EPE-based Huge-Capacity Reversible Data Hiding in Encrypted Images

Pauline Puteaux, William Puech

EPE-based Huge-Capacity Reversible Data Hiding in Encrypted Images

Pauline Puteaux and William Puech
LIRMM – CNRS, Univ. Montpellier
Montpellier, France
{pauline.puteaux, william.puech}@lirmm.fr

Abstract

Reversible data hiding in encrypted images (RDHEI) consists of embedding data in the encrypted domain. In current state-of-the-art methods, most of them use least significant bit (LSB) substitution or prediction, but fail to embed a significant amount of information. Recently, a new class of RDHEI method, based on most significant bit (MSB) substitution, has emerged. By exploiting the natural correlation between pixels in the clear domain, it is possible to have a payload close to 1 bpp with a very high image quality, without adding overhead. In particular, in the approach based on embedded prediction errors (EPE-based approach) [6], the authors propose to embed the prediction error location information in the encrypted MSB-plane. In this paper, we present a huge-capacity RDHEI (HC-RDHEI) method. In fact, we are interested in improving the proposed EPE-based RDHEI approach by using recursively other bit-planes, from MSB to LSB as long as it is possible. Indeed, depending on the image content, bit-planes can easily be predicted, and so most of them can be substituted by bits of a secret message. According to the obtained results, the payload can be much higher than 1 bpp (median equal to 1.749 bpp, on average 1.836 bpp, and 5.408 bpp in the best case), while preserving perfect reversibility.

1. Introduction

During the transmission or the archiving of encrypted images, it is often necessary to analyze or process them, without knowing the original content or the secret key used during the encryption phase. In recent years, this topic has attracted much research attention, and different image processing methods in the encrypted domain have been developed [3].

In particular, methods of reversible data hiding in encrypted images (RDHEI) are used for data enrichment and authentication in the encrypted domain. During the decoding phase, the message has to be extracted without error, and the original image has to be perfectly recoverable. Methods can be separated into two groups, depending if the space to embed the secret message is released before the encryption phase [4] or after [9]. In addition, encryption and data embedding can be done jointly [5] or separately [9, 7].

Wu and Sun describe two schemes: a joint and a separate version [8]. In the joint approach, according to the data hiding key, they select some pixels from the encrypted image for data embedding and they release some space by histogram shifting. In the separate approach, they use a most significant bit (MSB) substitution. During the decoding step, as most MSB values are lost, a median filter is then applied to the reconstructed image in order to remove the visual defects. Cao *et al.* propose a sparse coding technique and are able to hide a large amount of information (approximately 1 *bpp*), by exploiting the local correlation between pixels [2]. Zhang *et al.* use public-key cryptography for image encryption and embed data in the least significant bit-planes (LSB-planes) of the encrypted pixels [10]. As the image is slightly altered, the secret message can be extracted without error and the original image can be perfectly recovered. Recently, Puteaux and Puech introduced a new RDHEI method based on MSB prediction [6], which is one of the first methods proposing to use MSB prediction instead of LSB prediction. The main benefit is that, in the clear domain, predicting the MSB values is easier than the LSB values. For this reason, the numbers of prediction errors (PE) is relatively low. In the approach based on embedded prediction errors (EPE-based approach), they propose to embed the prediction error location information in the encrypted MSB-plane, without adding overhead. Finally, the authors achieve a high embedding capacity, using only one bit-plane for data embedding.

In this paper, we propose a huge-capacity RDHEI (HC-RDHEI) method. As an extension of the EPE-based approach proposed in [6], we suggest using all bit-planes in a recursive way, rather than the MSB-plane only. Indeed, the MSB-plane is not the only bit-plane which can be predicted and used for the data hiding step. In the proposed approach, starting from the MSB-plane, each bit-plane of the original image is analyzed recursively in order to highlight prediction errors, *i.e.* locate the pixels which cannot be

predicted according to their neighbors, and then encrypted. If the amount of data to embed the PE is available in the current encrypted bit-plane, then, the PE highlighting process is performed and the data hiding step can be applied for this bit-plane. After decoding, with this new proposed approach, we are able to perfectly reconstruct the original image, by using the PE location information and prediction.

The remainder of this paper is organized as follows. Section 2 describes in detail the proposed method. Experimental results are presented in Section 3. Finally, the conclusion is provided in Section 4.

