High-Capacity Data Hiding in Encrypted Images using MSB Prediction

Pauline Puteaux, Dave Trinel, William Puech

High-Capacity Data Hiding in Encrypted Images using MSB Prediction

Pauline Puteaux[1], Dave Trinel [21] and William Puech[1]
[1] LIRMM Laboratory, UMR 5506
CNRS, University of Montpellier
Montpellier, France
e-mail: {pauline.puteaux, dave.trinel, william.puech}@lirmm.fr
[2] TISBio, UMR 8576
UGSF, Glycobiology Structural and Functional Unit
CNRS, Lille University of Science and Technology
Lille, France

***Abstract*— In the last few years, visual privacy has become a major problem. Because of this, encrypted image processing has received a lot of attention within the scientific and business communities. Data hiding in encrypted images (DHEI) is an effective technique to embed data in the encrypted domain. The owner of an image encrypts it with a secret key and it is still possible to embed additional data without knowing the original content nor the secret key. This secret message can be extracted and the initial image can be recovered in the decoding phase. Recently, DHEI has become an investigative field, but the proposed methods do not allow a large amount of embedding capacity. In this paper, we present a new method based on the MSB (most significant bit) prediction. We suggest to hide one bit per pixel by pre-processing the image to avoid prediction errors and, thereby, to improve the quality of the reconstructed image. We have applied our method to various images and, in every cases, the obtained image is very similar to the original one in terms of PSNR or SSIM.**

***Keywords*— Image pre-processing, image encryption, image recovery, data hiding, MSB prediction.**

I. Introduction

Reversible data hiding (RDH) consists of embedding secret data in a signal (*e.g.* an image). After its extraction, it is fundamental to reconstruct the original image with a minimum of errors or preferably, none at all. Indeed, in some strict areas, like in the military or medical world, distortion is unacceptable: each bit of information is important.

Fridrich *et al.* were the first to describe a RDH method [2]. They suggested compressing the least significant binary (LSB) planes of an image and to insert the to-be-embedded message in the vacating room. Although easy to implement and with good quality for the reconstructed image, this technique is not robust to steganalysis. Thereafter, a lot of schemes were proposed. Zhang *et al.* suggested exploiting the set of modification direction for a pixel (EMD) [26]. Tian *et al.* presented their method of difference expansion (DE) [16], [17], using the Haar transform. It consists of calculating the differences between one pixel and its adjacent neighbors values and to select some of them to define the DE, where all the information is dissimulated. Methods based on histogram modification have also been described. Some proposed to build and to exploit the histogram according to the grayscale values [9], [14] and others by using statistical data [3], [18].

In RDH, 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). Recently, for the purpose of answering to this problem, new methods based on prediction error analysis (PE) and their expansion (PEE) have emerged. The main idea is to exploit the correlation between a pixel and its adjacent neighbors [6], [7], [10], [11], [13], [15], [19].

Otherwise, for data privacy, it is sometimes necessary to make an image unreadable. For this reason, a lot of encryption methods exist. These can be divided into two groups, depending if a block cipher or a stream cipher is used. Sometimes, in the second group, a new kind of algorithms, based on chaos, has been designed [1], [5], [20]. Moreover, in image processing, encryption can be full or selective.

In some cases, it may be interesting to combine image encryption and RDH (RDHEI). For example, in the medical community, a radiologist, bound by professional secrecy, has to encrypt a patient's X-rays. It must be possible to insert some information about the patient in these images without knowing their original content. 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.

Puech *et al.* proposed to encrypt an image before embedding the secret message by using AES [12]. After that, they embedded a bit of the hidden message at a randomly selected position in each block. The original image was reconstructed by comparing values of the standard deviation in each block, calculated by assuming that the value of the hidden bit is 0 or 1. Zhang *et al.* method [24] consisted of image encryption, with a XOR operation, and data insertion by changing a small part. The image was divided into blocks with the same size and partitioned it into two groups. The three LSB of each pixel were modified in one of these groups, according to the

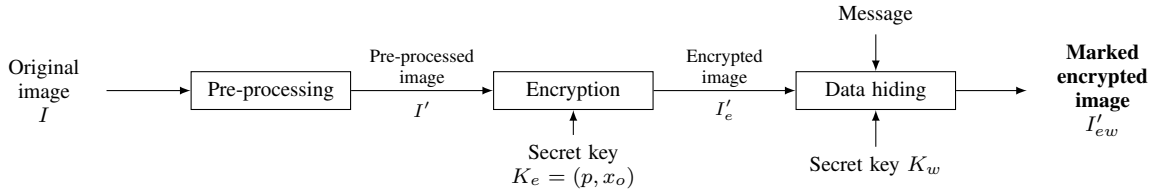


Fig. 1: Overview of the encoding method.

