https://doi.org/10.2352/ISSN.2470-1173.2017.7.MWSF-319

High-Capacity Reversible Data Hiding in Encrypted Images using MSB Prediction

Pauline Puteaux and William Puech; LIRMM Laboratory UMR 5506 CNRS, University of Montpellier; Montpellier, France

Abstract

With the development of cloud computing, data privacy has become a major problem. Reversible data hiding in encrypted images (RDHEI) is an effective technique to embed data in the encrypted domain. Indeed, a lot of methods have been proposed, but none allows a large amount of embedding capacity with a perfect reversibility. In this work, we present a new method of reversible data hiding in encrypted images using MSB (most significant bit) prediction. In order to reconstruct the original image without any errors during the decryption phase, we adapt the to-be-inserted message. Some of the pixels' MSB values are used to highlight the prediction errors and the remaining values are replaced by bits of the secret message. Results show that it is still possible to embed a large message (payload close to 1 bpp).

Introduction

In the last few years, the growth in information technology, especially in computer networks – and in particular, Internet – led to serious security problems such as hacking, copy or malicious usage of information. For the purpose to insure a secure transmission of multimedia content through public communication channel, two major techniques have been developed: data hiding and encryption.

Reversible data hiding (RDH) is a technique to conceal secret data in a signal (*e.g.* an image). After its extraction, the original image has to be losslessly recovered in order to satisfy some requirements of strict areas, like the military or the medical world, where distortion of the image may have a critical impact.

In 2003, Tian proposed his method of difference expansion data hiding [9]: the redundancy in digital images is explored. He calculated all the differences of two adjacent pixels and selected some of these values to define the difference expansion (DE) and to embed additional data. Zhang *et al.* suggested exploiting the set of modification direction for a pixel (EMD) [15]. Moreover, methods based on histogram modification have also been described. Some proposed to build and to exploit the histogram according to the grayscale values [5] and others by using statistical data [10]. In [5], Ni *et al.* calculated the occurrences of all pixel values in the cover image to generate the histogram. All pixels between the peak and the zero points are modified during the data hiding phase. Pixels in the peak point are selected to hide the secret message. In [10], Tsai *et al.* embedded the secret message in the residual images' histogram instead of the original image one. Thereafter, a lot of new schemes, based on prediction error analysis (PE) and their expansion (PEE) were proposed [3, 7, 8]. These methods achieve a better performance in comparison with the previous ones. Thodi and Rodriguez were the first to describe a PEE-based method [8]. The difference between the pixel and its prediction is expanded for data embedding.

Otherwise, for data privacy, it is sometimes necessary to make an image unreadable. For this reason, a lot of encryption methods exist: security is ensured by scrambling, partially or completely, the information using a secret key. Two groups of techniques can be identified according to the use of a block cipher or a stream cipher. In the second group, algorithms based on chaos have been designed [1, 2, 11].

Reversible data hiding in encrypted image (RDHEI) is an effective technique to embed data in the encrypted domain without knowing the original content of the image. After the extraction of the message, it must be possible to reconstruct the original image with a minimum of errors – or preferably none at all – using the encryption key. The challenge lies in finding the best trade-off between the embedding capacity (in bpp) and the reconstructed image quality (in terms of PSNR or SSIM). Methods were also proposed to overcome this problem. Some suggested vacating room to embed data after the encryption phase (VRAE), others reserving room before image encryption (RRBE). In addition, encryption and data hiding can be joint, when data extraction and image reconstruction occur at the same time, or separate.

In previous work, Puech *et al.* proposed to analyze the local standard deviation of the marked encrypted image in order to reconstruct the original one without any errors during the decryption step [6]. The embedding rate was 1 bit for 16 pixels. First, the image is encrypted by using AES. After that, one bit of the message is embedded in each block, at a randomly selected position. Zhang [14] suggested to compress a part of the encrypted image and to use the free space to conceal secret data. This is a separate method because it is possible to extract the message independently of the image decryption. The first RRBE technique was proposed by Ma *et al.* [4]. They used a histogram shifting method on the clear image to release space. After the encryption step, they replaced some LSB values by bits of a secret message. Zhang *et al.* analyzed the prediction errors (PE) of some pixels and used the PE-histogram shifting technique before image encryption [13]. In their paper [12], Wu and Sun described two schemes. The first one is joint. They encrypted the image in the same way as Zhang in [14]. According to a data hiding key, some pixels of the encrypted image are selected to embed data and some space is released by applying the histogram shifting method. The second technique is separative. The to-be-inserted bits were hidden by MSB substitution. During the decoding phase, the data hiding key serves to extract the secret data and the original image was reconstructed thanks to a median filter on the watermarked image.

