreover, during the same process the image is lossless compressed. Nevertheless, the compression rate depends on the upper bound of message size to embed in the image. The main idea is to decompose the original image into two sub-images and to apply various processes to each sub-image in order to gain space and increase the amount of embedded data. The two sub-images are then scrambled and partially encrypted. The most significant characteristic of the proposed method is the utilization of a single procedure to simultaneously perform the compression, the reversible data hiding and the partial encryption rather than using three separate procedures. Our approach reduces then the computational effort and the required computation time. This method is specially suited for medical images where one can associate the patient diagnostic to the concerned medical image for safe transfer purpose.

**Keywords** fast and safe image transmission · lossless compression · reversible data hiding · partial encryption · image protection · real time image processing.

---

W. Puech · J.M. Rodrigues  
Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II, 161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE  
E-mail: [william.puech@lirmm.fr](mailto:william.puech@lirmm.fr), [jose-marconi.rodrigues@lirmm.fr](mailto:jose-marconi.rodrigues@lirmm.fr)

J.E. Develay-Morice  
Centre Hospitalier Universitaire de Montpellier-Nîmes, FRANCE  
E-mail: [jean-eric.develay-morice@wanadoo.fr](mailto:jean-eric.develay-morice@wanadoo.fr)

---

## 1 Introduction

A fast and safe transfer of data should combine compression and protection. The image protection problem can be solved by data hiding or encryption or both. In this paper we present a novel reversible method combining compression, data hiding and partial encryption of images in a single processing step.

The protection of binary information became mandatory in many fields such as medical, commercial and military. Basically, there are two levels of security: the full encryption which ensures the total data obscurity and the partial encryption which assumes a lower level security. The first type can be employed in the military and in the commercial domain, while the second one can be applied in the medical and educational fields, and for database searches. Several encryption procedures have been developed for image encryption [4, 10, 14, 17].

Data hiding has drawn extensive interests from the research community, but most of current data hiding techniques belong to lossy data hiding methods [5]. That means that after a lossy process, the data is different and for some applications, e.g. law enforcement and medicine, maintaining the fidelity of the original data is critical. The reversible data hiding is an approach that fully recovers the original image. Many reversible data embedding techniques have been developed [9, 7], but few ensure the reduction in the file size [6]. Recent work proposed a reversible data hiding algorithm which can embed more data than many of the existing reversible data hiding algorithms [13]. Moreover, none of them can perform simultaneously reversible data hiding, partial encryption and lossless compression. In this work we present a new reversible technique for fast and safe image transfer. Indeed, the fact to have a single procedure to simultaneously perform the compression, the reversible data hiding and the partial encryption rather than using three separate procedures, reduces the computational effort and the required computation time. This method is then suitable for real time applications and applicable for var-

ious processing device capabilities such as, for example, wireless devices.

This paper is organized as follows. Section 2 elaborates the method and its steps: decomposition, lossless compression, reversible data hiding and partial encryption as well as the decoding procedure. We present the results and analyses of the robustness in Section 3. We applied our method to more than 100 various images. Section 4 concludes the paper.

## 2 The proposed method

The main idea of the proposed method is to decompose the gray level original image into two sub-images, called Semi-Pixel Images (SPI(1-4) and SPI(5-8)), and to process each of them in order to get a protected image that carries hidden information. For each block of the SPI(5-8) we apply the compression and acquire space for data hiding. The proposed algorithm for compression generates gaps that are used to hide data. Sequentially, bits of the message are embedded in the blocks which are then scrambled with SPI(1-4) block and partially encrypted using a secret key. The processed final image is then ready to be transmitted. The steps of the method are illustrated in the diagram from Fig. 1.

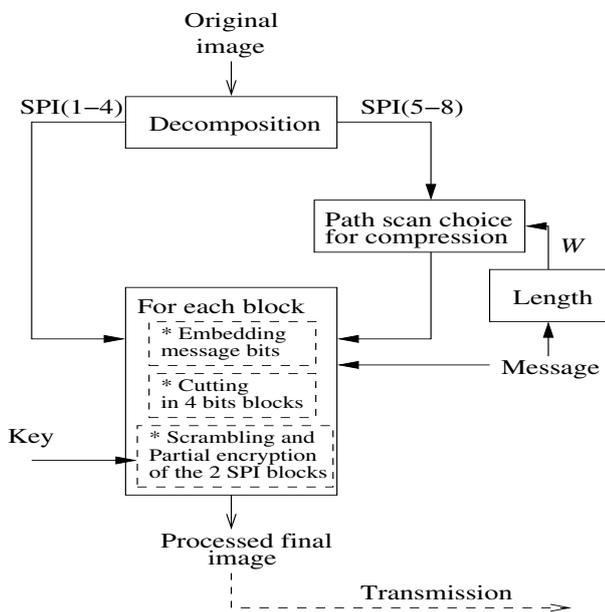


Fig. 1 Global method overview.

Before elaborating on the details of the method in the following sections, it is worthwhile to note some considerations. The compression applied to SPI(5-8) is based on the work of Maniccam and Bourbakis [11]. In their article, they decompose the image in eight binary images.

They show that the highest compression factor is accomplished in the four binary images corresponding to the most significant bits. The SPI(5-8) is thus more adapted for compression.

The most significant characteristic of the proposed method is the utilization of a single procedure to simultaneously perform the compression, the reversible data hiding and the partial encryption rather than using three separate procedures such as: LE4RLE [6] for data hiding, lossless JPEG2000 [2] for compression, and AES [3] for encryption. Our approach will substantially reduce the computational requirements because only binary and Boolean operators<sup>1</sup> are utilized. Thus, the method can be employed in low power systems such as in wireless devices.

Section 2 is organized as follows. Section 2.1 presents the original image decomposition. Section 2.2 details the lossless compression method. The reversible data hiding method is presented in Section 2.3 and the partial encryption procedure is detailed in Section 2.4. Section 2.5 presents statistic analysis of the security. Finally, the image decoding is presented in Section 2.6.