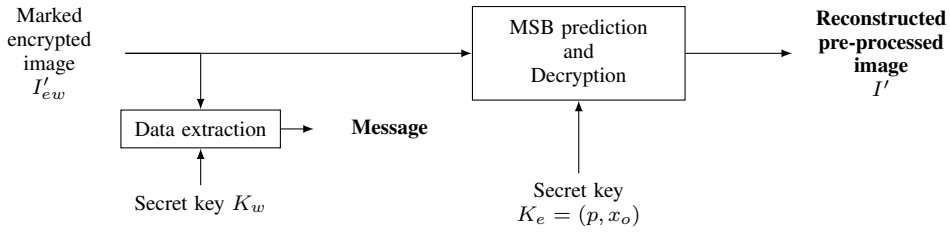


Fig. 2: Overview of the decoding method.

to-be-inserted bit. Hong *et al.* [4] improved this technique by considering the local complexity for each pixel during the encryption step. Zhang [25] proposed a separate method: a part of the encrypted image was compressed and the vacated space served to embed secret data. Data extraction could be done before or after image decryption. Ma *et al.* were the first to introduce a RRBE technique [8]: a part of the original image was released before encryption, by using a RDH method of histogram shifting and some LSB were substituted in the encrypted image to embed data. Zhang *et al.* analyzed the prediction errors (PE) of some pixels and made space to hide data by PE-histogram shifting before image encryption [23]. Wu and Sun also proposed a joint and a separate methods [22]. In the first one, they encrypted the original image in the same way as Zhang in [25]. After that, they partitioned the encrypted image to select pixels to embed data, according to a data hiding key. Then, they used a histogram shifting method. In the separative method, the to-be-inserted bits were hidden by MSB substitution. During the decoding phase, these bits could be extracted with the data hiding key and the original image was reconstructed by using a median filter on the watermarked image.

None of these 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 $+\infty$. In [8], 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 approximately 0.1 bit per pixel at most [22].

In this paper, we introduce a new data hiding method for encrypted images based on MSB prediction with a very high capacity. During a pre-processing step, some pixel values are modified to avoid prediction errors, without significantly altering image quality. Thanks to this, it is possible to hide one bit per pixel by MSB substitution.

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

II. Proposed method

In this section, we introduce our separate data hiding method in encrypted images. The encoding phase consists of three steps: image pre-processing, encryption and data hiding by MSB substitution, as shown in Fig. 1. The original image owner pre-processes and encrypts it by using the encryption key K_e and another person embeds an additional message by using the data hiding key K_w. For the decoding phase, there are three possible schemes. If the recipient has only the encryption key, they can only obtain the pre-processed image but not the embedded message. Conversely, if they only have the watermarking key, they can just extract the message. Obviously, when both the encryption and the watermarking keys are obtained, the recipient can extract the original message and reconstruct the pre-processed image. The overview of this decoding method is presented in Fig. 2.

A. Pre-processing

Our method proposes to hide the secret message by replacing each MSB value of the image pixels by one bit of the hidden message. Thereby, the original MSB values are lost during the data hiding phase and, to reconstruct the original image, it is important to be able to predict them without any errors.

	$p(i-1,j)$
$p(i,j-1)$	$p(i,j)$

Fig. 3: Context of a pixel $p(i,j)$.

Then, we propose to process the original image I in order to obtain an image I' without any error prediction. For the prediction, for each current pixel $p(i,j)$, we propose to use its two adjacent pixels $p(i,j-1)$ and $p(i-1,j)$, as illustrated Fig. 3. First, we calculate the average of the two values. Then, we calculate the difference between this average and $p(i,j)$ but also between this average and the inverse value of $p(i,j)$, which is $(p(i,j)+128) \bmod 256$. If the difference with the inverse value is smaller than with $p(i,j)$, there will be a prediction error in the decoding phase. To avoid this, we must change the pixel value $p(i,j)$ to get $p'(i,j)$. The detailed method is presented in Algorithm 1.

Algorithm 1: Pre-processing algorithm.