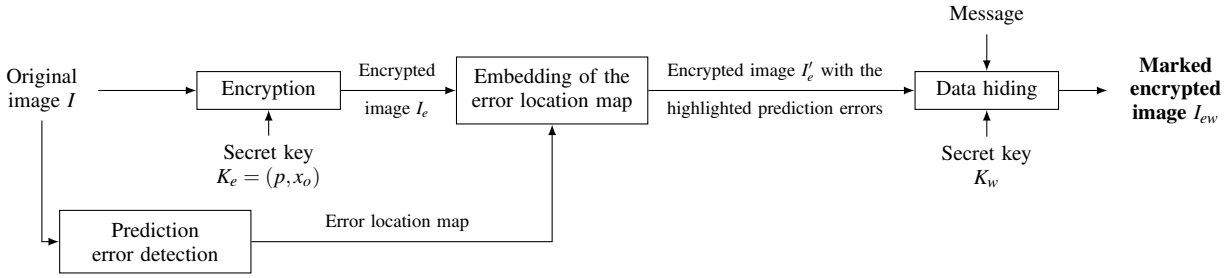


Figure 1: Overview of the encoding method.

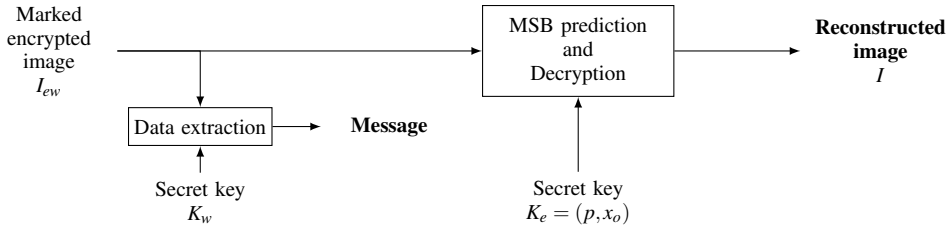


Figure 2: Overview of the decoding method.

To date, none of the existing methods succeeds in combining high payload embedding and high visual quality. Indeed, some of them are considered as reversible though PSNR is not equal to ∞. In [4], the payload can be high (0.5 bpp), but the reconstructed image is altered when compared with the original one (PSNR $\approx$ 40 dB). Moreover, other methods, such as Wu and Sun, propose a "high" embedding capacity, but it is only possible to embed 0.1563 bit per pixel at most [12].

In most cases, in the methods based on prediction error analysis (PE) or using a histogram shifting technique, the LSB (least significant bit) values of some pixels are replaced to hide bits of the secret message. However, if an image is encrypted, it is difficult to detect if it has a watermark or not. In fact, the pixel values of an encrypted image are pseudo-randomly generated. So, there is no correlation between a pixel and its adjacent neighbors. For this reason, we propose to watermark the MSB values instead of the LSB values. With this approach, in the encrypted domain, confidentiality is still the same and, during the decryption, the prediction of their values is easier to obtain than those of the LSB.

In this paper, we introduce a new reversible data hiding method for encrypted images based on MSB prediction with a very high capacity. In order to avoid the prediction errors, we adapt the to-be-inserted message to highlight the problematic pixels without significantly reducing the embedding capacity.

The remainder of this paper is organized as follows. Section 2 describes the proposed method. Experiment results are provided in Section 3. And finally, the conclusion is drawn and the future work is discussed in Section 4.

Proposed method

In this section, we introduce our proposed separate reversible data hiding method in encrypted images. The encoding phase consists of four steps: the prediction error detection, the encryption, the embedding of the error location map and the reversible data hiding by MSB substitution, as shown in Fig. 1. For the decoding phase, there are three possible outcomes. If the recipient has just the encryption key, they can only obtain the original image, but not the embedded message. On the contrary, if they only have the watermarking key, they can just extract the message. Obviously, when they are in possession of both the encryption and the watermarking keys, the recipient can extract the secret message and reconstruct the original image. The overview of this decoding method is presented in Fig. 2.