2. Proposed HC-RDHEI method

In this section, we introduce our proposed recursive method of huge capacity reversible data hiding in encrypted images, which is EPE-based and allows a very high capacity.

An overview of the encoding phase is illustrated in Fig. 1. We suggest processing recursively each bit-plane of the original image I. First step consists of detecting the prediction errors (PE), *i.e.* pixels which cannot be predicted according to their neighbors, and calculating the PE location information. Then, each bit-plane is encrypted using the encryption key K_e and, if possible, the PE location information is embedded in the encrypted bit-plane. When a bit-plane cannot be marked, the PE detection process is interrupted and the remaining bit-planes are just encrypted. The to-be-marked encrypted image I_{e_pe} and the byte *marked*, used to understand which bit-plane has been marked, are then transmitted. The data hider can then embed the secret message by bit substitution in the to-be-marked bit-planes using a data hiding key K_{dh}. The detailed encoding phase is described in Section 2.1 and Section 2.2.

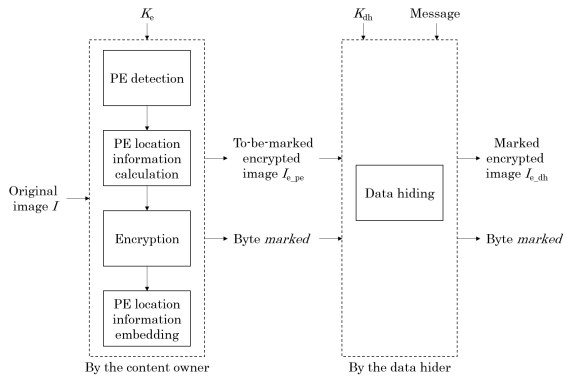


Figure 1. Overview of the encoding phase of our proposed EPE-based HC-RDHEI method.

2.1. On the content owner side

In this method, the original image I is considered as a stack of eight bit-planes $I^{[k]}$, with $0 \leq k \leq 7$. From the MSB-plane $I^{[0]}$ to the LSB-plane $I^{[7]}$, each bit-plane $I^{[k]}$ is processed recursively. Finally, the to-be-marked encrypted image I_{e_pe} is obtained after the processing of all bit-planes, as illustrated in Fig. 2. This image is also transmitted to the data hider, along with the byte *marked*. Each bit of index k from this byte, with $0 \leq k \leq 7$, from MSB to LSB, indicates if the bit-plane $I_{e\_pe}^{[k]}$ has been marked or not.

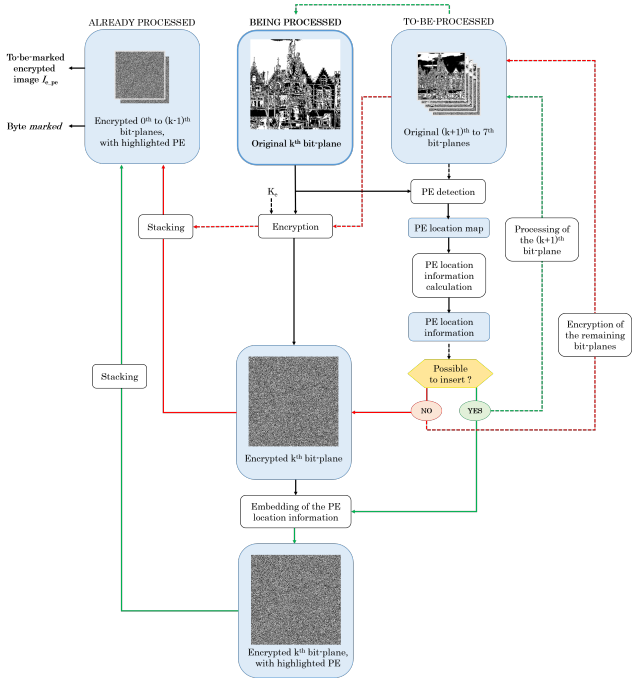


Figure 2. Overview of the original image I processing method, by the content owner.