Data: Original $m \times n$ image I
Result: Pre-processed image I'
for $i \leftarrow 0$ **to** m **do**
  for $j \leftarrow 0$ **to** n **do**
    $inv(i,j) \leftarrow (p(i,j)+128)\%256$;
    if $i = 0$ **and** $j = 0$ **then**
      $pred(i,j) \leftarrow p(i,j)$;
    else if $i = 0$ **then**
      $pred(i,j) \leftarrow p(i,j-1)$;
    else if $j = 0$ **then**
      $pred(i,j) \leftarrow p(i-1,j)$;
    else
      $pred(i,j) \leftarrow \frac{p(i-1,j)+p(i,j-1)}{2}$;
    if $\mid pred(i,j) - p(i,j) \mid \geq \mid pred(i,j) - inv(i,j) \mid$ **then**
      if $p(i,j) < 128$ **then**
        $p'(i,j) = pred(i,j) - 63$;
      else
        $p'(i,j) = pred(i,j) + 63$;
    else
      $p'(i,j) = p(i,j)$

B. Image encryption

During this phase, the pre-processed image I' is encrypted to obtain the encrypted image I'_e. Piecewise Linear Chaotic Map (PWLCM) is one of the simplest chaotic systems [1], [20]. Only simple operations are needed for each iteration:

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

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

Use elements of the secret key $K_e = (p, x_0)$ as parameters of this chaotic generator. As shown in Fig. 4, a sequence of pseudo-random bits $b(i,j)^k$ is obtained and used to encrypt the original image, pixel by pixel:

$$p'_e(i,j)^k = b(i,j)^k \oplus p'(i,j)^k, \quad (2)$$

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

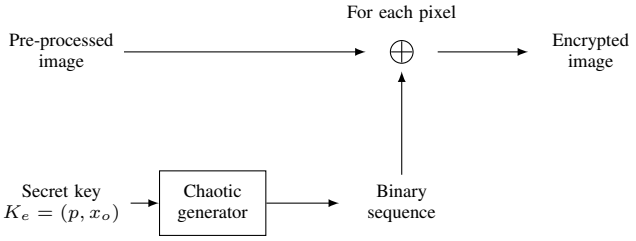


Fig. 4: Encryption step.

C. Data embedding

In the data embedding phase, we can embed data in the encrypted image even if we do not have the encryption key and, so, we cannot access the original image content. By using the data hiding key, the to-be-inserted message is encrypted. In this way, it is not possible to detect its presence after the embedding in the marked encrypted image. After that, we scan pixels of the encrypted image from left to right, then from top to bottom (S-order) and substitute the MSB of each pixel by one bit b_l, with $0 \leq l < m \times n$, of the secret message, except for the first pixel:

