A Recursive Reversible Data Hiding in Encrypted Images Method With a Very High Payload

Pauline Puteaux, William Puech

A Recursive Reversible Data Hiding in Encrypted Images Method with a Very High Payload

Pauline Puteaux, *Student Member, IEEE* and William Puech, *Senior Member, IEEE*

***Abstract*—Reversible data hiding in encrypted images (RDHEI) can be used as an effective technique to embed additional data directly in the encrypted domain and therefore, without any invasion to privacy. In this way, RDHEI is especially useful for labeling encrypted images in cloud storage. In this paper, we propose a new method of data hiding in encrypted images, which is fully reversible and has a very high payload. All the bit-planes of an image are processed recursively, from the most significant one to the least significant by combining error prediction, reversible adaptation, encryption and embedding. For pixel prediction, the Median Edge Detector, also called LOCO-I and known to be efficient in JPEG-LS compression standard, is used for each bit-plane. Moreover, conversely to current state-of-the-art methods, in our proposed method, there is no pre-processing step to correct incorrectly predicted pixels and no flags to highlight them. Indeed, a reversible adaptation of the bit-planes is performed in order to make it possible to detect and correct all incorrectly predicted pixels during the decoding step. Thanks to the high correlation between pixels in the clear domain, a large part of the bits of an image can be substituted by bits of a secret message. Our experiments show that we can generally embed bits of the secret message until the fourth most-significant bit-plane of an image, this allows us to have an average payload value of 2.4586 *bpp*.**

***Index Terms*—Image security, image encryption, reversible data hiding, recursive process, bit-plane prediction, signal processing in the encrypted domain.**

I. Introduction

In recent years, with the development of cloud computing, more and more users upload their personal data to a remote or cloud server. However, this procedure leads to serious security leaks, where confidentiality, authentication and integrity are constantly threatened. In order to deal with these problems, data hiding and/or encryption can be used for multimedia security, depending on the requirements of the user.

Data hiding (DH) aims to embed a secret message into a clear image, by modifying its content. It is particularly suitable for authentication and data enrichment. Moreover, during the decoding step, after data extraction, the original image can be recovered without loss of quality or error. In this specific last case, the term "reversible data hiding" (RDH) is used. Methods are mainly based on lossless compression appending [6], difference expansion [9], [23], [24], histogram shifting [7], [12], [22], integer transform [1], [14] or prediction error (PE) consideration [13], [21].

For privacy protection, clear images are typically encrypted before they are uploaded to the cloud. Indeed, selective or full encryption can provide visual confidentiality and prevents an unauthorized person from accessing the original image content [25]. During the transmission or archiving of these encrypted images, it may be interesting to be able to analyze or process them directly in the encrypted domain, and that without knowing the encryption key or the original image content. Recently, different image processing methods in the encrypted domain have been developed because of a growing interest from the scientific community [5]. The main applications are visual secret sharing, research and indexing in encrypted databases, recompression of crypto-compressed images and reversible data hiding in encrypted images (RDHEI).

For authentication, data enrichment or retrieval purposes in the confidential domain, reversible data hiding in encrypted images (RDHEI) is effective. In recent years, many state-of-the-art methods have been designed. In these kind of methods, an image owner performs the encryption of the image to guarantee its content privacy, and uploads it on to a cloud server. The server has then no access to the original image content and embeds the secret message – labels for example – directly on the encrypted data without the encryption key [20]. The space to embed the message is vacated before [3], [10], [16]–[18], [31] or after [8], [15], [19], [29], [30] the encryption phase and, during the decoding phase, image reconstruction and data extraction are processed at the same time [15] or separately [4], [26]–[28], [30]. There exists a trade-off between the payload, the number of erroneous extracted bits of the message and the reconstructed image quality with respect to the original image. Moreover, previous methods have shortcomings such as low payloads, errors in data extraction or during the image reconstruction, or poor image quality using high payloads.

In this paper, we present a recursive method for RDHEI, which allows us to achieve a very high payload. Based on the fact that pixels are highly correlated in the clear domain, bit-planes should be predictable. Therefore, we propose to use a maximum amount of bit-planes in an image for data embedding. Starting from the MSB-plane, each bit-plane is processed recursively, until the second LSB-plane as a function of the image content. The first step is the PE consideration which consists of analyzing the content of an image in order to identify all the pixels which cannot be predicted according to their neighbors. A PE location map and a PE value list are then computed according to the location of these pixels and their associated PE values. Then, the size of the PE value list is analyzed in order to know if PE values can be embedded in the current bit-plane. If embedding is possible, the image composed of the current bit-plane and all further bit-planes is then adapted to highlight PE by drawing some specific configurations. Note that this adaptation is fundamental to be able to perfectly reconstruct the original image during the decoding phase. The current bit-plane is then separately processed. Indeed, this current bit-plane is encrypted, marked by PE values and finally, by bits of the secret message to embed. The same steps are repeated to recursively process the next bit-planes. If the PE value list cannot be embedded, then this analysis is ended and,

P. Puteaux and W. Puech are with the LIRMM, CNRS, Univ. Montpellier, France (e-mail: pauline.puteaux@lirmm.fr; william.puech@lirmm.fr).

in this case, the current and all the remaining bit-planes are only encrypted. During the decoding phase, bit-planes are processed in the reverse order, from the LSB-plane to the MSB-plane. The secret message is extracted without error and the eight bit-planes of the original image can be losslessly recovered by using previously reconstructed bit-planes and prediction. Indeed, all adapted pixels can be located by identifying the specific configurations and corrected using the PE value list. According to our obtained results, the possible payload is very high and much higher than previous state-of-the-art methods. Indeed, on average using 10,000 images, we can embed almost 2.46 bpp per image. Note that, unlike the methods presented in [17] and [18], which are smart extensions of the method proposed in [16] on several or all the possible bit-planes, this new method is neither an extension, nor a combination of the proposed approaches presented in [16], [17] or [18].

The main contributions and key-points of our proposed paper can be then summarized as follows:

- Our proposed method is neither an extension or a combination of previous state-of-the-art approaches.
- In our new MSB-based RDHEI method, all the bit-planes of an image are processed recursively, from the most significant one to the least significant by combining error prediction, adaptation, encryption and embedding, as long as it is possible.
- We have successfully implemented the use of the Median Edge Detector (MED) predictor also called LOCO-I from JPEG-LS (which is a very powerful compression encoder). As a result, the number of errors decreases and then, the obtained payload is more significant.
- Contrary to previous methods of the state-of-the-art, there is no pre-processing step to correct incorrectly predicted pixels. This allows us to achieve perfect reversibility.
- Moreover, no flags are used to highlight incorrectly predicted pixels. Instead, a reversible adaptation of the bit-planes is performed to make it possible to detect and correct all incorrectly predicted pixels during the decoding step. Moreover, from a security point of view, the new proposed approach in our paper is more robust to a statistical analysis than methods using flags.