Algorithm 1 presents the recursive function to process each bit-plane $I^{[k]}$ of the original image. First step consists in prediction error (PE) detection in the current bit-plane, and calculation of the PE location information. Therefore, the bit-plane is encrypted using the encryption key K_e. The algorithm then calculates if it is possible to embed the PE location information in the encrypted bit-plane $I_e^{[k]}$ (*i.e.* if the size of the PE location information is smaller than the bit-plane size). In this case, it is embedded by bit-substitution, and the to-be-marked encrypted bit-plane $I_{e\_pe}^{[k]}$ is obtained (the byte *marked* is modified in order to contain this information). Moreover, the next bit-plane $I^{[k+1]}$ of the original image is processed recursively. Or else, the current and all remaining bit-planes are just encrypted and cannot be marked ($I_{e\_pe}^{[k]} = I_e^{[k]}$).

2.1.1 Prediction error detection and highlighting

In the proposed method, message embedding is made by bit substitution. For this reason, original bit values are lost and have to be predictable, with the knowledge of the previously scanned values and the bit-planes in clear. Therefore, the

Algorithm 1: Recursive function to process each bit-plane $I^{[k]}$.

BP_processing *(k, K_e, $I^{[\![k,7]\!]}$, marked)*
/* k is the index of the current bit-plane; */
/* K_e is the encryption key; */
/* $I^{[\![k,7]\!]}$ is the original image, restricted to its $8-k$ least-significant bits; */
/* *marked* is a byte indicating which bit-planes can be marked */
begin
  if $k < 7$ **then**
    PE_map $\leftarrow$ ***PE_detection*** $(I^{[\![k,7]\!]})$;
    PE_info $\leftarrow$ ***PE_information_calculation*** *(PE_map)*;
    $I_e^{[k]} \leftarrow$ ***BP_encryption*** $(I^{[k]},\ K_e)$;
    if *size(PE_info)* $<$ *size*$(I_e^{[k]})$ **then**
      marked $\leftarrow$ *marked* $+ 2^{7-k}$;
      $I_{e\_pe}^{[k]} \leftarrow$ ***Bit_substitution*** $(I_e^{[k]}$, *PE_info*$)$;
      BP_processing $(k+1,\ K_e,\ I^{[\![k+1,7]\!]}$, *marked*$)$;
    else
      $I_{e\_pe}^{[k]} \leftarrow I_e^{[k]}$;
      for $i = k+1$ **to** $i = 7$ **do**
        $I_e^{[i]} \leftarrow$ ***BP_encryption*** $(I^{[i]},\ K_e)$;
        $I_{e\_pe}^{[i]} \leftarrow I_e^{[i]}$;
  else if $k = 7$ **then**
    $I_e^{[k]} \leftarrow$ ***BP_encryption*** $(I^{[k]},\ K_e)$;
    $I_{e\_pe}^{[k]} \leftarrow I_e^{[k]}$;

first step of our method consists in analyzing the original bit-plane $I^{[k]}$ content in order to detect the PE:

1. Let $I^{[\![k,7]\!]}$ be the clear image, restricted to its $8-k$ least-significant bits.
2. Consider $p^{[\![k,7]\!]}(i,j)$ as an element of $I^{[\![k,7]\!]}$, and its inverse $inv^{[\![k,7]\!]}(i,j)$, such as:
$$inv^{[\![k,7]\!]}(i,j) = (p^{[\![k,7]\!]}(i,j) + 2^{7-k}) \mod 2^{8-k}. \quad (1)$$
Actually, the value of $inv^{[\![k,7]\!]}(i,j)$ corresponds to the original value of $p^{[\![k,7]\!]}(i,j)$ with the inverse value of the most significant bit $p^{[k]}(i,j)$.
3. Calculate the absolute difference between each of these two values with the two neighbors $p^{[\![k,7]\!]}(i-1,j)$ and $p^{[\![k,7]\!]}(i,j-1)$. The smallest value gives the best predictor of $p^{[\![k,7]\!]}(i,j)$ for the decoding step. Record these differences as Δ and Δ^{inv}:
$$\begin{cases} \Delta & = \min(\left|p^{[\![k,7]\!]}(i,j) - p^{[\![k,7]\!]}(i-1,j)\right|, \\ & \quad \left|p^{[\![k,7]\!]}(i,j) - p^{[\![k,7]\!]}(i,j-1)\right|), \\ \Delta^{inv} & = \min(\left|inv^{[\![k,7]\!]}(i,j) - p^{[\![k,7]\!]}(i-1,j)\right|, \\ & \quad \left|inv^{[\![k,7]\!]}(i,j) - p^{[\![k,7]\!]}(i,j-1)\right|). \end{cases} \quad (2)$$
4. If $\Delta < \Delta^{inv}$ then, there is no PE since the original value of $p^{[\![k,7]\!]}(i,j)$ is closer to its predictor than the inverse value. Or else, there is an error and it must be pointed out as an error location.