Prediction error detection

The first step consists of analyzing the original image content in order to detect the prediction errors:

- Consider $p(i,j)$ and the inverse value of $p(i,j)$, which is $inv(i,j) = (p(i,j) + 128) \bmod 256$. Note that there is a difference equal to 128 between these two values.
- Calculate the absolute difference between each of these two values with $p(i,j-1)$ and with $p(i-1,j)$. The smallest value gives the pixel which will be considered to predict $p(i,j)$ during the decoding step. Record these differences as Δ and Δ^{inv}:

$$\begin{cases} \Delta = \min(|p(i,j) - p(i,j-1)|, |p(i,j) - p(i-1,j)|) \\ \Delta^{inv} = \min(|inv(i,j) - p(i,j-1)|, |inv(i,j) - p(i-1,j)|) \end{cases} \quad (1)$$

- Compare the values of Δ and Δ^{inv}. If $\Delta < \Delta^{inv}$, there is no prediction error because the original value of $p(i,j)$ is closer to its predictor than the inverse value. Otherwise, there is an error and we notify this in the error location map.

Image encryption

During the next step, the original image is encrypted by using the encryption key $K_e = (p, x_0)$. The elements of the secret key are used as parameters of a chaotic generator (Piecewise Linear Chaotic Map [1, 11]).

Only some simple operations are needed for each iteration:

$$x_i = F(x_{i-1}) = \begin{cases} x_{i-1} \times \frac{1}{p} & \text{if } 0 \le x_{i-1} < p, \\ (x_{i-1} - p) \times \frac{1}{0.5-p} & \text{if } p \le x_{i-1} < 0.5, \\ F(1 - x_{i-1}) & \text{else,} \end{cases} \quad (2)$$

where $p \in [0, 0.5]$ and $x_i \in [0, 1]$.

As shown in Fig. 3, a sequence of pseudo-random bits $b(i,j)^k$ is obtained and used to encrypt the original image, pixel by pixel:

$$p_e(i,j)^k = b(i,j)^k \oplus p(i,j)^k, \quad (3)$$

where $0 \le k < 8$ and refers to the number of the bit in a pixel (from MSB to LSB) and $\oplus$ represents the XOR operation.

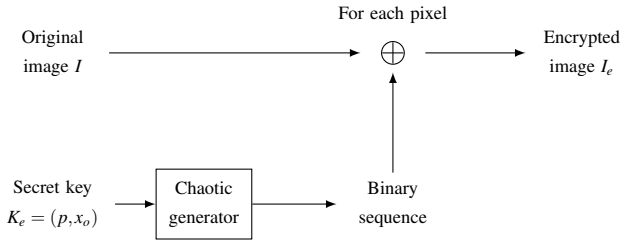


Figure 3: Encryption step.

Embedding of the error location map

Before the embedding step, the encrypted image is adapted to avoid prediction errors. Thanks to the error location map, it is possible to identify all the errors. The encrypted image is divided into blocks of eight pixels. We scan the encrypted image block by block and if there is at least one error in a block, we surround it by two flags: we substitute all the MSB values in the previous and following blocks by 1. In the current block, we replace the MSB value of a pixel by 1 if there is an error or otherwise by 0 (Fig. 4). If there is no error in a block, we pseudo-randomly substitute the MSB of each pixel by 0 or 1.

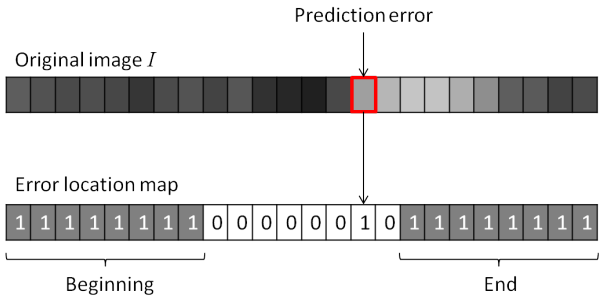


Figure 4: Building of the error location map.