The rest of this paper is organized as follows. Section II describes current modern methods using RDHEI, in particular recent schemes with high payload. The proposed method is then described in detail in Section III. Section IV reports the experimental results and comparisons with related work. Finally, this paper is concluded in Section V.

II. Related work

Methods of reversible data hiding in encrypted images (RDHEI) are effective to embed data in the encrypted domain, without knowing the secret key used during the encryption of the image or the content of the original image. They are mostly used for image annotation or authentication purposes, using labels, timestamps or a message about the origins of the image such as EXIF data. In these kinds of schemes, the owner of the image and the data hider are not necessary the same party, such as in a cloud scenario. In fact, encryption is completed by the content owner to protect image privacy and then, the encrypted image is sent over a network or uploaded on to a cloud server. Therefore, the server – considered as a data hider – has no access to the original image content and has to perform the secret message embedding directly on the encrypted data. In a cloud scenario [20], labels attached inside the encrypted image can provide better management for administrators. During the decoding phase, the secret message has to be extracted without error. In addition, when an authorized user downloads the marked encrypted image from the cloud, the original content has to be losslessly recovered after image decryption. Therefore, RDHEI methods provide an alternative way to traditional systems of file management, by accommodating additional information inside the encrypted image itself, instead of using a metadata file. For this purpose, many state-of-the-art methods have been proposed, in order to achieve the best trade-off between payload (expressed in bpp), the number of erroneous extracted bits of the message, and the reconstructed image quality in comparison with the expected original image (in terms of PSNR, in dB, or SSIM).

State-of-the-art methods can be divided into two categories, these are Reserving Room Before Encryption (RRBE) and Vacating Room After Encryption (VRAE). On one hand, in RRBE methods, the original image is pre-processed by the content owner before encryption to release some space to embed data. It is then encrypted and the data hider can embed bits of the secret message into specific positions [3], [10], [16], [31]. On the other hand, in VRAE methods, the original image content is blindly encrypted by the content owner, and the data hider modifies the encrypted data in order to hide bits of the secret message [8], [15], [19], [29], [30]. These two different approaches are effective, but have some limitations. In RRBE methods, the achieved payload can be higher, but a pre-processing phase before encryption is needed. This can be a problem and unpractical if the content owner does not know that the encrypted image has to be analyzed or processed later. In VRAE methods, the recipient of the marked encrypted image has to estimate the original image content to reconstruct it. Therefore, the recovered image is an estimation of the original image and perfect reversibility cannot be achieved. Moreover, in order to minimize the introduced distortion, a large payload cannot be used. Furthermore, during the decoding phase, message extraction and image reconstruction can be performed at the same time, or separately. In this last case, the secret message is extracted and the original image is recovered independently. Ma *et al.* were the first to propose a RRBE technique in [10]. Using a traditional method of RDH based on histogram shifting, they release some space in a first region of the original image by embedding LSB of some pixels into another region. The pre-processed image is then encrypted and transmitted to a network or to a cloud where a data hider can be. Therefore, the location of some LSB of the first region in the encrypted image are free and can be used to embed data of the secret message. Note that, not only the first LSB, but also the second and the third bits of each pixel which are selected for data embedding, can be used. The total payload of the marked encrypted image can reach 0.5 bpp. Consequently, in

comparison with previous state-of-the-art schemes, the amount of embedded data is more than ten times larger. Cao *et al.* suggested using patch-level representation to release a large amount of space to embed data [3]. The original image is represented by sparse coefficients, according to a dictionary. Moreover, the reconstructed residual errors are encoded and then self-embedded by using a RDH algorithm in the image. The encryption is then performed and the dictionary must be embedded in the encrypted image by the content owner. The data hider firstly recovers the position of the first room preserving patch and the room size for each patch. Then, he substitutes each bit of the encrypted image which can be marked by a bit of the secret message. Note that using this sparse coding representation, the average payload is about 1 bpp, but the reconstructed image is altered, as indicated by PSNR between 40 and 50 dB. Zhang *et al.* described two methods of data hiding for images which have been encrypted using a public-key cryptosystem with probabilistic and homomorphic properties. The first method is reversible and the second is lossless [31]. In the reversible scheme, the original image is firstly pre-processed to shrink its histogram. It is then encrypted with an additive homomorphic cryptosystem. After the encryption phase, the encrypted image is marked with bits of the secret message and error-correction codes. Note that the data embedding phase does not introduce overflow or underflow in the directly decrypted image, due to the histogram shrink. During the decoding phase, the original image can be reconstructed and the secret message can be extracted separately and losslessly. In the lossless scheme, the original image is directly encrypted without modification. The data embedding is then performed in the encrypted domain using multilayer wet paper coding. A large amount of information can be embedded (payload $\sim$ 0.5 bpp), without affecting the decryption and reconstruction of the original image.

In this paper, particular attention is drawn to high payload RDHEI methods. In fact, we aim to design a method that achieves perfect reversibility and a lossless message extraction, while embedding a large amount of data (*i.e.* with high payload). We then describe the most recent state-of-the-art methods which are motivated by the same objectives. In 2018, Puteaux and Puech proposed to use MSB values instead of LSB values to embed a secret message [16]. In fact, they stated that MSB substitution does not introduce artifacts in the encrypted domain and MSB prediction is easier than LSB prediction. Based on these two assumptions, in [16] they proposed two different high capacity reversible data hiding (HCRDH) approaches: "corrected prediction errors" (CPE) and "embedded prediction errors" (EPE). In both approaches, all pixels of the clear image which cannot be predicted according to their neighbors are identified. In the CPE-HCRDH approach, the original image is pre-processed to avoid all PE. The pre-processed image is encrypted and then the data hider can blindly replace all MSB values of the encrypted image by bits of the secret message. In this case, payload is equal to 1 bpp and the reconstructed image corresponds to the pre-processed image which is very close to the original one (PSNR > 50 dB). In the EPE-HCRDH approach, the original image is encrypted without any modification. After encryption, information about the location of all pixels which cannot be predicted is embedded by MSB substitution by the content owner. Then, the data hider can detect all bits which can be marked and replaces them by bits of the secret message. In this case, the payload is slightly less than 1 bpp but perfect reversibility is achieved. In [18], Puyang *et al.* proposed an extension of the EPE-HCRDH approach in [16]. They suggested using another predictor, which is more efficient. Moreover, they explained that the second MSB-plane can also be used for data embedding, in addition to the first one. As a result, the achieved payload is equal to 1.35 bpp on average. Puteaux and Puech also improved the EPE-HCRDH approach proposed in [16] by using iteratively all bit-planes of the image, from MSB to LSB as long as it is possible [17]. According to their results, the payload is significantly higher than the value obtained in [16], and in [18], with 1.84 bpp on average, while preserving a full reversibility. Yi *et al.* proposed a parametric binary tree labeling (PBTL) method, where spatial correlation between pixels is kept in the encrypted domain within small blocks [26]. Some pixels are used as reference values to compute prediction errors and PBTL serves to highlight these prediction errors. For data hiding, bits of a secret message are embedded using bit substitution by exploiting spatial redundancy. The achieved average payload is therefore in the order of 2 bpp. In [4], Chen and Chang designed a block-based MSB plane rearrangement (BMPR), which is an effective way to transform MSB-planes of an original image into a highly compressible bitstream using an extended run-length coding. Using this rearrangement and compression mechanism, they can efficiently generate room for high payload embedding. During the decoding phase, a receiver can extract the secret data directly from encrypted images with only the data hiding key or the high-quality marked image with only the encryption key. Note that knowing the two keys, the original image can be losslessly recovered.