Therefore, the PE location information is computed as following:

1. Sequences of b bits are considered and scanned.
2. If a sequence contains at least one PE, it is surrounded by two flags of b bits. In this case, the previous and following sequences are filled with bits to '1'.
3. In the current sequence, a PE location is highlighted by a '1'. If there is no error, there is a '0'.

If the size of the PE location information is smaller than the embedding capacity, it is stored by substitution in the current bit-plane after encryption. Conversely, as soon as a bit-plane cannot be marked, the PE highlighting process is interrupted and the byte *marked* contains this information. The remaining bit-planes are just encrypted, as described in Section 2.1.2.

Note that this mechanism allows us to achieve perfect reversibility during the decoding phase, without causing overflow.

2.1.2 Bit-plane encryption

First of all, the encryption key K_e is used as a seed for a pseudo-random number generator to obtain a pseudo-random sequence of $m \times n$ bits $s(i,j)$. Then, for a bit-plane $I^{[k]}$ of the original image, each bit is XOR-ed with the associated bit in the pseudo-random sequence to generate an encrypted bit $p_e^{[k]}(i,j)$ of the encrypted bit-plane $I_e^{[k]}$:
$$p_e^{[k]}(i,j) = s(i,j) \oplus p^{[k]}(i,j). \quad (3)$$
After encryption, if the encrypted bit-plane can be marked, the PE location information is inserted. With the byte *marked*, the data hider knows if he can embed bits of the secret message. As explained in Section 2.1.1, flag and error sequences serve to highlight the PE and are embedded by bit substitution. In a given PE location, bits of the encrypted bit-plane are therefore replaced, according to the result of the PE highlighting process. At the end of the process, the to-be-marked encrypted bit-plane $I_{e\_pe}^{[k]}$ is obtained.

2.2. On the data hiding side

The data embedding step can be completed directly in the encrypted domain, and without knowing the encryption key K_e used for the encryption step, as shown in the data hiding side in Fig. 1.

Firstly, the data hider checks if the current bit-plane $I_{e\_pe}^{[k]}$ can be marked by bits of the secret message, according to the bit of index k of the byte *marked*. After that, the data hiding key K_{dh} is used to encrypt the secret message. This way, it is not possible to detect its presence after embedding.

By following the S-order, the error location information, inserted by the owner of the original image, is therefore observed, and all the bits $p^{[k]}_{\text{e\_pe}}(i,j)$, which are not part of a flag or of an error sequence, can be used for data hiding. These available bits are substituted by bits b^l (with $l < L$, the number of bits which can be marked). Bits $p^{[k]}_{\text{e\_dh}}(i,j)$ of the marked encrypted bit-plane $I^{[k]}_{\text{e\_dh}}$ are thus obtained:

$$p^{[k]}_{\text{e\_dh}}(i,j) = b^l. \tag{4}$$

2.3. Decoding phase

During the decoding phase, there are three possible scenarios, depending if the recipient: only knows the data hiding key K_{dh}, only knows the encryption key K_{e}, or knows both keys K_{dh} and K_{e}.

As a reminder, the marked encrypted image $I_{\text{e\_dh}}$, of $m \times n$ pixels, consists of eight bit-planes:

- some significant bit-planes are marked by bits of the secret message and the PE location information.
- the remaining bit-planes are just encrypted.

Similarly to the process during the encoding phase, the eight bit-planes of $I_{\text{e\_dh}}$ are processed recursively, but in reverse order because the $(k+1)^{\text{th}}$ to 7^{th} bit-planes are used during the prediction of the k^{th} bit-plane. In any case, whatever key he knows, the recipient has to observe if the current bit-plane is marked, or simply encrypted, according to the bit of index k of the byte *marked*. On one hand, if the recipient knows the encryption key K_{e}, he generates the pseudo-random sequence of $8 \times m \times n$ bits and decrypts the only encrypted bit-planes. On the other hand, if the recipient knows the data-hiding key K_{dh} only, he just skips these planes.