Data embedding

In the data embedding phase, as shown in Fig. 1, someone can embed data in the encrypted image even if they do not have the encryption key and, so, they cannot access to the original image content. From the error location map, with a data hiding key, they can encrypt the to-be-inserted message. In this way, it is not possible to detect its presence after the embedding in the marked encrypted image I_{ew}. After that, they scan pixels of the encrypted image I'_e from left to right, and from top to bottom (S-order) and substitute the MSB of each pixel which can be marked by one bit b_k of the secret message, with $0 \le k < l$, where l is the number of pixels which can be marked:

$$p_{ew}(i,j) = b_k \times 128 + (p'_e(i,j) \bmod 128). \quad (4)$$

Data extraction and image recovery

In this phase, three cases are considered: (1) the recipient has only the data hiding key, (2) the recipient has only the encryption key and (3) the recipient has both the encryption and the watermarking keys.

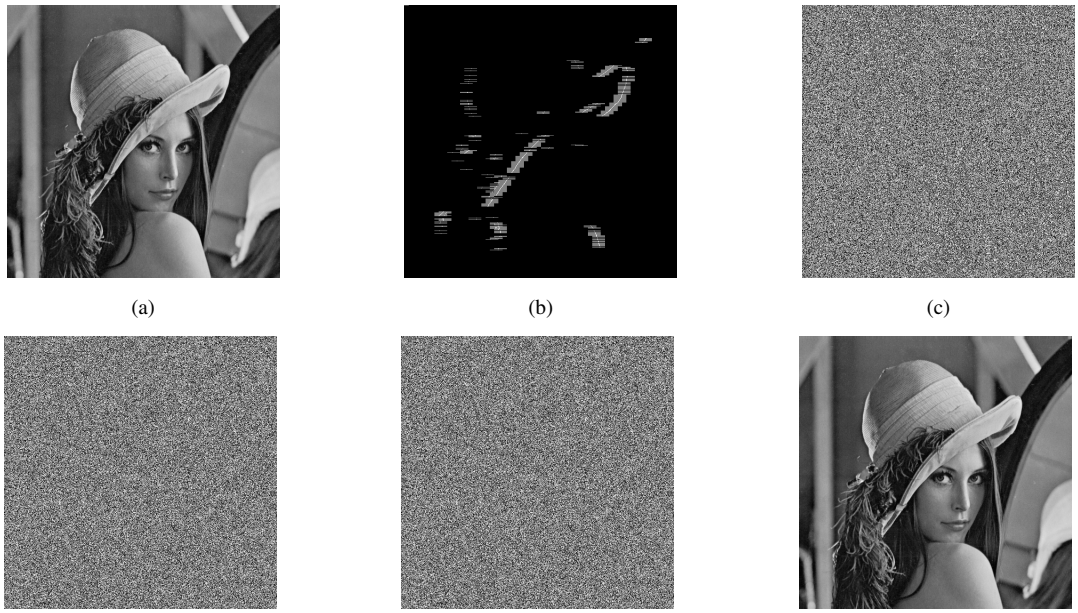
In the first case, the recipient can extract the secret message by following these steps:

1. Scan the pixels of the marked-encrypted image I_{ew} in the S-order and for each pixel, extract the MSB value and store it. Before the first sequence of eight MSB equal to 1, the extracted values are bits of the embedded message.
2. When such a sequence is encountered, it indicates the beginning of an error sequence: the next pixels were not marked during the data hiding step. So, scan pixels until the next sequence of eight MSB equal to 1, which indicates the end of the error sequence.
3. Repeat this process until the end of the image.
4. Finally, use the data hiding key to obtain the clear text of the secret message.