III. Proposed recursive RDHEI method

Previous methods presented in Section II have shortcomings, such as low payloads, errors in data extraction or during the original image reconstruction, or poor image quality using high payloads. In particular, in the CPE-HCRDH approach [16], the image pre-processing step significantly alters the original image, which cannot be perfectly recovered during the reconstruction step. Indeed, adapted pixels cannot be located and corrected. Moreover, in the EPE-HCRDH approach [16], flags are used to highlight the prediction error (PE) location. In addition to decreasing the payload value, this can be a security issue and, in some cases, some flags can be badly detected. Furthermore, in these two approaches, the maximal payload is equal to 1 bpp, because the secret message is only embedded in the MSB-plane. This last drawback is addressed in [18] and [17]. Addressing this issue, Puyang *et al.* proposed to use the first and the second MSB-planes to embed the secret message [18]. However, Puteaux and Puech suggested to iteratively process all bit-planes of the image, in order to fully exploit the redundancy in images [17]. But, even if these methods are smart extensions of the EPE-HCRDH approach, flags are still present and necessary for the PE highlighting process.

In this section, we describe a new method which strongly reduces these shortcomings. For this, we introduce a recursive method of reversible data hiding in encrypted images with a very high payload. During the reconstruction of an original image, the most significant bit-planes cannot be recovered without having to reconstruct all the least significant bit-planes beforehand. This new approach is perfectly reversible and has a very high payload due to the fact that our method is fully recursive. Conversely to [18] and [17], our method is neither an extension of the CPE-HCRDH or of the EPE-HCRDH approaches presented in [16], nor a combination of these two approaches. Indeed, with this new proposed method, there is no pre-processing step to correct incorrectly predicted pixels and no flags to highlight them. Conversely, an adaptation of the bit-planes is performed in order to draw several specific configurations to make it possible to detect and correct all of the incorrectly predicted pixels during the decoding step.

Firstly, in Section III-A, we give an overview of the encoding method, which involves the image content owner and the data hider. As presented in Section III-B, on the content-owner side, the original image is recursively processed, by scanning all the bit-planes from the most significant to the least significant one. For each bit-plane, this includes four main steps, which are: 1) a PE consideration using the Median Edge Detector as a predictor [11]; 2) a reversible adaptation of the bit-planes; 3) a current bit-plane encryption; 4) a PE value list embedding in the encrypted bit-plane. At the end of this stage, the fully encrypted image after embedding of PE values is obtained and transmitted to a network or to a cloud where a data hider can be. Therefore, the data embedding is described in Section III-C. On this side, the secret message is embedded by bit substitution in each encrypted bit-plane which can be marked to obtain the final marked encrypted image. Finally, we give details on the decoding phase in Section III-D. Since the method is separable, the secret message can be extracted and the original image can be reconstructed separately, without any error.

A. *Overview of the proposed method*

In our proposed method, the original image I, with a size of $m \times n$ pixels encoded on 256 grey-levels, is considered as a stack of 8 bit-planes $I^{[k]}$, with $0 \leq k \leq 7$. From the original image I, then we have $I^{[0,7]}$. We note $I_k^{[k,7]}$ corresponding to the original image after k modifications. During the encoding phase, bit-planes are processed recursively, from the most significant $I^{[0]}$ (also called MSB-plane), to the least significant one $I^{[7]}$ (LSB-plane). Indeed, the image $I_k^{[k+1,7]}$, composed by the $7-k$ least significant bit-planes is necessary to process the current bit-plane $I_k^{[k]}$. An overview of the encoding phase, which first involves the content owner, and finally the data hider, is presented in Fig. 1.

For each current bit-plane $I_k^{[k]}$, as illustrated in Fig. 2, the first step of the recursive process consists of the prediction error (PE) consideration on the image $I_k^{[k,7]}$, composed by the $8-k$ least significant bit-planes of the currently processed image. A PE location map and a PE value list are built during this step. At the end, a test is realized in order to know if the PE value list can be embedded in the current bit-plane $I_k^{[k]}$, *i.e.* if

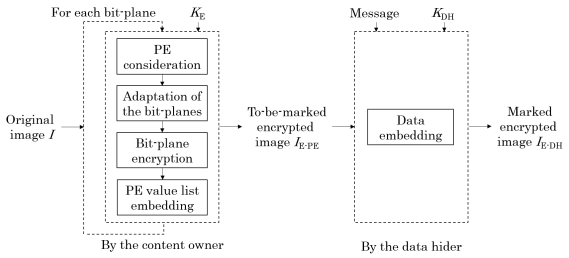


Fig. 1: Global overview of the encoding phase of our proposed recursive RDHEI method.

the size of the PE value list is smaller than the bit-plane size. If the PE value list can be embedded, the image $I_k^{[k,7]}$ is adapted in order to make the detection of the PE location possible, and finally the image $I_{k+1}^{[k,7]}$ is obtained. Note that in $I_{k+1}^{[k,7]}$, not only has the current plane been adapted, but all the bit-planes. After this step, the current bit-plane $I_{k+1}^{[k]}$, and only this one, is encrypted using the encryption key K_{E}. Then, the PE value list is embedded in the encrypted bit-plane by substituting the first bits as much as necessary in order to obtain the bit-plane $I_{\text{E-PE}}^{[k]}$. Therefore, all the previous steps are recursively repeated on the image $I_{k+1}^{[k+1,7]}$, obtained after adapting the image $I_k^{[k,7]}$ and processing separately the most significant bit-plane $I_{k+1}^{[k]}$. Conversely, if the PE value list cannot be embedded in the current bit-plane, the recursive process is ended. In this case, the $K = k$ first bit-planes (*i.e.* the encrypted bit-planes after embedding of the PE values) are then stacked in order to obtain the image $I_{\text{E-PE}}^{[0,K-1]}$. Otherwise, the current and all the remaining $8-K$ least significant bit-planes are directly and only encrypted using the encryption key K_{E} in order to obtain the encrypted image $I_{\mathrm{E}}^{[K,7]}$. Finally, the fully encrypted image after embedding of the PE values $I_{\text{E-PE}}$ consists of the stacking of $I_{\text{E-PE}}^{[0,K-1]}$ and $I_{\mathrm{E}}^{[K,7]}$.