$$p'_{ew}(i,j) = b_l \times 128 + (p'_e(i,j) \bmod 128). \quad (3)$$

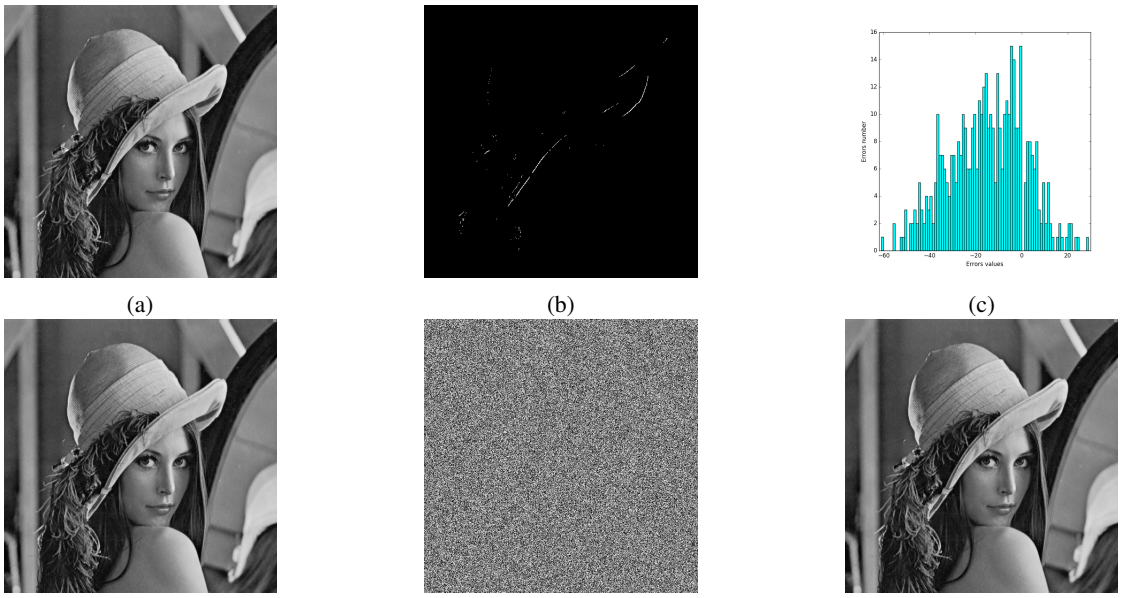
D. Data extraction and image recovery

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 scans the pixels in the S-order from the marked encrypted image I'_{ew}, and extracts the MSB of each pixel, except for the first pixel. After that, they just need to use the data hiding key to obtain the clear text. However, they cannot retrieve the pre-processed image I'.

In the second case, the recipient can reconstruct I' by adopting the following approach:

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:
   - If the considered pixel is not located on the first line or column, average $p'(i,j-1)$ and $p'(i-1,j)$ values. On the other hand, just consider $p'(i,j-1)$, if it is on the first line, or $p'(i-1,j)$, if it is on



(a) (b) (c)

(d) (e) (f)

Fig. 5: Experiment using our method, embedding rate = 1 bpp. a) Original image, b) Errors' location, number of errors = 448 (0.2%), c) Histogram of the estimated prediction errors, d) Pre-processed image, PSNR = 48.67 dB, e) Marked encrypted image, f) Reconstructed image, PSNR = 48.67 dB, SSIM = 0.9998.

the first column. Record this value as a predictor $pred(i,j)$.

- 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 $pred(i,j)$ and each of these two values; finally, the smaller value gives the original pixel value:

If $\mid p'(i,j)^{\text{MSB}=0} - pred(i,j) \mid < \mid p'(i,j)^{\text{MSB}=1} - pred(i,j) \mid$
then $p'(i,j) = p'(i,j)^{\text{MSB}=0}$,
Else
$p'(i,j) = p'(i,j)^{\text{MSB}=1}$.

In the third case, if the receiver has the data hiding and encryption keys, he can extract the secret message and reconstruct the pre-processed image in the same way as explained previously.

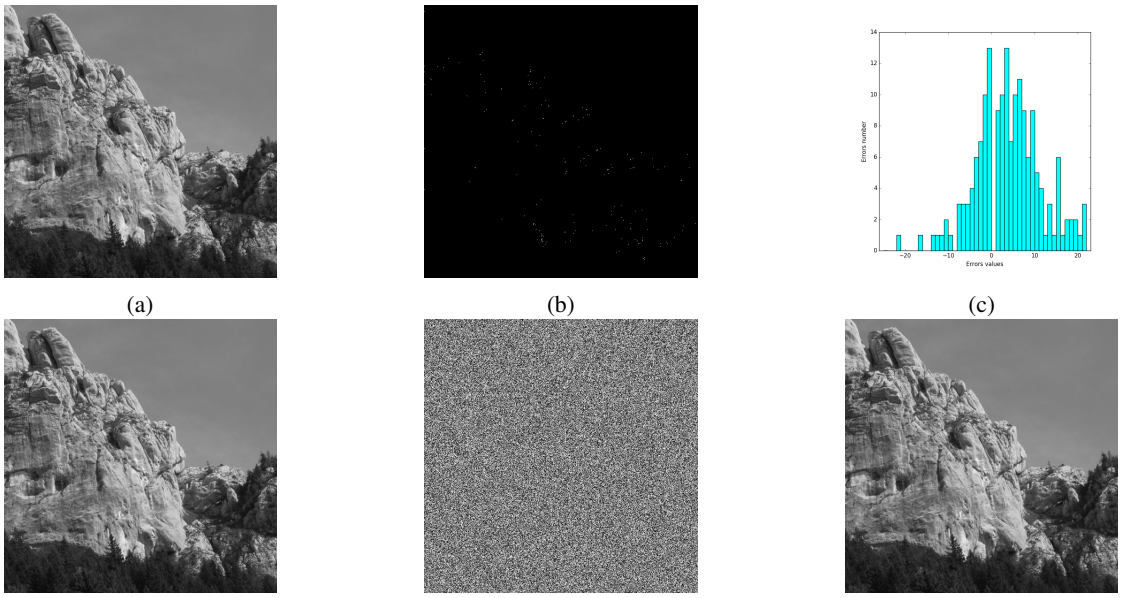
III. Experimental results

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.