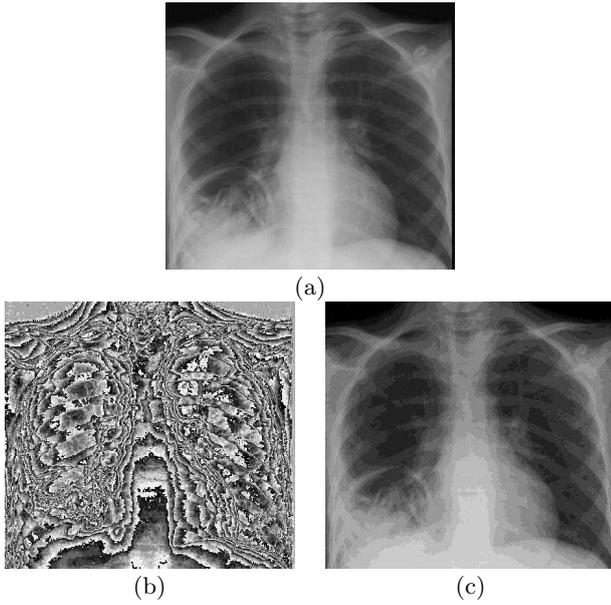
### 2.1 Decomposition

The first step is the decomposition of original image into two Semi-Pixel Images (SPI). This decomposition is very simple and fast because it is done by carrying out the left-shift and right-shift bit operations on the original 8 bit pixels. Thus, each pixel of the original image is divided into two semi-pixels of four bits each. The four most significant bits ( $5^{th} - 8^{th}$ ) constitute the SPI(5-8), and the four least significant bits ( $1^{st} - 4^{th}$ ) make up the SPI(1-4).

Fig. 2 shows an example of image decomposition. Fig. 2.a illustrates an original radiographic image which is a two-dimensional representation of the thorax. One can notice in Fig. 2.c, showing SPI(5-8), a large number of homogeneous areas. This property is favorable for achieving a good compression rate. We can also notice that the SPI(5-8) carries the most representative part of the original image while the SPI(1-4), shown in Fig. 2.b, carries the details.

Without this decomposition, the lossless compression, presented in the next section, cannot be efficient because of the heterogeneity of the images. We have chosen a static decomposition in order to reduce computation time and the coding cost. This static decomposition splits the image into two SPI of the same size because experimentally the most significant heterogeneity is in the four least significant bits making up the SPI(1-4).

<sup>1</sup> Such as shift-bit-left, shift-bit-right, AND, OR and XOR



**Fig. 2** a) Original image, b) SPI(1-4), c) SPI(5-8).

## 2.2 Lossless compression

### 2.2.1 Run length encoding for data hiding

In order to obtain space for data hiding without increasing the image size, it is necessary to compress the image. Thus, SPI(5-8) is compressed using a variation of the Run Length Encoding (RLE) algorithm [8]. RLE is a lossless compression that assigns short codes to long runs of identical symbols. RLE algorithm has been successfully used for images that contain homogeneous areas. RLE algorithm replaces a sequence of identical pixels by a block  $B$ . This block  $B$  is constituted by  $(f, q, C)$ , where  $f$  is a flag to identify the block,  $q$  is the quantity of repeated elements in the sequence and  $C$  is the pixel color.

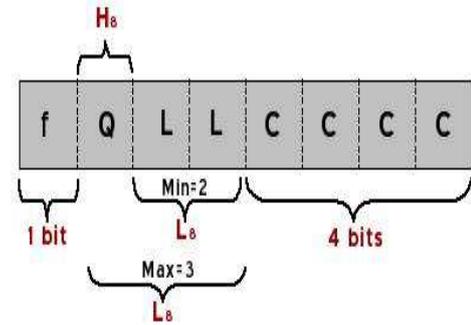
In this work, we propose a variation of the RLE algorithm: the RLE for Data Hiding (RLE4DH) which always reserves space for data hiding. In RLE4DH, the block  $B$  is constituted by  $(f, Q, C)$ , where  $Q(H, L)$  is function of the available space for data hiding  $H$  and the longest sequence length  $L$ .

In the RLE4DH we propose to build two types of blocks  $B$ : the blocks  $B_8$  with 8 bits and the blocks  $B_{16}$  with 16 bits. For these two types of blocks, the numbers of bits used for the flag  $f$  (1 bit) and for the pixel color  $C$  (4 bits) are static. However, the numbers of bits used for  $H$  and  $L$  are variable and they depend on each other: the larger the  $H$ , the smaller the  $L$ , as illustrated in Figs. 3 and 4. The lengths of  $H$  and  $L$  also depend on the block type ( $B_8$  or  $B_{16}$ ) and on the image content: the larger the homogeneous areas, the longer the  $L$ . The number

of bits used for data hiding  $H_8$  and  $H_{16}$  are given by the following equations:

$$\begin{cases} H_8 = \begin{cases} 1 & \text{if } 1 \leq L_8 \leq 4 \\ 3 - \lceil \log_2(L_8) \rceil & \text{if } 4 < L_8 \leq 8 \end{cases} \\ H_{16} = 11 - \lceil \log_2(L_{16}) \rceil & \text{if } 8 < L_{16} \leq 512. \end{cases} \quad (1)$$

	Value	Size in bits
f	0	1
H	hidden data	$0 \leq H_8 \leq 1$
L	$1 \leq L_8 \leq 8$	$2 \leq L_8 = 3 - H_8 \leq 3$
C	color	4
	Total	8



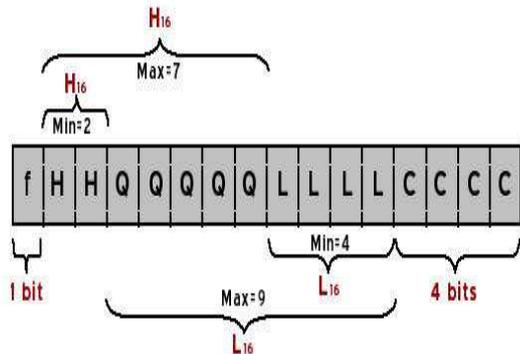
**Fig. 3** Constitution of the  $B_8$  block, 1 bit for flag, 3 bits for data hiding and sequence length and 4 bits for color.