Fig. 3 presents an overview of the decoding phase for each marked encrypted bit-plane $I^{[k]}_{\text{e\_dh}}$. If the recipient knows the data hiding key K_{dh}, he can easily extract the secret message from the bit-plane. Actually, he just needs to scan the marked encrypted bit-plane in the S-order and extract bits by considering the inserted error location information. Finally, the extracted bits of the message can be decrypted with the key K_{dh}.

If the recipient knows the encryption key K_{e}, he scans the bit-plane $I^{[k]}$ in the S-order, and each bit $p^{[k]}(i,j)$ is predicted according to its previously reconstructed neighbors and using $I^{[k+1,7]}$. Indeed, as illustrated in Fig. 3, the $(k+1)^{\text{th}}$ to 7^{th} bit-planes of the original image are necessary for the prediction:

1. Let $p^{[\![k+1,7]\!]}(i,j)$ be the current pixel value of $I^{[k+1,7]}$. In order to reconstruct $p^{[k]}(i,j)$, we consider the two possible values for $p^{[\![k,7]\!]}(i,j)$: $p^{[\![k,7]\!]}(i,j)^0 = p^{[\![k+1,7]\!]}(i,j)$ and $p^{[\![k,7]\!]}(i,j)^1 = p^{[\![k+1,7]\!]}(i,j) + 2^{7-k}$.

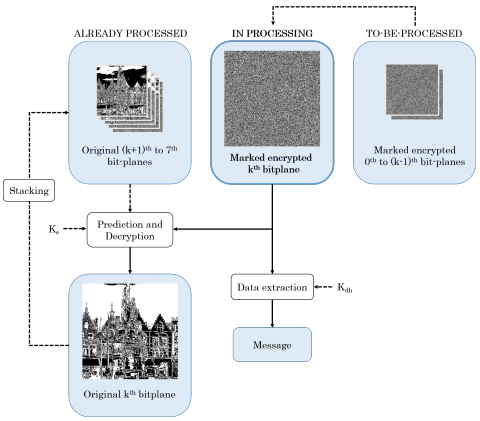


Figure 3. Overview of the decoding phase for a marked encrypted bit-plane $I^{[k]}_{\text{e\_dh}}$.

2. Compute the absolute difference between each of these two values with $p^{[\![k,7]\!]}(i-1,j)$ and $p^{[\![k,7]\!]}(i,j-1)$:

$$\begin{cases} \Delta^0 = \min(\left|p^{[\![k,7]\!]}(i,j)^0 - p^{[\![k,7]\!]}(i-1,j)\right|, \\ \qquad\qquad \left|p^{[\![k,7]\!]}(i,j)^0 - p^{[\![k,7]\!]}(i,j-1)\right|), \\ \Delta^1 = \min(\left|p^{[\![k,7]\!]}(i,j)^1 - p^{[\![k,7]\!]}(i-1,j)\right|, \\ \qquad\qquad \left|p^{[\![k,7]\!]}(i,j)^1 - p^{[\![k,7]\!]}(i,j-1)\right|). \end{cases} \tag{5}$$

3. If the current location does not correspond to a PE, we have:

$$p^{[k]}(i,j) = \begin{cases} 0, \text{ if } \Delta^0 < \Delta^1, \\ 1, \text{ else.} \end{cases} \tag{6}$$

However, if it is concerned by a PE, we have to perform the inverse prediction:

$$p^{[k]}(i,j) = \begin{cases} 1, \text{ if } \Delta^0 < \Delta^1, \\ 0, \text{ else.} \end{cases} \tag{7}$$

Without the encryption key K_{e}, the prediction mechanism fails. Actually, note that the $(k+1)^{\text{th}}$ to 7^{th} bit-planes of the original image have to be recovered in order to predict the value of the current bit-plane until the MSB-plane.

3. Experimental results

In this section, we present results obtained by applying our EPE-based HC-RDHEI method, using flags with $b = 8$ bits. For RDHEI approaches, we are interested in finding the best trade-off between the number of erroneous extracted bits of the message (BER), the payload (also called embedding rate, expressed in *bpp*) and the reconstructed image quality after data extraction (in terms of PSNR and SSIM). As our proposed method allows perfect reversibility (PSNR $\rightarrow +\infty$, SSIM $= 1$), and message extraction without error (BER$= 0$), we aim to have the largest possible payload.