To evaluate image quality, we used two indicators: peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [21]. We show the results of the proposed method applied on the test image Lena (512 × 512 pixels), Fig. 5.a and on another image with a lower number of errors (512 × 512 pixels), Fig. 6.a. For these two images, we used the secret key $(p, x_o) = (0.123456789, 0.567894123)$. Fig. 5.b and Fig. 6.b show the location of the problematic pixels, *i.e.* pixels of the original image whose MSB would be badly predicted if their value is not changed. Histograms in Fig. 5.c and Fig. 6.c display the required pixels modifications to avoid these prediction errors. After this pre-processing step, the adapted images are obtained (Fig. 5.d and Fig. 6.d). In Fig. 5.e and Fig. 6.e, one can notice that the initial information is not visible anymore after the encryption and watermarking steps. Finally, after data extraction, reconstructed images are exactly the same as pre-processed images and very similar to original images, even when some pixel values were changed during the pre-processing step (Fig. 5.f and Fig. 6.f). In both cases, PSNR is high and SSIM is close to 1: PSNR = 48.67 dB, SSIM = 0.9998 and PSNR = 61.11 dB, SSIM = 0.9999 for Lena and the second image respectively.

We applied our method on 500 different 512 × 512 gray level images[1]. In all cases, the embedding rate is 1 bpp and all bits of the hidden message are correctly extracted. In 9.2% of cases, when there is no prediction error (*i.e.* all the original pixel values are below or up to 128), our method is totally reversible. In this case, the original image is recovered without any errors, as indicated by PSNR of $+\infty$ and SSIM of 1. In other cases, the reconstructed image quality is high, as presented in Table 1.

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



(a) (b) (c)

(d) (e) (f)

Fig. 6: Experiment using our method, embedding rate = 1 bpp. a) Original image, b) Errors' location, number of errors = 174 (0.1%), c) Histogram of the estimated prediction errors, d) Pre-processed image, PSNR = 61.11 dB, e) Marked encrypted image, f) Reconstructed image, PSNR = 61.11 dB, SSIM = 0.9999.

	Best case (9.2%)	Worst case	Average
Number of MSB prediction errors in the original image	0%	3.2%	0.3%
PSNR (dB)	$+\infty$	36.06	54.84
SSIM	1	0.9966	0.9998

Table 1: Images quality measurements on a database of 500 images.

In order to generate Fig. 7, we randomly selected 100 images among the 500 tested and applied three methods: LSB substitution, naive MSB method (without pre-processing) and our proposed method. All of these methods have the same payload (1 bpp). However, ours allows to reconstruct images globally with a better visual quality. Indeed, by using the LSB substitution, PSNR is close to 51 dB in all cases. Note that it was easy to speculate this using with a simple probability calculation. With a naive MSB method, results are generally worse because there is a phenomenon of error propagation during the pixel values prediction, excepting when there is no prediction error (9.2% of the cases). Ignoring these 9.2% of cases where PSNR is equal to $+\infty$, PSNR is often below 20 dB. Sometimes, it is a little better – when there are only a few errors – but there are artifacts in the reconstructed image for all the pixels with the bad MSB value (difference of 128 with the correct value). Finally, with our method, PSNR is high, as illustrated in Table 1 with an average of 54.84 dB for 500 images: we avoided all the prediction errors thanks to the pre-processing step.

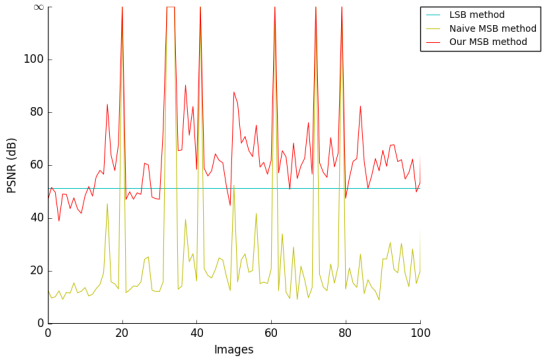


Fig. 7: Image quality comparison between our method and other methods with the same payload (1 bpp).