Fig. 3 exhibits the tabular and graphical representation of the  $B_8$  blocks, where  $H_8$  is the number of bits used for data hiding,  $L_8$  is the number of bits necessary to represent the maximal length of the longest sequence (MLLS) and  $Q$  can be used for either  $H_8$  or  $L_8$ . We remark that the  $B_8$  block has at most one bit for data hiding.

Fig. 4 shows the disposition of the  $B_{16}$  blocks, where  $H_{16}$  is the number of bits used for data hiding,  $L_{16}$  is the number of bits necessary to represent the MLLS and  $Q$  can be used as either  $H_{16}$  or  $L_{16}$ . Thus, in the block  $B_{16}$  we have at least 2 bits for data embedding.

If the sequence length  $L$  can be represented with 3 bits (values from 1 to 8), we use a block  $B_8$ . If the sequence length is from 9 to 512, we use a block  $B_{16}$ . Thus, if SPI(5-8) has a significant quantity of homogeneous regions and if their lengths are very long, our algorithm will use many  $B_{16}$  blocks. On the contrary, if those homogeneous areas are relatively small, the algorithm opts for  $B_8$  blocks.

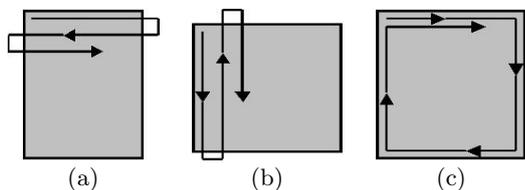
	Value	Size in bits
f	1	1
H	hidden data	$2 \leq H_{16} \leq 7$
L	$8 < L_{16} \leq 512$	$4 \leq L_{16} = 11 - H_{16} \leq 9$
C	color	4
	Total	16



**Fig. 4** The tabular and graphical representation of  $B_{16}$ , 1 bit for flag, 11 bits for data hiding and sequence length and 4 bits for color.

### 2.2.2 SPI(5-8) scanning

In the compression procedure, we scan SPI(5-8) in three distinct paths as illustrated in Fig. 5, looking for sequences of identical pixel colors. The selected path to scan the SPI(5-8) will determine the quantity of blocks  $B_8$  and  $B_{16}$ . Those paths will influence the magnitude and the number of homogeneous regions. There are many scan paths and Maniccam and Bourbakis [12] present an approach for compressing using various scan paths. In their work, the image is divided in square blocks of  $4 \times 4$ ,  $8 \times 8$  or  $16 \times 16$  pixels for example, and then each block is scanned by 32 different scan patterns. The Maniccam and Bourbakis compression method is efficient for compressing purpose but it requires a long processing time because of the 32 scans and various block sizes.



**Fig. 5** Three basic paths for scanning SPI(5-8): a) By row, b) By column, c) By spiral.

The main goal of our work is to decrease the computation time. We propose then to find a good compromise between the compression rate and the computation time

by using only the three basic scan paths exhibited in Fig. 5. Indeed, the three selected paths are among the more standard to scan pixels in images. To decrease the computation time, for each path, we chose to start with the first pixel of the first row.

### 2.2.3 Maximal Length of the Longest Sequence (MLLS)

After scanning the image and finding the values of  $L_8$  and  $L_{16}$  for each path, we need to compute the *MLLS* for  $B_8$  and  $B_{16}$ . The two values of the *MLLS* specify the quantity of bits,  $H_8$  and  $H_{16}$ , that will be employed for data hiding as described by equation (1). The *MLLS* is dynamically determined for the two kinds of blocks. It is a function of both the image size and the space  $W$  allocated for data hiding.  $W$  is the required space, measured in bits, to embed the data in the image and is determined by:

$$W = n_8 H_8 + n_{16} H_{16} + 8\Delta, \quad (2)$$

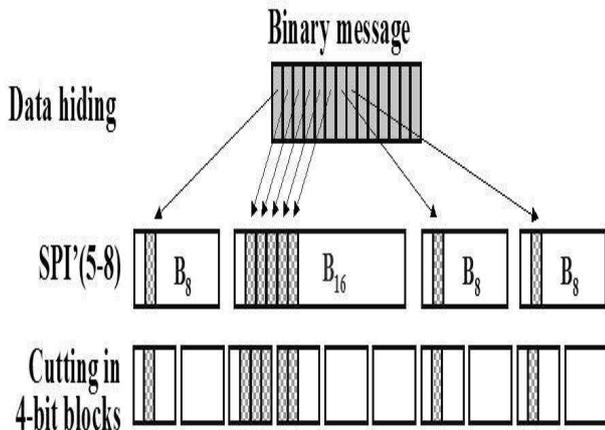
where  $n_i$  is the number of blocks of type  $B_i$  and  $\Delta$  is the number of pixels necessary to pad the last row of the final image. Indeed, since we have chosen to preserve the original image width, the end of the last row of the final compressed image can be used in order to increase the space  $W$  allocated for data hiding.

The algorithm works as follows. It scans the image considering that the longest sequence length  $L_8$  for the block  $B_8$  can be 4 or 8. It is worthwhile to explain that we use 4 because it is the longest sequence length (1 to 4) that can be stored in two bits. Consequently, the value 8 is the longest sequence length that can be saved in three bits (1 to 8). The same process is employed for  $B_{16}$  using the set of values  $\{16, 32, 64, 128, 256, 512\}$ . The reasoning used for the  $B_8$  blocks can also be applied for the  $B_{16}$ . The maximum value is 512, because it is the largest value that can be represented by nine bits (1 to 512). After this computation, the algorithm defines the *MLLS* for embedding the message according to the size of  $W$  and to the maximum achievable image compression rate.

## 2.3 Reversible data hiding

During the compression procedure we can, at the same time, embed the message. Indeed for each block  $B_i$ , we know the space magnitude  $H_i$  (space used for data hiding) we can use.