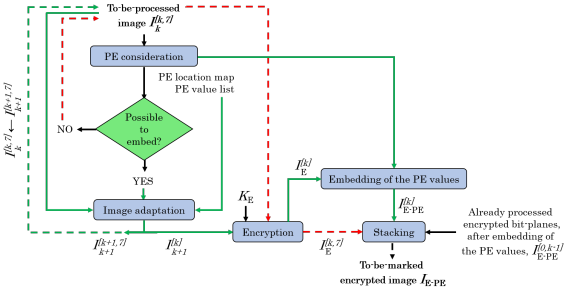


Fig. 2: Processing of the image $I_k^{[k,7]}$ during the encoding phase, on the content owner side.

On the data hiding side, the secret message is firstly encrypted using the data hiding key K_{DH}. The encrypted message is then embedded by bit substitution in each encrypted bit-plane which can be marked $I_{\text{E-PE}}^{[k]}$ with $0 \leq k < K$, after the PE value list. In this way, the final marked encrypted bit-plane $I_{\text{E-DH}}^{[k]}$ is obtained. Finally, as shown in Fig. 1, the marked encrypted image $I_{\text{E-DH}}$ is obtained and consists of eight encrypted bit-planes, where some are marked. Indeed, some significant bit-planes are marked by bits of the PE value list and of the secret message, and the remaining least significant bit-planes are only encrypted.

B. On the content owner side

On the content owner side, each current image $I_k^{[k,7]}$, composed by the most-significant bit-plane $I_k^{[k]}$ and the $7-k$ least significant bit-planes $I_k^{[k+1,7]}$, is processed recursively, as presented in Algorithm 1. The first step consists of PE consideration. During this step, the PE location map $\mathbf{L_{loc}^k}$ and the PE value list $\mathbf{L_{val}^k}$ are generated. Then, the algorithm calculates if it is possible to embed $\mathbf{L_{val}^k}$ in the current most significant bit-plane $I_k^{[k]}$. If the size of $\mathbf{L_{val}^k}$ is smaller than the bit-plane size, then the image $I_k^{[k,7]}$ is processed to generate the adapted image $I_{k+1}^{[k,7]}$, where the PE location can be identified according to the pixel values. After that, the most significant bit-plane $I_{k+1}^{[k]}$ is processed separately. This bit-plane is encrypted ($I_{\mathrm{E}}^{[k]}$), and then marked by bits of $\mathbf{L_{val}^k}$ ($I_{\text{E-PE}}^{[k]}$). The adapted image $I_{k+1}^{[k+1,7]}$ is then processed recursively. When the embedding of $\mathbf{L_{val}^k}$ is not possible, the current and all the remaining bit-planes are only encrypted and not marked. Note that the least significant bit-plane of the image cannot be predicted, but is needed for the PE consideration of all the other bit-planes. Consequently, in all cases, the LSB-plane is not marked and therefore, only encrypted. Section III-B1 describes the full error consideration mechanism. Section III-B2 deals with the necessary adaptation of the image to highlight the PE. Section III-B3 presents the bit-plane encryption scheme.

Algorithm 1: Recursive function to process the image $I_k^{[k,7]}$.

```
Im_processing (k, K_E, I_k^[k,7])
    /* k: index of the current most significant bit-plane;   */
    /* K_E: encryption key;                                  */
    /* I_k^[k,7]: image before adaptation, with pixels encoded on
       8 − k bits;                                           */
    begin
        if k < 7 then
            L_loc^k, L_val^k ← PE_consideration (I_k^[k,7]);
            if size(L_val^k) < size(I_k^[k]) then
                I_{k+1}^[k,7] ← Im_adaptation (I_k^[k,7], L_loc^k, L_val^k);
                I_E^[k] ← BP_encryption (I_{k+1}^[k], K_E);
                I_E-PE^[k] ← Bit_substitution (I_E^[k], L_val^k);
                Im_processing (k + 1, K_E, I_{k+1}^[k+1,7]);
            else
                for i = k to i = 7 do
                    I_E^[i] ← BP_encryption (I_k^[i], K_E);
                    I_E-PE^[i] ← I_E^[i];
        else if k = 7 then
            I_E^[k] ← BP_encryption (I_k^[k], K_E);
            I_E-PE^[k] ← I_E^[k];
```

1) Prediction error consideration: in our proposed method, bits of the secret message are embedded by substitution. Consequently, original bit values of the current bit-plane are lost during the data hiding step. In order to be able to perfectly reconstruct the original image, original bit values have to be predictable. We propose to predict each pixel value $p_k^{[k,7]}(i,j)$ of the image $I_k^{[k,7]}$, using the $7-k$ least significant bit-planes $I_k^{[k+1,7]}$ and the previously scanned pixels. Therefore, the first step of the bit-plane $I_k^{[k]}$ processing in our proposed recursive method consists of analyzing the content of the image $I_k^{[k,7]}$, composed of the $8-k$ least significant bit-planes of the image in clear. During this process, all the pixels which cannot be predicted according to their neighbors are identified and their associated prediction errors (PE) values are evaluated. Note that the first bit value $p_k^{[k]}(0,0)$ cannot be predicted, and therefore, is kept unmodified (*i.e* only encrypted) during the whole process and serves to initialize the prediction.

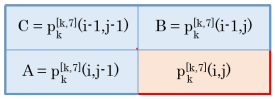


Fig. 3: Context for the prediction of the pixel $p_k^{[k,7]}(i,j)$.

Let us consider a pixel $p_k^{[k,7]}(i,j)$ from $I_k^{[k,7]}$. It is made of $8-k$ bits and defined as:

$$p_k^{[k,7]}(i,j) = \sum_{l=k}^{7} p_k^{[l]}(i,j) \times 2^{7-l}, \tag{1}$$

where $p_k^{[l]}(i,j)$ is the bit of index l.