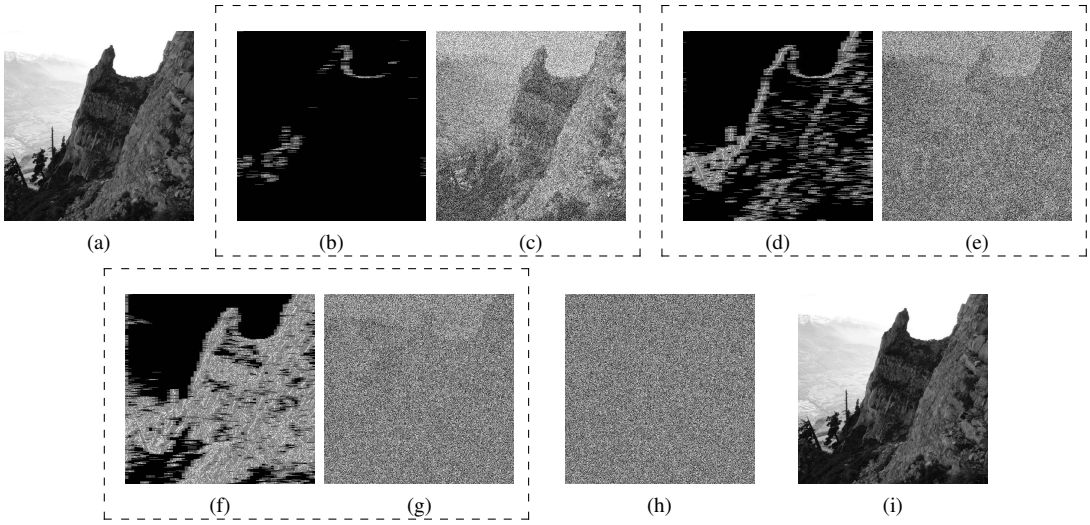
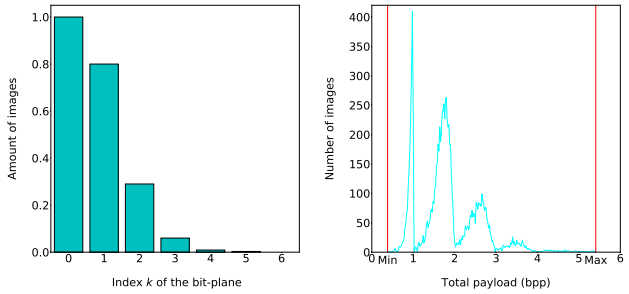


Figure 4. Illustration of our proposed EPE-based HC-RDHEI method: (a) Original *Mountain* image of 512×512 pixels, (b) PE location information on the 1^{st} bit-plane (MSB-plane), number of errors $=$ 358 (0.1%), (c) Original image with only the MSB-plane which is marked encrypted, payload $=$ 0.975 *bpp*, (d) PE location information on the 2^{nd} bit-plane, number of errors $=$ 4252 (1.6%), (e) Original image with only the 1^{st} and the 2^{nd} bit-planes which are marked encrypted, payload $=$ 1.754 *bpp*, (f) PE location information on the 3^{rd} bit-plane, number of errors $=$ $24,733$ (9.4%), (g) Original image with only the 1^{st} to 3^{rd} bit-planes which are marked encrypted, payload $=$ 2.156 *bpp*, (h) Image after encryption using all bit-planes and data embedding in the 1^{st} to 3^{rd} bit-planes, (i) Reconstructed image I, (PSNR $\rightarrow$ $+\infty$, SSIM = 1).