The embedding process is done bit by bit. The message is embedded in the bits  $H_8$  of  $B_8$  and  $H_{16}$  of  $B_{16}$  in the SPI(5-8). Fig. 6 shows an example of the embedding process. In this case, for the data hiding purpose  $H_8 = 1$  and  $H_{16} = 5$ . One can notice that the message is spread over SPI(5-8). It will depend on the disposition of the blocks  $B_8$  and  $B_{16}$  and these blocks depend on the image content (homogeneous/heterogeneous zones).



**Fig. 6** Data hiding diagram and SPI'(5-8) cutting.

The message that is embedded in SPI(5-8) can be any binary data like image, text or sound. After the data hiding process, let SPI(5-8) becomes SPI'(5-8). SPI'(5-8) is sliced into groups of four bits, as illustrated in Fig. 6, which are used in the partial encryption procedure.

## 2.4 Partial Encryption

The proposed process is based on three concomitant steps which are the scrambling, the concatenation and the partial encryption. The aim of the scrambling and concatenation steps is to group the semi-pixel images SPI(1-4) and SPI'(5-8), while that of the partial encryption is to ensure the secrecy.

Despite the data hiding procedure of spreading the message over the SPI'(5-8), ensuring the secrecy, it cannot be considered as an encryption method. An encryption method which depends only on the secrecy of the encryption process is not truly an encryption method [15]. Our proposed partial ciphering method uses a key with standard length of 128 bits which is suitable for a large spectrum of privacy tasks. But, a key length of at least 80 bits is required for the protection against the brute force attacks in a symmetric encryption. In reality, in our method, the size of the key can be decided by the level of the required confidentiality. Even if we know that our applications do not require a full encryption process, in order to increase the security of our proposed system we use some of the AES (Advanced Encryption Standard [1]) operations such as Shift-Rows() and Mix-Columns().

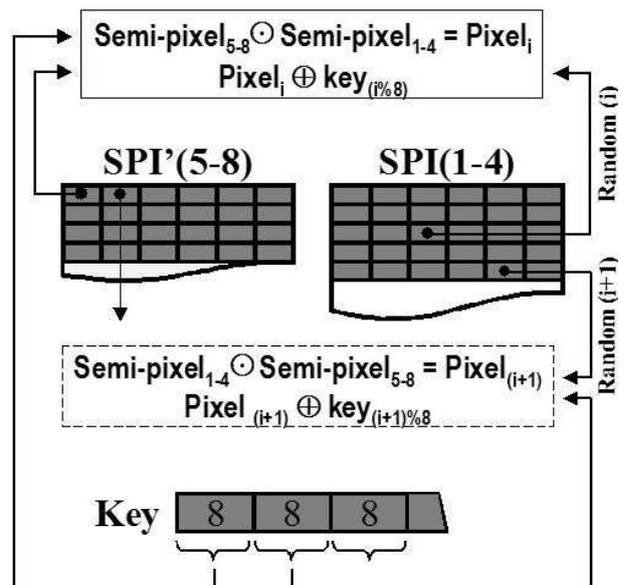
The key has an important role in our partial encryption procedure. It is divided in sub-keys of eight bits each. Then, these sub-keys are XOR-ed with the pixels obtained from the concatenation of SPI'(5-8) and SPI(1-4). The key is further used as a seed of a pseudo-random

number generator (PRNG). This PRNG produces numbers that will be employed in three points of the process:

- to locate the critical data in the final image for the purpose of extraction,
- to determine the semi-pixel in SPI(1-4),
- to decide the inversion type such as:
  - Odd:  $SPI(5-8) \odot SPI(1-4)$ .
  - Even:  $SPI(1-4) \odot SPI(5-8)$ , where  $\odot$  is the concatenation operator.

The first random number generated by the PRNG is used to embed the critical data which is inserted in nine extra bytes (72 bits) and is made up of:

- 1 bit to specify the  $H_8$  size.
- 3 bits to specify the  $H_{16}$  size.
- 2 bits for scan mode (row, column, spiral).
- 16 bits for the number of columns of the original image.
- 16 bits for the number of rows of the original image.
- 16-bits for the size of embedded information.
- 18 bits for the SPI'(5-8) size.



**Fig. 7** Partial encryption process, with  $\odot$  the concatenation operator and  $\oplus$  the XOR operation.

The partial ciphering process works as follows. It generates random numbers that are used to get semi-pixels in SPI(1-4). Then, if the random number is odd, the concatenation is made in the order  $SPI(5-8) \odot SPI(1-4)$ , else the order is  $SPI(1-4) \odot SPI(5-8)$ , where  $\odot$  is the concatenation operator. Even if the security level of the proposed method is lower than the AES security level, we precise that the algorithm AES also scrambles data (Shift-Rows(), Mix-Columns()) to hinder unauthorized

extraction [1]. Then, the proposed partial encryption algorithm uses a part of the key to make an XOR operation with the pixel previously generated by the concatenation. The proposed partial encryption process is presented in Fig. 7. Since the size of SPI(5-8) is smaller than the size of SPI(1-4), this process is performed until the end of SPI(5-8). The remaining semi-pixels of the SPI(1-4) will be concatenated with others randomly chosen among those remaining SPI(1-4) semi-pixels.