In the second case, the recipient can reconstruct the original image I by using MSB prediction:

1. Use the encryption key to generate the pseudo-random chaotic sequence.
2. Scan the pixels of the marked-encrypted image I_{ew} in the S-order and for each pixel, retrieve the seven least significant bits (LSB) of $p(i,j)$ by XORing the marked encrypted pixel value $p_{ew}(i,j)$ with the associated binary sequence in the pseudo-random chaotic stream. Only the MSB value could be wrong.
3. Predict the MSB value:
   - Consider $p(i,j)^{\text{MSB}=0}$ and $p(i,j)^{\text{MSB}=1}$ as the pixel value with MSB = 0 and MSB = 1, respectively. Note that there is a difference equal to 128 between these two values.
   - Calculate the absolute difference between each of these two values with $p(i,j-1)$ and with $p(i-1,j)$:

$$\begin{cases} \Delta^0 = \min(\left|p(i,j)^{\text{MSB}=0} - p(i,j-1)\right|, \left|p(i,j)^{\text{MSB}=0} - p(i-1,j)\right|) \\ \Delta^1 = \min(\left|p(i,j)^{\text{MSB}=1} - p(i,j-1)\right|, \left|p(i,j)^{\text{MSB}=1} - p(i-1,j)\right|) \end{cases} \quad (5)$$



(a) (b) (c)

(d) (e) (f)

Figure 5: Illustration of our proposed method on the test image Lena (512 x 512): a) Original image I, b) Unmarked pixels' location (errors and flags), number of errors = 442 (0.2%), c) Encrypted image I_e, d) Encrypted image I'_e with the pointed out prediction errors, e) Marked encrypted image I_{ew} with an embedding rate = 0.9641 bpp, f) Reconstructed image I, PSNR = ∞, SSIM = 1.

- The smaller value gives the original pixel value:

$$p(i,j) = \begin{cases} p(i,j)^{\text{MSB}=0}, \text{ if } \Delta^0 < \Delta^1 \\ p(i,j)^{\text{MSB}=1}, \text{ else.} \end{cases}$$

Finally, if the receiver has both the data hiding and encryption keys, they can extract the secret message and reconstruct the original image.

Experimental results

For data hiding methods in encrypted images, we have to measure different performances: embedding rate, number of incorrect extracted bits and recovered image quality after data extraction. It is necessary to find a trade-off between all of these parameters.

We applied our method on 10,000 different 512×512 gray level images[1]. We used the secret key $(p, x_o) = (0.123456789, 0.567894123)$. Except in very particular cases where there is an error in the detection of the flags, our method is fully reversible (PSNR $= \infty$ and SSIM $= 1$). As presented in Table 1, the embedding rate is high even though there are some MSB prediction errors. In order to better visualize the distribution of different image payloads, in Fig. 7 we randomly selected 500 images among the 10,000 tested images and applied our method.

[1]By using the image data base of BOWS-2: `http://bows2.ec-lille.fr/`

	Best case	Worst case	Average
Number of MSB prediction errors in the original image	0%	5.3%	0.2%
Payload (bpp)	1	0.3805	0.9681

Table 1: Payload measurements on a database of 10,000 images.

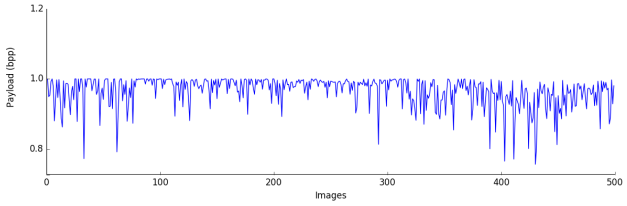
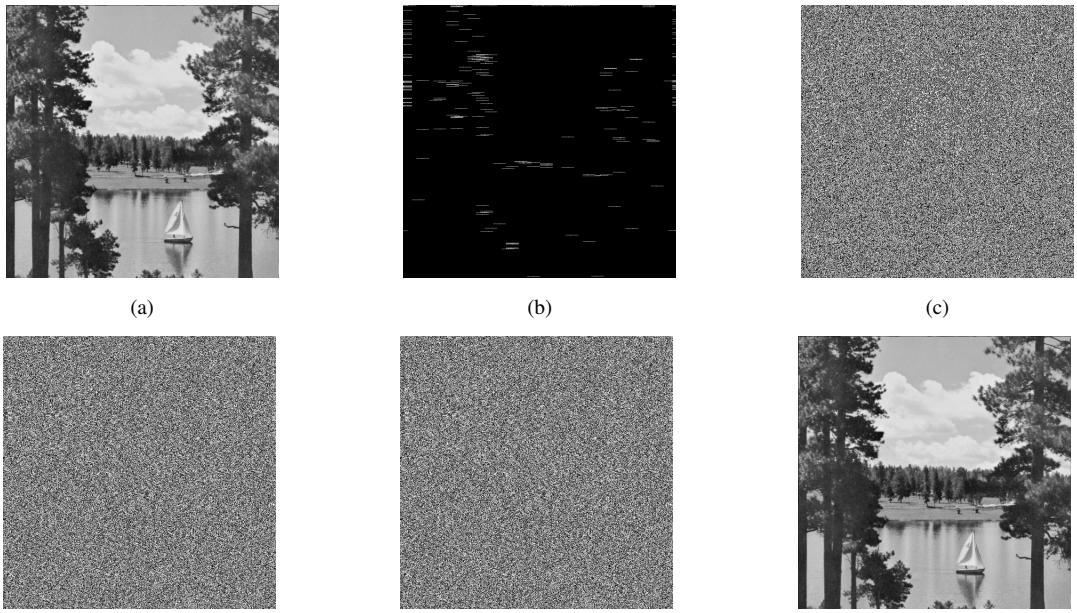