First, we applied our method to the original *Mountain* image of 512×512 pixels, from the BOWS-2 database [1], illustrated in Fig. 4.a. During the encoding phase, bit-planes of the original image are processed separately. The first step consists of detecting PE on the 1^{st} bit-plane (MSB-plane). Fig. 4.b illustrates the location of the bits which cannot be marked on this bit-plane. Looking at the white areas, we can see the bits which cannot be predicted using the neighboring pixel values and, in grey, the bits used as flags to highlight the PE. Note that the number of PE is very low (0.1%), because neighboring pixels are strongly similar. In Fig. 4.c, the MSB-plane of the original image has been encrypted and then, marked with bits of the secret message according to the PE location. The payload of the obtained image is equal to 0.975 *bpp*, which is already high. In order to be able to reconstruct the original image without error during the decoding phase, 0.025 *bpp* of the image is then used to highlight PE. Note that, after this step, seven other bit-planes are still in the clear domain. After the 1^{st} bit-plane processing, we repeat the same process on the 2^{nd} bit-plane (PE detection, PE location information calculation, encryption and data embedding). In Fig. 4.d, we can see the unmarkable bits on this bit-plane. There are more PE than for the MSB-plane, but the error rate is still low (1.6%). Actually, bits of the 2^{nd} bit-plane remain highly correlated in natural images. Fig. 4.e corresponds to the image obtained from Fig. 4.c, by encrypting and then marking bits of the 2^{nd} bit-plane. The payload is then equal to 1.754 *bpp*, which indicates a gain of 0.779 *bpp*. Note that, in this case, the loss due to the PE location information embedding is equal to 0.246 *bpp*. Similarly, after the 1^{st} and 2^{nd} bit-planes processing, we apply the encoding method on the 3^{rd} bit-plane. Fig. 4.f presents the unmarkable bits on this bit-plane (error rate equal to 9.4%), and Fig. 4.g is the image obtained from Fig. 4.e, by encrypting and then marking bits of the 3^{rd} bit-plane. The payload is also equal to 2.156 *bpp*, which indicates a gain of 0.402 *bpp*. Note that, in this case, the loss due to the PE location information embedding is equal to 0.598 *bpp*. Therefore, the PE highlighting process is interrupted, because the PE location information of the 4^{th} bit-plane is too large and cannot be stored by substitution. Consequently, 4^{th} to 8^{th} bit-planes are just encrypted. In Fig. 4.h, we present the final marked encrypted image, where all bit-planes are encrypted and the 1^{st} to 3^{rd} bit-planes are also marked. Finally, in Fig. 4.i, we can see that our method achieves exact reversibility. Actually, during the decoding phase, if the recipient has the encryption key, they can perfectly reconstruct the original image (PSNR $\rightarrow$ $+\infty$ and SSIM $=$ 1 between Fig. 4.a and Fig. 4.i).

We have applied our proposed method to the BOWS-2 database [1], which is composed of 10,000 images of 512×512 pixels with different statistical properties. Fig 5 presents the results we obtained from this database using flags of $b = 8$ bits to highlight the PE. In Fig. 5.a, we can see that data embedding can be achieved in the MSB-plane for all images, until the 2^{nd} bit-plane for 80% of the images, until the 3^{rd} bit-plane for 29% of the images, and until the 4^{th} bit-plane for 6% of the images. However, data hiding in the less significant bit-planes is rare (less than 1% of

the images, for $k > 3$). Fig. 5.b and Fig. 5.c represent the image repartition as a function of the payload value. In accordance with the Fig. 5.a showing that the data embedding in the 1st and 2nd is possible for most of the images, the first quartile value is equal to 1.431 *bpp*. Conforming to the median value, a payload larger than 1.749 *bpp* is achieved for half of the images. Note that the average payload value is 1.836 *bpp*, which is close to the median value. Finally, for a quarter of the images, the payload is larger than 2.305 *bpp*. This means that the data embedding phase is completed at least in the first three most significant bit-planes of these images. Moreover, the maximal payload value is also equal to 5.408 *bpp*, which indicates that the first six most significant bit-planes of the image are marked with bits of the message.



(a) (b)

1st quartile 25%	Median 50%	3rd quartile 75%
1.431 *bpp*	1.749 *bpp*	2.305 *bpp*

(c)

Figure 5. Performance measurements on the BOWS-2 database [1], using our proposed EPE-based HC-RDHEI method: (a) Amount of images where the data embedding phase can be achieved depending of the index k of the bit-plane (with $0 \leq k \leq 6$), (b) Image repartition according to the payload value (in *bpp*) , (c) Repartition by quartile of the results presented in b).