## 2.5 Security level analysis

In this section we present how to get a certain security level by measuring the variations in the protected images. We can first measure the image information content with the entropy  $H(S)$ . If an image has  $N$  gray levels  $\alpha_i$ , with  $0 \leq i < N$ , and the probability of gray level  $\alpha_i$  is  $P(\alpha_i)$ , the entropy  $H(S)$ , without considering the correlation of gray levels, is defined as:

$$H(S) = - \sum_{i=0}^{N-1} P(\alpha_i) \log_2(P(\alpha_i)). \quad (3)$$

Information redundancy  $r$  is defined as:

$$r = b - H(S), \quad (4)$$

where  $b$  is the smallest number of bits with which the image quantization levels can be represented. In theory, the robustness against statistical attacks is maximal when the probability of each gray level is  $P(\alpha_i) = \frac{1}{N}$ . In this case  $H(S) = \log_2(N)$  bits/pixel. The security level is acceptable when  $r \simeq 0$ . However, an image is a two-dimensional information and the pixels  $p(j)$  are correlated among themselves. Theoretically, an image is an order- $M$  Markov source, with  $M$  the image size. In order to reduce the complexity, the image is cut in small block of size  $n$  and considered as an order- $n$  Markov source. The alphabet of the order- $n$  Markov source, called  $S'$ , is  $\beta_i$  with  $0 \leq i < N^n$  and the order- $n$  entropy  $H(S')$  is defined as:

$$H(S') = H(S^n) = - \sum_{i=0}^{N^n-1} P(\beta_i) \log_2(P(\beta_i)). \quad (5)$$

For our experiments we used  $N = 256$  gray levels and blocks of  $n = 2$  or  $3$  pixels corresponding to a pixel and its preceding neighbors. Consequently, to get information redundancy  $r \simeq 0$ , equation (4), we should have  $b = 8$  bits/pixel for equation (3) and  $b = 16$  or  $24$  bits/block for equation (5). We also analyzed the variation of the local standard deviation  $\sigma$  for each pixel  $p(j)$  taking account of its neighbors to calculate the local mean  $\bar{p}(j)$ :

$$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^m (p(j) - \bar{p}(j))^2}, \quad (6)$$

with  $m$  the size of the pixel block to calculate the local mean and standard deviation, and  $0 \leq j < M$ , if  $M$  is the image size.

## 2.6 The decoding process

The decoding of the proposed method works as follows. It uses the secret-key as the seed for the PRNG. The first generated number indicates the localization of the critical data necessary for the decoding. The decoding process is then composed of the decryption, the extraction of the data and the reconstruction of the original image without any loss.

## 3 Experimental results

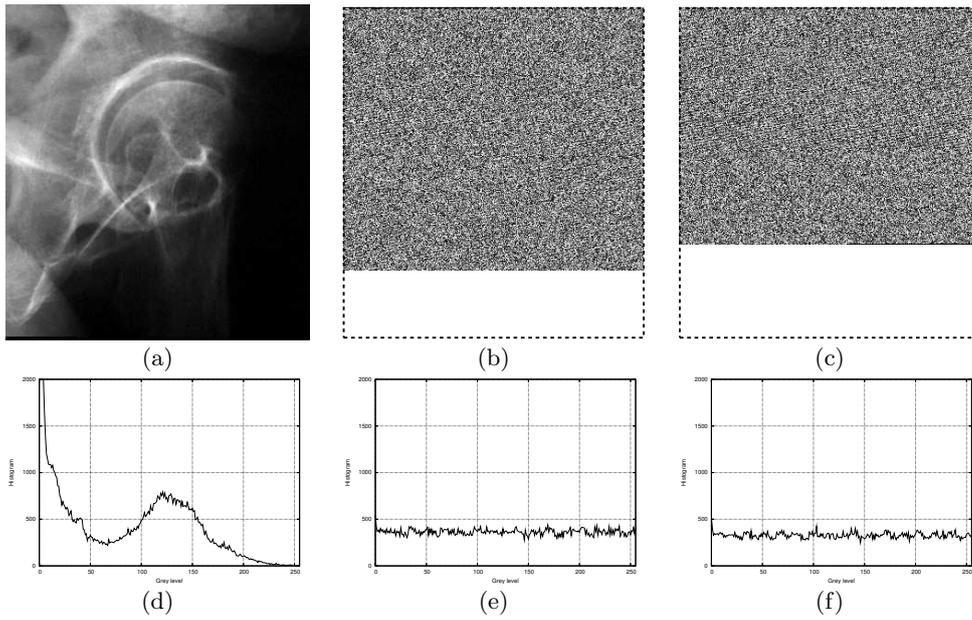
We applied our proposed method to more than 100 various images. In this section, we explain the results obtained for two of these images: a radiographic image showing an osteoarthritis of the hip, Fig. 8.a, and the image of Lena, Fig.10.a. They were ciphered with a 128-bit key and fully filled with binary messages. The entire space reserved for data hiding was filled by the binary message.

Fig. 8.b shows the result of our method applied to the medical image with a maximal space ( $W$ ) for embedding purpose. A message of 9544 bytes was embedded in the image. This message size is enough to embed the complete medical history of the patient. Although we have inserted a large quantity of information, we have accomplished a compression rate of  $\tau = 1.26$ .

Fig. 8.c presents the result of our method applied to the medical image with a maximal compression. We have embedded a binary message of 890 bytes which corresponds to a medical condition summary of the patient. The size of the final image is 84012 bytes. We have then accomplished a compression rate of  $\tau = 1.40$ . Table 3 presents the results for the two images from Figs. 8.a and 10.a. For the image shown in Fig. 8.b the MLLS sequence for block  $B_8$  is 4 and for block  $B_{16}$  is 16. For the image Fig. 8.c these values are 8 and 512 respectively.

We also show the histograms for the above-cited examples. The original one, Fig. 8.d and the final images, Figs. 8.e and 8.f. The latter two histograms are flat, and from equation (3) we get very high entropy  $H(S)$  of 7.99 bits/pixel ( $H(S) = 7.21$  bits/pixel for the original image). The information redundancy  $r$ , equation (4), is then equal to 0.01 bit/pixel. The order-2 entropy,  $H(S^2)$ , of equation (5), is equal to 15.40 bits/block for Fig. 8.b and 15.33 bits/block for Fig. 8.c (11.09 bits/block for the original image). The information redundancy  $r$ , equation (4), is then less than 0.67 bit/block. The order-3 entropy,  $H(S^3)$ , of equation (5), is equal to 22.92 bits/block for Fig. 8.b and 22.72 bits/block for Fig. 8.c (15.03 bits/block for the original image). The information redundancy  $r$ , equation (4), is then less than 1.28 bit/block.