For the prediction of this pixel, we examine the context illustrated in Fig. 3. For the predictor, we use the Median Edge Detection (MED) predictor, also known as LOCO-I [11]. This predictor consists of detecting the maximal value between the horizontal and vertical edges in the LOCO-I algorithm. It is known to be efficient in JPEG-LS compression standard. Therefore, predictor $pred(i,j)$ of the pixel $p_k^{[k,7]}(i,j)$ is obtained according to:

$$\begin{aligned} pred(i,j) &= \text{MED}(p_k^{[k,7]}(i,j)) \\ &= \begin{cases} \min(A,B) & \text{if } C \geq \max(A,B), \\ \max(A,B) & \text{if } C \leq \min(A,B), \\ A+B-C & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

The PE consideration algorithm is presented in Algorithm 2. After the predictor calculation, the inverse of $p_k^{[k,7]}(i,j)$ is computed: $inv(i,j) = (p_k^{[k,7]}(i,j) + 2^{7-k}) \mod 2^{8-k}$. This value is actually obtained by flipping the most significant bit $p_k^{[k]}(i,j)$ and by not modifying any of the remaining least significant bits $p_k^{[k+1,7]}(i,j)$. Consequently, there is a difference of 2^{7-k} between $p_k^{[k,7]}(i,j)$ and its inverse $inv(i,j)$. The absolute differences between $p_k^{[k,7]}(i,j)$ and $pred(i,j)$, and between $inv(i,j)$ and $pred(i,j)$ are therefore calculated and recorded as Δ and Δ_{inv}. If Δ is smaller than Δ_{inv}, then the original bit value can be predicted. Indeed, this means that the correct pixel value is closer to its predictor than the inverse value. Conversely, if $\Delta > \Delta_{inv}$, there is an error during the prediction of the current pixel, which is highlighted in the PE location map $\mathbf{L_{loc}^k}$. The amplitude of the PE is also computed and stored in the PE value list $\mathbf{L_{val}^k}$.

Note that we must also check if Δ or Δ_{inv} are different to 2^{6-k} and $2^{6-k}+2^{7-k}$. Indeed, in these special cases, we cannot determine the correct value of $p_k^{[k,7]}(i,j)$. For this reason, in these cases, we also highlight an error in $\mathbf{L_{loc}^k}$ and we store a special code in $\mathbf{L_{val}^k}$: $-(2^{6-k}+1)$ if $p_k^{[k]}(i,j) = 0$, and $(2^{6-k}+1)$ if $p_k^{[k]}(i,j) = 1$.

After scanning the entire bit-plane $I_k^{[k]}$, we calculate the size of $\mathbf{L_{val}^k}$. If this size, in bits, is smaller than the current bit-plane size, it is stored by substitution with the first bits

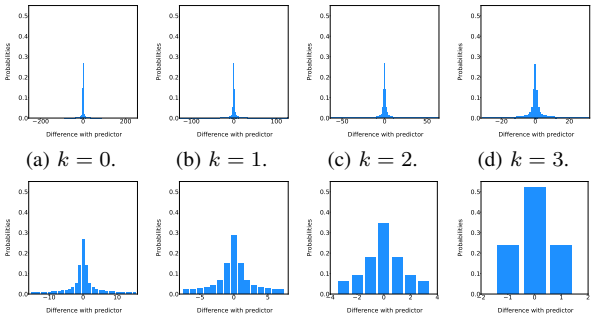
of this bit-plane after encryption. If this is not the case, the detection of the pixels which cannot be predicted and the PE evaluation process are ended and the current and all the remaining least significant bit-planes are only encrypted, as described in Section III-B3.

Algorithm 2: PE consideration for the k^{th} MSB-plane $I_k^{[k]}$.

Data: Clear $m \times n$ image $I_k^{[k,7]}$, with pixels $p_k^{[k,7]}(i,j)$ encoded on $8-k$ bits
Result: $\mathbf{L_{loc}^k}$ PE location map and $\mathbf{L_{val}^k}$ PE value list
$\mathbf{L_{loc}^k} \leftarrow [\,]$;
$\mathbf{L_{val}^k} \leftarrow [\,]$;
for $i \leftarrow 0$ **to** m **do**
  for $j \leftarrow 0$ **to** n **do**
    if $i = 0$ **or** $j = 0$ **and** $(i,j) \neq (0,0)$ **then**
      unidirectional prediction;
    else
      $pred(i,j) \leftarrow \text{MED}(p_k^{[k,7]}(i,j))$;
    $inv(i,j) \leftarrow (p_k^{[k,7]}(i,j) + 2^{7-k}) \mod 2^{8-k}$;
    $\Delta \leftarrow |pred(i,j) - p_k^{[k,7]}(i,j)|$;
    $\Delta_{\text{inv}} \leftarrow |pred(i,j) - inv(i,j)|$;
    if $(\Delta < \Delta_{inv})$ **and** $(\Delta \neq 2^{6-k}$ **or** $\Delta_{inv} \neq 2^{6-k} + 2^{7-k})$ **then**
      /* There is no prediction error */
      $\mathbf{L_{loc}^k}$.append(0);
    else if $(\Delta > \Delta_{inv})$ **and** $(\Delta \neq 2^{6-k} + 2^{7-k}$ **or** $\Delta_{inv} \neq 2^{6-k})$ **then**
      /* There is a prediction error */
      $\mathbf{L_{loc}^k}$.append(1);
      if $p_k^{[k,7]}(i,j) < 2^{7-k}$ **then**
        if $pred(i,j) \leq inv(i,j)$ **then**
          $x \leftarrow pred(i,j) - p_k^{[k,7]}(i,j) - 2^{6-k}$;
        else
          $x \leftarrow pred(i,j) - inv(i,j) - 2^{6-k}$;
      else
        if $pred(i,j) \geq inv(i,j)$ **then**
          $x \leftarrow pred(i,j) - p_k^{[k,7]}(i,j) + 2^{6-k}$;
        else
          $x \leftarrow pred(i,j) - inv(i,j) + 2^{6-k}$;
      $\mathbf{L_{val}^k}$.append(x);
    else
      /* There is a prediction error */
      $\mathbf{L_{loc}^k}$.append(1);
      if $p_k^{[k,7]}(i,j) < 2^{7-k}$ **then**
        $x \leftarrow -(2^{6-k} + 1)$;
      else
        $x \leftarrow 2^{6-k} + 1$;
      $\mathbf{L_{val}^k}$.append(x);
return $\mathbf{L_{loc}^k}$ and $\mathbf{L_{val}^k}$;

In Fig. 4, we represent the differences between each pixel and its associated predictor as a function of the current bit-plane. These experiments are obtained by averaging the values computed on the full BOWS-2 database [2], composed of 10,000 grey-level images, of 512×512 pixels, with different statistical properties. For the bit-plane of index k ($0 \leq k \leq 8$), the value of the difference between a pixel and its predictor is in the range of $-2^{8-k}+1$ to $2^{8-k}-1$. However, whatever the bit-plane, these values are often close to zero. Therefore, the theoretical model for these differences follows a Laplacian distribution. As these values are very small, this means that the amount of incorrectly predicted pixels is not significant, even using a simple predictor as the MED. Moreover, in the case of PE, the to-be-stored value in $\mathbf{L_{val}^k}$ is not large, in particular for the MSB-planes.