Figure 7: Payload measurements (in bpp) on a sample of 500 images.

We also present the detailed results of the proposed method applied on the 512×512 test images Lena (Fig. 5.a) and Lake (Fig. 6.a). Fig. 5.b and Fig. 6.b show the location of problematic pixels in white, *i.e.* pixels of the original images whose MSB would be badly predicted, and pixels in medium gray which will be used to highlight them. Note that the prediction errors are often on the contours and there are sometimes more than one error in the same block. Moreover, if there are errors in two adjacent blocks, the flag which indicates the end of the error sequence is shifted. In these two cases, the loss of embedding capacity is then less important. All other pixels, in black, are used to embed bits of the secret message. Fig. 5.c and Fig. 6.c are the encrypted images. Fig. 5.d and Fig. 6.d are the encrypted images with the highlighted



(a) (b) (c)

(d) (e) (f)

Figure 6: Illustration of our proposed method on the test image Lake (512 x 512): a) Original image I, b) Unmarked pixels' location (errors and flags), number of errors = 202 (0.1%), c) Encrypted image I_e, d) Encrypted image I'_e with the pointed out prediction errors, e) Marked encrypted image I_{ew} with an embedding rate = 0.9839 bpp, f) Reconstructed image I, PSNR = ∞, SSIM = 1.

prediction errors: the initial information and the errors location are not visible anymore. Fig. 5.e and Fig. 6.e present the marked encrypted images obtained in the final step of the encoding. In Fig. 5.f and Fig. 6.f, note that the reconstructed images are exactly the same as the original ones: all pixels are correctly reconstructed (PSNR $= \infty$, SSIM = 1).

We also made some comparisons between our proposed method and two existing ones: Zhang's method [14] and Wu and Sun's method [12], according to which is explained in the paper [12]. We used the four images presented in Fig. 8.

(a) Lena (b) Baboon (c) Airplane (d) Lake

Figure 8: Test images

First of all, note that our method is the only to be fully reversible in all cases. Indeed, only Lena image is perfectly the same than the original one by using Wu and Sun's method and none of the four images is identical with Zhang's method. Moreover, regarding the embedding capacity, our method obtains the best results: the payload is very high (close to 1 bpp) contrary to that obtained by Zhang and Wu and Sun (0.1563 bpp).

To conclude, our method proposes a very good trade-off between the embedding capacity and the reconstructed image quality: it is possible to hide a large amount of data in an encrypted image and to recover perfectly the original image after data extraction.

Test images	Methods	Embedding rate (bpp)	PSNR (dB)
Lena	Our	**0.9641**	$+\infty$
	Zhang	0.1563	44.65
	Wu and Sun	0.1563	$+\infty$
Baboon	Our	**0.7478**	$+\infty$
	Zhang	0.1563	38.79
	Wu and Sun	0.1563	40.57
Airplane	Our	**0.9889**	$+\infty$
	Zhang	0.1563	42.08
	Wu and Sun	0.1563	60.17
Lake	Our	**0.9839**	$+\infty$
	Zhang	0.1563	39.88
	Wu and Sun	0.1563	54.84

Table 2: Performance comparisons between Zhang's method [14], Wu and Sun's method [12] and our proposed method.