From equation (6) we also analyzed the variation of the local standard deviation  $\sigma$  for each pixel while taking its neighbors into account. The mean local standard deviation is equal to 69.27 gray levels for the final image



**Fig. 8** a) Original radiographic image showing an osteoarthritis of the hip ( $361 \times 325$  pixels), b) Final image ( $287 \times 325$  pixels) with maximal  $W$ , c) Final image ( $258 \times 325$  pixels) with maximal compression, d) Original image histogram, e) Histogram of image (b), f) Histogram of image (c).

Image	Radiography				Lena			
Original image size (bytes)	117325				262144			
Maximal Scan type	Size of $W$		Compression		Size of $W$		Compression	
	row		col.		row		col.	
$H_8$   $H_{16}$	1	7	0	2	1	7	0	4
$n_8$	18707		20374		74884		65881	
$n_{16}$	7878		2335		14885		6140	
Final image size (bytes)	93437		84012		236032		209408	
Compression $\tau$	1.26		1.40		1.11		1.25	
$W$ (bytes)	9544		890		22690		3245	
$W$ percentage	8.13 %		0.76 %		8.66 %		1.24 %	
entropy (bits/pixel)	7.99		7.99		7.99		7.99	
Order-2 entropy (bits/block)	15.40		15.33		15.78		15.75	
Order-3 entropy (bits/block)	22.92		22.72		23.67		23.43	

**Table 1** Processing summary.

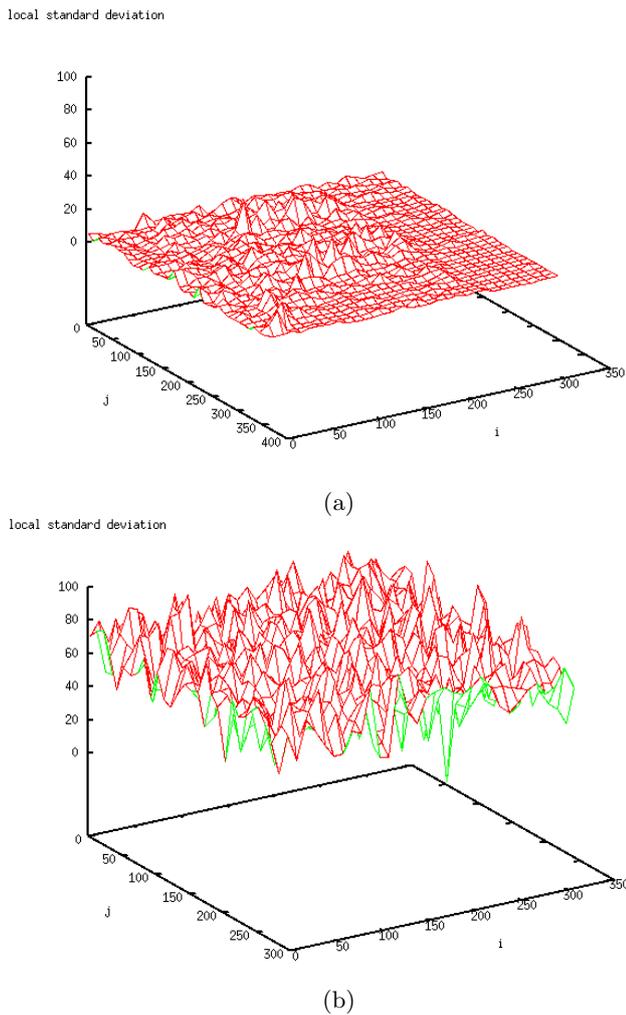
Method	compression rate	payload (bytes)	encryption	time (s)
Maniccam and Bourbakis [11]	1.64	$\emptyset$	yes	54
Ni <i>et al.</i> [13]	1	678	no	0.1
Proposed Fig. 10.b	1.11	22690	yes	0.33
Proposed Fig. 10.c	1.25	3245	yes	0.33

**Table 2** Comparison between other reversible methods and our proposed method on Lena image. Note that Maniccam and Bourbakis method does not embed data and Ni *et al.* method does not compress the image.

illustrated Fig. 8.b, and to 68.94 gray levels for the final image illustrated Fig. 8.c. The mean local standard deviation is equal to 3.79 gray levels for the original medical image Fig. 8.a. Figs. 9.a and b illustrate the local standard deviation of the original medical image and of the final image of Figure 8.b. These analyzes show that the two final images, Figs. 8.b and c, are protected against

statistical attacks. The information redundancy is very small and thus statistical attacks would be difficult [16].

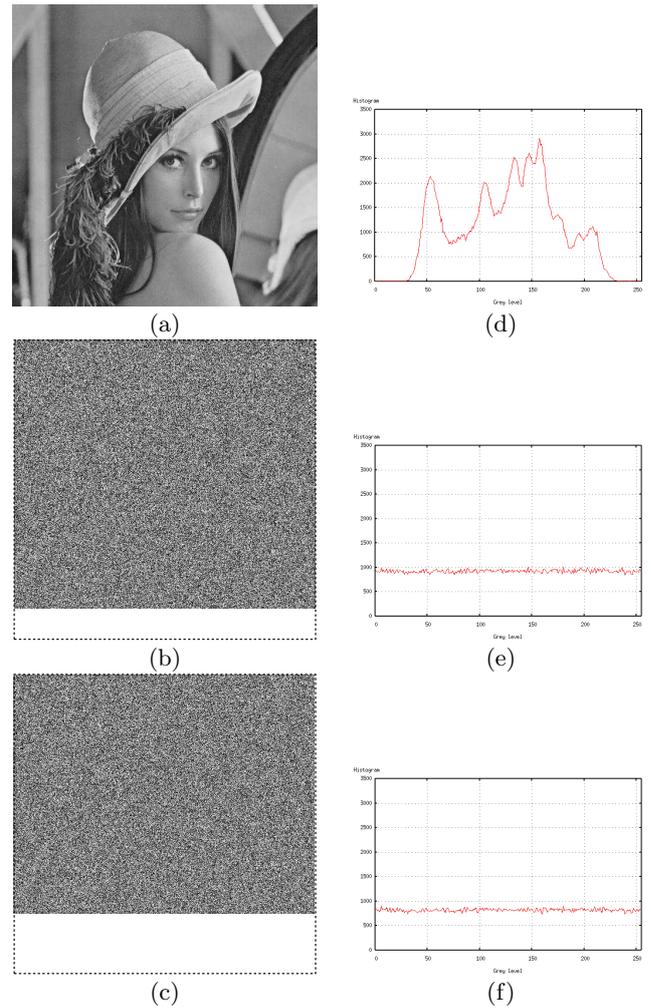
The same analyzes have been made on the Lena image presented in Fig. 10.a. The maximal embedding space for the image from Fig. 10.b, is 22690 bytes. It corresponds to 8.66 % of the original image size. Fig. 10.c carries a message of 3245 bytes and its compression rate