In Fig. 5, we have provided an illustration of this analysis on the *Lena* image. On the first row, we have shown the results obtained using the full image (pixels encoded on 8 bits) and then, on the second row, those obtained using



(a) $k = 0$. (b) $k = 1$. (c) $k = 2$. (d) $k = 3$.

(e) $k = 4$. (f) $k = 5$. (g) $k = 6$. (h) $k = 7$.

Fig. 4: Differences between each pixel and its associated predictor as a function of the bit-plane of index k (average values using 10,000 images from the BOWS-2 database [2]).

the seven least significant bit-planes only (pixels encoded on 7 bits). We can first observe the original image and its associated histogram. Then, the difference map between its pixel values and their associated predictors (MED) and its associated histogram are displayed. Whatever the number of bit-planes, we can see that the difference distribution can be always considered as a Laplacian distribution (with different variance values) and is zero-centered. Small difference values are due to high similarities between the pixel values and their associated predictors, and by extension, to the high correlation between the pixel values and their neighboring values.

2) Reversible adaptation of the bit-planes: after the PE consideration, the PE location map $\mathbf{L_{loc}^k}$ and the PE value list $\mathbf{L_{val}^k}$ are obtained. If embedding is possible, the next step consists of adapting the clear image $I_k^{[k,7]}$, according to the PE values in order to be able to detect their location during the decoding phase. Note that this step is fundamental to be able to perfectly reconstruct the original image.

Algorithm 3 describes the necessary steps to adapt the image $I_k^{[k,7]}$. According to $\mathbf{L_{loc}^k}$, all incorrectly predicted pixels $p_k^{[k,7]}(i,j)$ can be identified. Therefore, when an incorrectly predicted pixel is encountered, the first step consists of reading the associated PE value in $\mathbf{L_{val}^k}$. By adding this PE value to the pixel $p_k^{[k,7]}(i,j)$, we obtain the adapted pixel $p_{k+1}^{[k,7]}(i,j)$ as described in Algorithm 3. After this modification, there is a difference of 2^{6-k} or $2^{6-k} + 2^{7-k}$ between the adapted pixel value and its predictor, and of 2^{6-k} between the inverse of $p_{k+1}^{[k,7]}(i,j)$ and its predictor. Note that, during the reconstruction of the original image, these special cases can be identified on the adapted image $I_{k+1}^{[k,7]}$, and the original values can be recovered using $\mathbf{L_{val}^k}$ which has been embedded at the beginning of the current bit-plane. Conversely, $\mathbf{L_{loc}^k}$ is not required for the reconstruction and does not have to be transmitted as an auxiliary file. Contrary to the image pre-processing process in the CPE-HCRDH approach [16], the adaptation step is performed to highlight the location of all incorrectly predicted pixels: its goal is not to correct them. Moreover, original pixel values before adaptation can be recovered without errors and the reconstructed image is not an approximation, but the original image itself.

3) Bit-plane encryption: the encryption key K_{E} is used as a seed for a cryptographically secure pseudo-random number

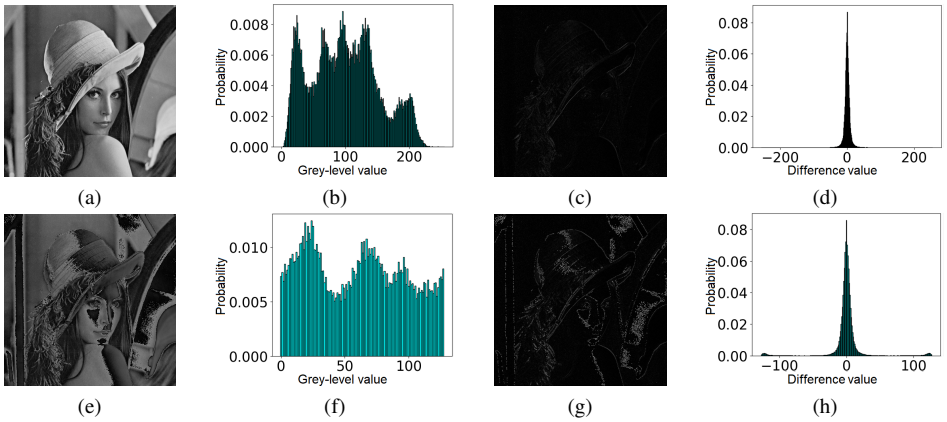


Fig. 5: Illustration of the difference value analysis: a) Original *Lena* image (pixels encoded on 8 bits), b) Histogram associated to the grey-level values of (a), c) Map of the differences between each pixel and its associated predictor from (a), d) Histogram of the difference values (obtained from (c)), e) Image composed of the seven LSB-planes of (a) (pixels encoded on 7 bits), f) Histogram associated to the grey-level values of (e), g) Map of the differences between each pixel and its associated predictor from (e), h) Histogram of the difference values (obtained from (g)).

Algorithm 3: Reversible adaptation of the image $I_k^{[k,7]}$.