Conclusion

In this paper, we propose a high capacity reversible data hiding method in encrypted images based on the MSB prediction. By analyzing the content of the original image, all the prediction errors are highlighted and the encrypted image is modified accordingly. After that, by substituting most of the MSB values in the image, it is possible to hide a large message (payload close to 1 bpp). Finally, in the extraction phase, the original image is reconstructed without any errors (PSNR $= \infty$) and the secret message is perfectly extracted.

Further directions of this work include testing other predictors or other error flags in order to reduce the number of prediction errors and, in this way, improve the embedding capacity.

References

[1] A. Baranovsky and D. Daems. Design of one-dimensional chaotic maps with prescribed statistical properties. *International Journal of Bifurcation and Chaos*, 5(6): 1585–1598, 1995.

[2] L. Kocarev. Chaos-based cryptography: a brief overview. *IEEE Circuits and Systems Magazine*, 1(3): 6–21, 2001.

[3] X. Li, B. Ying and T. Zeng. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Transactions on Image Processing*, 20(12): 3524–3533, 2011.

[4] K. Ma, W. Zhang, X. Zhao, N. Yu and F. Li. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, 8(3): 553–562, 2013.

[5] Z. Ni, Y. Q. Shi, N. Ansari and W. Su. Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(3): 354–362, 2006.

[6] W. Puech, M. Chaumont and O. Strauss. A reversible data hiding method for encrypted images. *Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, San Jose, CA, USA, 6819: 68191E-1–68191E-9, 2008.

[7] V. Sachnev, H. J. Kim, J. Nam, S. Suresh and Y. Q. Shi. Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7): 989–999, 2009.

[8] D. M. Thodi and J. J. Rodriguez. Expansion embedding techniques for reversible watermarking. *IEEE Transactions on Image Processing*, 16(3): 721–730, 2007.

[9] J. Tian. Reversible data embedding using a difference expansion. *IEEE Transactions on circuits and systems for video technology*, 13(8): 890–896, 2003.

[10] P. Tsai, Y. C. Hu and H. L. Yeh. Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Processing*, 89: 1129–1143, 2009.

[11] Y. Wang and L. Yang. Design of pseudo-random bit generator based on chaotic maps. *International Journal of Modern Physics B*, 26(32), 2012.

[12] X. Wu and W. Sun. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Processing*, 104(2014): 387–400, 2014.

[13] W. Zhang, K. Ma and N. Yu. Reversibility improved data hiding in encrypted images. *Signal Processing*, 94: 118–127, 2014.

[14] X. Zhang. Separable reversible data hiding in encrypted images. *IEEE Transactions on Information Forensics and Security*, 7(2): 826–832, 2012.

[15] X. Zhang and S. Wang. Efficient steganographic embedding by exploiting modification direction. *IEEE Communications Letters*, 10(11): 781–783, 2006.

Author Biography

Pauline PUTEAUX received the M.S. degree in Computer Science and Applied Mathematics from the University of Grenoble, France, in 2016. Currently, she specializes in cryptography. Her work has focused on multimedia security in particular in image processing in the encrypted domain.

William PUECH received a diploma in Electrical Engineering from the University of Montpellier, France, in 1991 and a 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 has been an Assistant Professor at the University of Toulon, France, with research interests including methods of active contours applied to medical images sequences. Between 2000 and 2008, he has been Associate Professor and since 2009, he is full Professor in image processing at the University of Montpellier, France. He works in the LIRMM Laboratory (Laboratory of Computer Science, Robotic and Microelectronic of Montpellier). His current interests are in the areas of protection of visual data (images, videos and 3D objects) for safe transfer by combining watermarking, data hiding, compression and cryptography. He has applications on medical images, cultural heritage and video surveillance. He is the head of the ICAR team (Image & Interaction) and he has published more than 40 journal papers, 16 book chapters, more than 100 conference papers and 3 patents. W. Puech is associate editor of J. of Advances in Signal Processing, Springer, Signal Processing: Image Communications, Elsevier and Signal Processing, Elsevier and he is reviewer for more than 15 journals (IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, IEEE TCSVT, IEEE TIFS, Signal Processing: Image Communication, Multimedia Tools and Applications ...) and for more than 10 conferences (IEEE ICIP, EUSIPCO, ...).