**Fig. 9** Local standard deviation: a) Original medical image of Figure 8.a, b) final image of Figure 8.b.

is 1.25. From Table 3, we can notice that the embedding capacity of the proposed method depends strongly on the chosen path and the characteristics of the original image. The values of  $W$  give upper bounds of message sizes that can be embedded. Comparison results between other reversible methods and our proposed method on Lena image are presented in Table 2. Maniccam and Bourbakis method [11] encrypts and compress during the same process, the compression is very good but the computation time is very long and this method does not embed hidden data in the final image. Ni *et al.* [13] method does not compress the image and the final image is not encrypted with a very good PSNR (48.2 dB) before the reversible extraction of the data. From these comparisons, it is observed that our proposed technique has achieved a good compromise between compression rate, the payload, the partial encryption and the computation time.

Table 3 presents the average value results for 100 various images. We achieve to hide in the image a maximum of information corresponding to 8.22% of the original

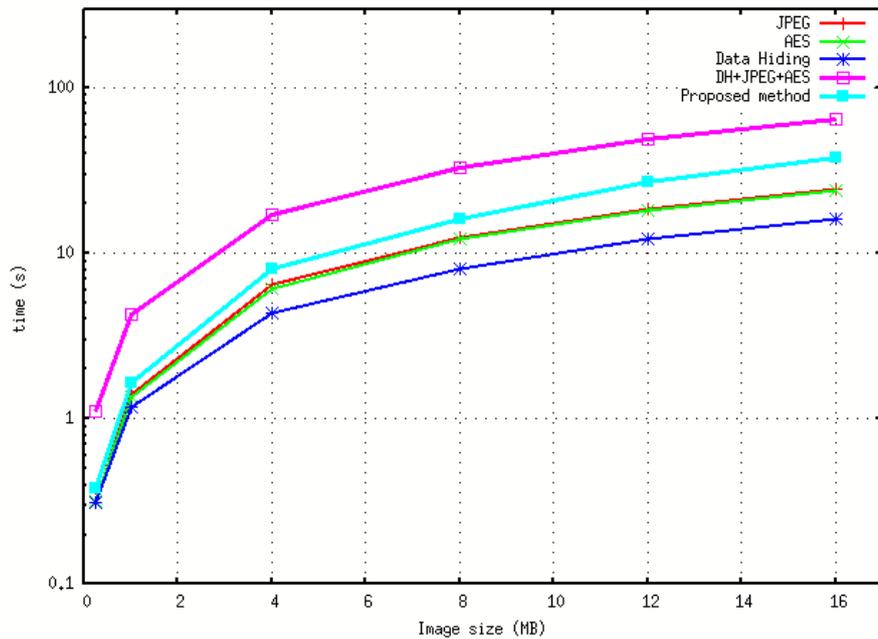


**Fig. 10** a) Original image Lena ( $512 \times 512$  pixels), b) Final image ( $461 \times 512$  pixels) with maximal  $W$ , c) Final image ( $409 \times 512$  pixels) with maximal compression, d) Original image histogram, e) Histogram of image (b), f) Histogram of image (c).

		Capacity %	Compression rate	Entropy bits/pixel bits/block
Maximal $W$	Mean	8.22	1.30	7.99
	$\sigma$	0.99	0.08	15.58
Maximal Compression	Mean	1.21	1.51	7.99
	$\sigma$	0.56	0.09	15.49

**Table 3** Mean value and standard deviation for 100 various images.

image size. Despite this embedding capacity our method yields a compression rate of 1.30. While maximizing the compression, we have got an average compression rate of 1.51 and a data hiding capacity of 1.21 % of the original image size. For these two cases the entropy is similar, about 7.99 bits/pixel and 15.5 bits/block for the order-2



**Fig. 11** Computation time (s) as a function of the image size (MB): pink curve for the use of three separate algorithms and cyan curve for our proposed method.

entropy. We can remark, with the standard deviations from Table 3, that the variations are very small.

The use of three separate algorithms (lossless data hiding, lossless compression and encryption) is not suitable for several reasons. For example, we propose to compare the proposed method with the use of lossless JPEG2000 as standard image compressor, RL4RLE [6] as lossless data hiding method and AES as standard cipher. Application of the RL4RLE for data hiding and JPEG2000 in the lossless mode is not a good proposition. Indeed, the RL4RLE compresses and embeds information, but the gain in the compression rate is not justifiable given the high JPEG2000 processing time. In addition, if the JPEG2000 compressor is applied first, the RL4RLE will change the JPEG2000 image format. Furthermore, the application of three separate procedures increases the computation time which is not suitable for real time applications. We present in Fig. 11 the comparison of the computation time as a function of the image size (MB) between the use of three separate algorithms and our proposed method<sup>2</sup>. Because of the SPI(5-8) scanning, we concede that our method is slower as compared to a single standard processing such as JPEG2000. But it must be noted that, to hide data, compress and partially encrypt the image, our method is faster as a whole than three standard methods separately applied. For example, in the case of transfer of medical images, the size of medical images is generally between 1 and 4 MB and the data transmission must be fast. From Fig. 11 we

can remark that computation time is divided by two if you compare with a standard approach based on three separate algorithms. For example for a medical image of 2 MB we need only 4 s instead of 8 s to process the image. Thus, for this kind of applications and image sizes, the proposed method is very interesting because the time processing is twice smaller and is lower than 10 s, which is minimum time to analyze a medical image before transmitting a new image.