Data: Clear $m \times n$ image $I_k^{[k,7]}$, with pixels $p_k^{[k,7]}(i,j)$ encoded on $8-k$ bits, $\mathbf{L_{loc}^k}$ PE location map and $\mathbf{L_{val}^k}$ PE value list, for the k^{th} bit-plane $I_k^{[k]}$

Result: Adapted clear $m \times n$ image $I_{k+1}^{[k,7]}$, with pixels $p_{k+1}^{[k,7]}(i,j)$ encoded on $8-k$ bits

$index \leftarrow 0$;
for $i \leftarrow 0$ **to** m **do**
  for $j \leftarrow 0$ **to** n **do**
    if $\mathbf{L_{loc}^k}[i \times n + j] = 1$ **then**
      /* There is a prediction error */
      if $\left|\mathbf{L_{val}^k}[index]\right| \neq 2^{6-k} + 1$ **then**
        $p_{k+1}^{[k,7]}(i,j) \leftarrow p_k^{[k,7]}(i,j) + \mathbf{L_{val}^k}[index]$;
      $index \leftarrow index + 1$;
    else
      $p_{k+1}^{[k,7]}(i,j) \leftarrow p_k^{[k,7]}(i,j)$;
return $I_{k+1}^{[k,7]}$;

generator to obtain a pseudo-random sequence of $m \times n$ bits $s(i,j)$. Then, for the current bit-plane $I_{k+1}^{[k]}$, *i.e.* the most significant bit-plane of the clear image $I_{k+1}^{[k,7]}$, each bit is XOR-ed with the associated bit in the pseudo-random sequence to give an encrypted bit $p_{\text{E}}^{[k]}(i,j)$ of the encrypted bit-plane $I_{\text{E}}^{[k]}$:

$$p_{\text{E}}^{[k]}(i,j) = s(i,j) \oplus p_{k+1}^{[k]}(i,j). \tag{3}$$

After encryption, if data hiding is possible for the current encrypted bit-plane, the PE value list $\mathbf{L_{val}^k}$ is also encrypted using encryption key K_{E} and embedded by bit substitution at the beginning of the bit-plane from the third bit. Indeed, as explained previously, the first bit is used to initialize the prediction and is kept unmodified. The second bit serves to know if the next bit-plane is marked or not (1 if it is marked, 0 if it is just encrypted). We also have to embed a flag EOL (End Of List) in order to indicate to the data hider the end of $\mathbf{L_{val}^k}$. This flag can be, for example, a sequence of eight consecutive bits equal to 1. At the end of the process, the to-be-marked encrypted bit-plane $I_{\text{E-PE}}^{[k]}$, which contains $\mathbf{L_{val}^k}$, is obtained.

C. On the data embedding side

The data embedding step is completed directly in the encrypted domain, and without knowing encryption key K_{E} used for the encryption step, as shown in the data hiding side (see Fig. 1). Firstly, the data hider knows if bits of the secret message can be embedded in the current bit-plane $I_{\text{E-PE}}^{[k]}$, by checking the second bit of the previous bit-plane (note that we assume that the MSB-plane can always be marked). Then, 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 statistical attack. By following the S-order, the flag EOL is detected and, after that, all the remaining bits $p_{\text{E-PE}}^{[k]}(i,j)$ are used to hide the secret message. The available bits are blindly substituted by bits b^l (with $l < L$, the number of bits which can be marked). Bits $p_{\text{E-DH}}^{[k]}(i,j)$ of the marked encrypted bit-plane $I_{\text{E-DH}}^{[k]}$ are thus obtained:

$$p_{\text{E-DH}}^{[k]}(i,j) = b^l. \tag{4}$$

Fig. 6 illustrates the configuration of a marked encrypted bit-plane $I_{\text{E-DH}}^{[k]}$ at the end of the encoding process. The first bit is not marked, because its value is used to initialize the prediction mechanism. The second bit serves to indicate if the next bit-plane is marked or not. Starting from the third bit, there is the embedded PE value list. The list is followed by the flag EOL which indicates the end of the list, and thus, the beginning of the embedded message. Finally, all remaining bits correspond to bits of the embedded secret message.

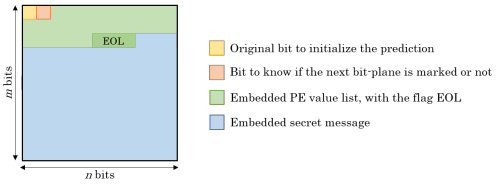


Fig. 6: Composition of each marked encrypted bit-plane at the end of the encoding process.

D. Data extraction and image recovery

On the recipient side, because our proposed method is separable, three situations are considered. If a recipient only has the data hiding key K_{DH}, he can losslessly extract additional data from the marked encrypted image $I_{\text{E-DH}}$. If a recipient only has the encryption key K_{E}, he can perfectly reconstruct

Algorithm 4: Image reconstruction algorithm.

Data: $\mathbf{L^k_{val}}$ list of the prediction error values, $m \times n$ image $I^{[k+1,7]}_{k+1}$, with pixels $p^{[k+1,7]}_{k+1}(i,j)$ encoded on $7-k$ bits, and marked encrypted bit-plane $I^{[k]}_{\text{E-DH}}$
Result: Clear $m \times n$ image $I^{[k,7]}_{k}$, with pixels $p^{[k,7]}_{k}(i,j)$ encoded on $8-k$ bits
/* Initialization of the prediction */
/* $\mathcal{D}(\cdot)$ is the decryption function */
$index \leftarrow 0$;
for $i \leftarrow 0$ **to** m **do**
  for $j \leftarrow 0$ **to** n **do**
    if $(i,j) = (0,0)$ **then**
      $p^{[k,7]}_{k}(0,0) = \mathcal{D}(p^{[k]}_{\text{E-DH}}(0,0)) \times 2^{7-k} + p^{[k+1,7]}_{k+1}(0,0)$;
    else
      $p_0(i,j) \leftarrow p^{[k+1,7]}_{k+1}(i,j) + 0$;
      $p_1(i,j) \leftarrow p^{[k+1,7]}_{k+1}(i,j) + 2^{7-k}$;
      if $i = 0$ **or** $j = 0$ **then**
        unidirectional prediction;
      else
        $pred(i,j) \leftarrow \text{MED}(p^{[k,7]}_{k}(i,j))$;
      $\Delta_0 \leftarrow |pred(i,j) - p_0(i,j)|$;
      $\Delta_1 \leftarrow |pred(i,j) - p_1(i,j)|$;
      if $(\Delta_0 \neq 2^{6-k}$ **or** $\Delta_1 \neq 2^{6-k} + 2^{7-k})$
      and $(\Delta_0 \neq 2^{6-k} + 2^{7-k}$ **or** $\Delta_1 \neq 2^{6-k})$
      and $(\Delta_0 \neq 2^{6-k}$ **or** $\Delta_1 = 2^{6-k})$
      and $(\Delta_0 = 2^{6-k}$ **or** $\Delta_1 \neq 2^{6-k})$ **then**
        if $\Delta_0 < \Delta_1$ **then**
          $p^{[k,7]}_{k}(i,j) \leftarrow p_0(i,j)$;
        else
          $p^{[k,7]}_{k}(i,j) \leftarrow p_1(i,j)$;
      else
        if $|\mathbf{L^k_{val}}[index]| \neq 2^{6-k} + 1$ **then**
          $p_0(i,j) \leftarrow (p_0(i,j) - \mathbf{L^k_{val}}[index]) \mod 2^{8-k}$;
          $p_1(i,j) \leftarrow (p_1(i,j) - \mathbf{L^k_{val}}[index]) \mod 2^{8-k}$;
          $\Delta_0 \leftarrow |pred(i,j) - p_0(i,j)|$;
          $\Delta_1 \leftarrow |pred(i,j) - p_1(i,j)|$;
          if $\Delta_0 > \Delta_1$ **then**
            $p^{[k,7]}_{k}(i,j) \leftarrow p_0(i,j)$;
          else
            $p^{[k,7]}_{k}(i,j) \leftarrow p_1(i,j)$;
        else
          if $\mathbf{L^k_{val}}[index] = -(2^{6-k}+1)$ **then**
            $p^{[k,7]}_{k}(i,j) \leftarrow p_0(i,j)$;
          else
            $p^{[k,7]}_{k}(i,j) \leftarrow p_1(i,j)$;
        $index \leftarrow index + 1$;