We also made comparisons, in terms of embedding rate and reconstructed image quality (using the encryption key K_e only), between our method and four recent state-of-the-art approaches proposed by Puteaux and Puech [6], Zhang *et al.* [10], Cao *et al.* [2] and Wu and Sun [8]. In Fig. 6, we present the results obtained using the *Lena* image. Firstly, we can see that only our proposed method allows us to embed more than 1 *bpp*. Indeed, with Zhang *et al.*, Cao *et al.* and the Wu and Sun' methods, the payload is smaller than 0.75 *bpp* and, with the approach described by Puteaux and Puech [6], the maximal payload is 1 *bpp*. Note that, with our proposed method, it is possible to mark the 1st to 3rd bit-planes in the Lena image. Indeed, the payload is equal to 1.935 *bpp*, when the three bit-planes are used for data embedding. Furthermore, by examining the reconstructed image quality, we can see that the methods of Zhang *et al.* [10] and Cao *et al.* [2] do not achieve perfect reversibility, contrary to the approach of Wu and Sun [8], Puteaux and Puech [6], and ours, which is indicated by a PSNR which tends towards infinity. In conclusion, in addition to being error-free during the secret message extraction, our method obtains better results than other current state-of-the-art approaches. It achieves a very good trade-off between the embedding rate and reconstructed image quality.

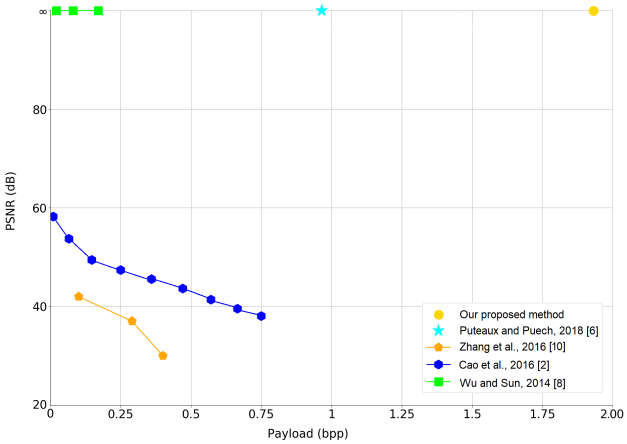


Figure 6. Performance comparison between our proposed method and previous work for the *Lena* image.

4. Conclusion

In this paper, we proposed a recursive huge-capacity reversible data hiding method in the encrypted domain, which fully exploits the correlation between neighboring pixels in clear images. As demonstrated by Puteaux and Puech in [6], the first MSB value of a pixel can be predicted according to previously decrypted pixels. Actually, we have shown, in this EPE-based method, that other bit-planes are also strongly correlated, depending on the clear image content. Starting from the MSB-plane, each bit-plane of the original image is analyzed in order to highlight the prediction errors, and then encrypted. If the amount of prediction errors is acceptable, most bits of the current bit-plane are substituted by bits of the secret message and the next bit-plane is also analyzed. Otherwise, the data hiding process is interrupted. Finally, the payload is very high, on average 1.836 *bpp*, which indicates a gain of 0.868 *bpp* compared to the use of the MSB-plane only. Moreover, the original image can be perfectly recovered and the secret message is extracted without error.

In future work, we are searching for a new prediction error mechanism in order to enhance the embedding capacity. We are also developing the use of flags of different sizes ($b^{[k]}$ bits, with $0 \leq k \leq 6$) during the calculation of the PE location information, depending on the PE location configuration in the current bit-plane. Actually, there is a real trade-off between the probability of bad detection of a flag, and the loss of embedding capacity, which deserves to be further analyzed in detail.

References

[1] P. Bas and T. Furon. Image database of BOWS-2. http://bows2.ec-lille.fr/. 5, 6

[2] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo. High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Transactions on Cybernetics*, 46(5):1132–1143, 2016. 1, 6

[3] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni. Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing. *EURASIP Journal on Information Security*, 2007:17, 2007. 1

[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. 1

[5] W. Puech, M. Chaumont, and O. Strauss. A reversible data hiding method for encrypted images. In *Electronic Imaging 2008*, pages 68191E–1–68191E–9. International Society for Optics and Photonics, 2008. 1

[6] P. Puteaux and W. Puech. An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Transactions on Information Forensics and Security*, 13(7):1670–1681, 2018. 1, 6

[7] H.-T. Wu, Y.-M. Cheung, and J. Huang. Reversible data hiding in Paillier cryptosystem. *Journal of Visual Communication and Image Representation*, 40:765–771, 2016. 1

[8] X. Wu and W. Sun. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Processing*, 104:387–400, 2014. 1, 6

[9] X. Zhang. Separable reversible data hiding in encrypted image. *IEEE Transactions on Information Forensics and Security*, 7(2):826–832, 2012. 1

[10] X. Zhang, J. Long, Z. Wang, and H. Cheng. Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1622–1631, 2016. 1, 6