## 4 Conclusion

In this paper, a fast reversible method for protecting an image transmission was presented. This approach is founded on image decomposition, RLE based-content compression, data hiding and partial encryption.

The compression rate versus space for data hiding is dynamic and according to the message size to embed. Based on the average values for 100 studied various images, we can conclude that the proposed method is able to hide as much information as around 8% of the original image size. In this case, it can reach a compression rate of about 1.3. For these 100 studied various images, for maximal compression rate, it accomplishes an embedding capacity of the order of 1.2% of the original image size and an average compression rate of 1.5. We have achieved a maximal entropy of near 8 bits/pixel and a order-2 entropy of near 15.5 bits/block. This approach offers a very efficient partial ciphering algorithm and the complete recovery of the original image.

<sup>2</sup> We have used a 1.5 GHz Intel Pentium PC.

In our method, the three processes (lossless compression, data hiding and partial encryption) are performed in a single procedure. Therefore, it will shrink the computational effort and the required computation time. Consequently, it is suitable for real time applications and applicable for various processing device capabilities such as, for example, wireless devices.

---

## References

1. AES. Announcing the Advanced Encryption Standard. *Federal Information Processing Standards Publication*, 2001.
2. C. Charilaos, S. Athanassios, and E. Touradj. The JPEG2000 Still Image Coding System An Overview. *IEEE Trans. on Consumer Electronics*, 46(4):1103–1127, Nov. 2000.
3. J. Daemen and V. Rijmen. *The Design of Rijndael*. SpringerVerlag New York, Inc. Secaucus, NJ, USA, 2002.
4. M. Van Droogenbroeck and R. Benedett. Techniques for a Selective Encryption of Uncompressed and Compressed Images. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, pages 90–97, Sept. 2002.
5. J. Fridrich. Applications of Data Hiding in Digital Images. In *ISPACS'98 Conference*, 1998.
6. J. Fridrich, M. Goljan, Q. Chen, and V. Pathak. Lossless Data Embedding with File Size Preservation. In *Proc. EI SPIE San Jose, CA*, Jan 2004.
7. J. Fridrich, M. Goljan, and R. Du. Lossless Data Embedding New Paradigm in Digital Watermarking. *EURASIP Journal Applications on Signal Processing*, 2002:185–196, Feb 2002.
8. R. C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Pearson Education (2002), Elsevier, 2002.
9. C.W. Honsinger, P.W. Jones, M. Rabbani, and J.C. Stoffel. Lossless Recovery of an Original Image Containing Embedded Data. *US Pat. 6,278,791*, 2001.
10. X. Liu and A. Eskicioglu. Selective Encryption of Multimedia Content in Distribution Networks: Challenges and New Directions. In *IASTED Communications, Internet & Information Technology (CIIT), USA*, November, 2003.
11. S.S. Maniccam and N.G. Bourbakis. Lossless image compression and encryption using SCAN. *Pattern Recognition*, 34:1229–1245, 2001.
12. S.S. Maniccam and N.G. Bourbakis. Lossless Compression and Information Hiding in Images. *Pattern Recognition*, 37:475–486, 2004.
13. Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su. Reversible Data Hiding. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(3):354–362, Mar. 2006.
14. R. Norcen, M. Podesser, A. Pommer, H.P. Schmidt, and A. Uhl. Confidential Storage and Transmission of Medical Image Data. *Computers in Biology and Medicine*, 33:277–292, 2003.
15. B. Schneier. *Applied cryptography*. Wiley, New-York, USA, 1995.
16. D. R. Stinson. *Cryptography: Theory and Practice, (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC Press, New York, November 2005.
17. A. Uhl and A. Pommer. *Image and Video Encryption: From Digital Rights Management to Secured Personal Communication*. Springer, 2005.

**W. Puech** was born in December 1967, in France. He received the diploma of Electrical Engineering from the University of Montpellier, France, in 1991 and the Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France in 1997. He started his research activities in image processing and computer vision. He served as a Visiting Research Associate to the University of Thessaloniki, Greece. From 1997 to 2000, he had been an Assistant Professor in the University of Toulon, France, with research interests including methods of active contours applied to medical images sequences. Since 2000, he is an Associate Professor of the University of Montpellier, France. He works now in the LIRMM Laboratory (Laboratory of Computer Science, Robotic and Microelectronic of Montpellier). His current interests are in the areas of security of digital image transfer (watermarking, data hiding, compression and cryptography) and edges detection applied to medical images and road security.

**José M. Rodrigues** was born in Brazil, in 1959. He received the diploma of Civil Engineering from Federal University of Ceara (UFC), Brazil in 1983. He is an expert adviser to a lot of computer companies and a council member of universities in Brazil. He got his diploma of database system designer from OIC, Japan in 1992. He received the M.S degree in Computer Science from UFC, Brazil in 2002 and the Ph.D. degree in Information, Structures and Systems in Computer Science from the University of Montpellier II, France in 2006. Nowadays, he is an engineer of Information Technology at the UFC.

**J.-E. Develay-Morice** was born in 1959, in France. He received the diploma of doctor in medicine from the Medicine University of Montpellier. In 1991, he received his Ph.D. Degree with the title "Echographie de l'épaule" and a diploma of general echography. He is a doctor referent in obstetric echography for Montpellier and Nîmes hospital since their creation. He is a hospital expert attache with the maternity of the CHU of Montpellier and the maternity of the CHU of Nîmes and has a private activity within a group of imagery in Valmédica, in Nîmes. It has a strong implication in the remote echographic expertise, begun with the project Maternet, project joining together several hospitals and private clinics having already made it possible to appreciably improve the efficacy of tracking, with moreover one greater comfort for the patients. He was an in particular doctor "adviser technical" in the development of the SonoPC project, with the participation of the french ministry for industry, having led to the realization of a controllable portable echograph remotely by Internet with several European demonstrations.