We also made some comparisons between our proposed method and two existing ones: Zhang's method [25] and Wu and Sun's method [22], according to which is explained in the paper [22]. We used very well known images such Lena, Baboon, Airplane and Lake. Note in Table 2 that our method has a very high payload (1 bpp), contrary to that obtained by Zhang and Wu and Sun (0.1563 bpp). Regarding the reconstructed image quality, none of these three methods is totally reversible in every cases. Only the Lena image is exactly the same as the original one by using Wu and Sun's method. The results of our method are better than those of Zhang. They are not quite as good as those of Wu and Sun, but very similar. In order to compare the recovered image quality exclusively, we chose to embed regularly one pixel every six in such a way as to have approximately the same

embedding capacity (0.1667 bpp) as Zhang and Wu and Sun's methods. Our results are better than those of Zhang and Wu and Sun: PSNR is higher, except when the original image is perfectly reconstructed. Finally, note that the results of our method and Wu and Sun's one are similar in terms of image quality when we watermark regularly one pixel every two (payload = 0.5 bpp).

In conclusion, our method allows a very good trade-off between the embedding rate and the recovered image quality after data extraction.

Test images	Methods	Embedding rate (bpp)	PSNR (dB)
Lena	Our	**1**	48.67
		0.5	52.08
		0.1667	57.58
	Zhang	0.1563	44.65
	Wu and Sun	0.1563	$+\infty$
Baboon	Our	**1**	39.41
		0.5	44.00
		0.1667	**48.82**
	Zhang	0.1563	38.79
	Wu and Sun	0.1563	40.57
Airplane	Our	**1**	57.24
		0.5	60.88
		0.1667	**64.55**
	Zhang	0.1563	42.08
	Wu and Sun	0.1563	60.17
Lake	Our	**1**	52.23
		0.5	54.31
		0.1667	**61.89**
	Zhang	0.1563	39.88
	Wu and Sun	0.1563	54.84

Table 2: Performance comparisons between Zhang's method [25], Wu and Sun's method [22] and our proposed method.

IV. Conclusion

In this work, we proposed a new data hiding method in encrypted images based on prediction with a very high capacity (1 bpp). Indeed, by replacing all the MSB in the image, it is possible to hide one bit per pixel. In addition to this excellent embedding capacity, the reconstructed image quality is high (SSIM close to 1, PSNR ≈ 55 dB).

Future work on this method is to improve the image recovery. Indeed, during the pre-processing step, we modify only the current pixel to avoid all prediction errors. In most cases, it is a small modification, but sometimes – when the inverse pixel value is very close to the predictor value – it is necessarily greater. To obtain the best possible image quality, it is better to slightly change some pixels rather than making a big modification of one value, according to the mean squared error calculation. So, it may be interesting to use a new predictor, similar to the MED predictor. Finally, our ultimate aim would be to find a fully reversible method with the same high payload.

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] J. Fridrich, M. Goljan and R. Du. Reliable detection of LSB steganography in color and grayscale images. *Proc. International Workshop on Multimedia and Security*, 27–30, 2001.

[3] X. Gao, L. An, Y. Yuan, D. Tao and X. Li. Lossless data embedding using GSQ histogram. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8): 1061–1070, 2011.

[4] W. Hong, T. S. Chen and H. Wu. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Processing Letters*, 19(4): 199–202, 2012.

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

[6] 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.

[7] X. Li, W. Zhang, X. Gui and B. Yang. Reversible data hiding based on multiple histograms modification. *IEEE Transactions on Information Forensics and Security*, 10(9): 2016–2027, 2015.

[8] 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.

[9] 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.

[10] B. Ou, X. Li, Y. Zhao, R. Ni and Y. Q. Shi. Pairwise prediction error expansion for efficient reversible data hiding. *IEEE Transactions on Image Processing*, 22(12): 5010–5021, 2013.

[11] F. Peng, X. Li and B. Yang. An adaptive PEE-based reversible data hiding scheme exploiting referential prediction-errors. *IEEE International Conference on Multimedia and Expo*, 2015.

[12] 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.

[13] 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.

[14] W. L. Tai, C. M. Yeh and C. C. Chang. Reversible data hiding based on histogram modification of pixel difference. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(6): 906–910, 2009.

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

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

[17] J. Tian. Reversible watermarking by difference expansion. *Proc. International Workshop on Multimedia and Security*, 19–22, 2002.

[18] 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.

[19] L. L. Wan, F. Chen, H. J. He and L. Zhang. Reversible data hiding scheme based on prediction error sorting and double prediction. *Proceedings of APSIPA Annual Summit and Conference*, 2015.

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

[21] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

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

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

[24] X. Zhang. Reversible data hiding in encrypted image. *IEEE Signal Processing Letters*, 18(4): 255–258, 2011.

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

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