the original image using bit prediction, as indicated by a PSNR with the original image which tends to be towards infinity. Finally, if a recipient has both K_{DH} and K_{E} keys, he can losslessly extract the secret message and recover the original image without any alteration.

In the case where the recipient only has the data hiding key K_{DH}, he has to scan the bit-planes of the marked encrypted image $I_{\text{E-DH}}$, starting from the most significant one. First, he knows if the current bit-plane $I^{[k]}_{\text{E-DH}}$ is marked or not, according to the value of the second bit of the previous bit-plane. Note that in Section III-C, we assumed that the MSB-plane is always marked. If this is the case, the first bits consist of the PE value list $\mathbf{L^k_{val}}$ and should not be extracted as bits of the secret message. Therefore, the recipient has to detect the end of $\mathbf{L^k_{val}}$, which is indicated by the flag EOL. Then, he can blindly extract all remaining bits of the bit-plane in order to obtain a part of the embedded data. As soon as a bit-plane indicates that the next bit-plane is not marked, the remaining bit-planes do not contain bits of the secret message. Consequently, the message extraction is completed, and the recipient has to concatenate the extracted data from each marked bit-plane, from the MSB-plane to the last marked LSB-plane. Finally, the complete encrypted secret message is obtained and can be decrypted using the data hiding key K_{DH}. Note that there is no loss, and then no error in the whole process.

If the recipient only has the encryption key, he can reconstruct the original image without alteration thanks to the reversible adaptation step performed on the content owner side. Similarly to the process during the encoding phase, the eight bit-planes of the marked encrypted image $I_{\text{E-DH}}$ are processed recursively, from the LSB-plane to the MSB-plane because the $(k+1)^{\text{th}}$ to 7^{th} bit-planes in the clear domain are used during the prediction of the k^{th} bit-plane. Firstly, the recipient has to observe if the current bit-plane $I^{[k]}_{\text{E-DH}}$ is marked or simply encrypted. In this last case, he generates the associated pseudo-random sequence using the encryption key K_{E} and performs the decryption with a XOR operation. Conversely, if the current bit-plane is marked, all encrypted bits have been replaced by bits of the PE value list $\mathbf{L^k_{val}}$ and bits of the embedded secret message, except the first two bits. Therefore, instead of a full decryption, a prediction is necessary starting from the second bit. In fact, only the first bit is directly decrypted and then $\mathbf{L^k_{val}}$ is extracted from the third bit to the flag EOL. Bit values then have to be predicted, using the first bit to initialize the prediction and $\mathbf{L^k_{val}}$. Moreover, during the recursive process in the encoding phase, as explained in Section III-B2, note that not only the current bit-plane values are modified to make the prediction possible, but also the other least significant bit-plane values. Consequently, the recursive prediction mechanism during the decoding phase takes $\mathbf{L^k_{val}}$, the adapted image $I^{[k+1,7]}_{k+1}$, and the marked encrypted bit-plane $I^{[k]}_{\text{E-DH}}$ as input data, in order to give as output the image $I^{[k,7]}_{k}$ before adaptation. This input and output data, and the different steps of the recursive process are presented in Algorithm 4. The image composed of $I^{[k]}_{\text{E-DH}}$ and $I^{[k+1,7]}_{k+1}$ ($I^{[k]}_{\text{E-DH}} + I^{[k+1,7]}_{k+1}$) is scanned in the S-order, in order to predict each associated pixel value before adaptation $p^{[k,7]}_{k}(i,j)$. Two values are possible for each current adapted pixel: $p_0(i,j) = p^{[k+1,7]}_{k+1}(i,j) + 0$, when the most significant bit $p^{[k]}_{k+1}(i,j)$ is equal to 0, and $p_1(i,j) = p^{[k+1,7]}_{k+1}(i,j) + 2^{7-k}$, when the most significant bit $p^{[k]}_{k+1}(i,j)$ is equal to 1. Actually, the most significant value is the only one that can be wrong at this step. Then, the predictor value $pred(i,j)$ is computed as the MED applied on $p^{[k,7]}_{k}(i,j)$, similarly to the process described in Section III-B1. Note that the neighboring values of $p^{[k,7]}_{k}(i,j)$ before adaptation which serve to the prediction are already reconstructed. Δ_0 and Δ_1 are then calculated as the absolute differences between the two possible pixel values and their predictor $pred(i,j)$. Depending on the Δ_0 and Δ_1 values, different cases have to be considered:

- If no special case is identified, there is no PE. The value of the pixel before adaptation $p^{[k,7]}_{k}(i,j)$, that is equal to the value of the pixel after adaptation $p^{[k,7]}_{k+1}(i,j)$, is obtained by:

$$\begin{aligned} p^{[k,7]}_{k}(i,j) &= p^{[k,7]}_{k+1}(i,j) \\ &= \begin{cases} p_0(i,j) & \text{if } \Delta_0 < \Delta_1, \\ p_1(i,j) & \text{otherwise.} \end{cases} \end{aligned} \quad (5)$$

- The current location corresponds to a PE if a special case is highlighted, *i.e.* $\Delta_0 = \Delta_1 = 2^{6-k}$ or ($\Delta_0 = 2^{6-k}$ and $\Delta_1 = 2^{6-k} + 2^{7-k}$) or ($\Delta_0 = 2^{6-k} + 2^{7-k}$ and $\Delta_1 = 2^{6-k}$). In this case, this means that the current pixel has been adapted during the encoding phase. However, the value of the pixel before adaptation can be recovered by using $\mathbf{L^k_{val}}$ and inverse prediction. Indeed, if such a configuration is encountered, the associated PE value in $\mathbf{L^k_{val}}$ is extracted:
  - If the absolute value of the PE is different to $2^{6-k}+1$, the two possible values for the pixel before adaptation $p_k^{[k,7]}(i,j)$ are obtained by subtracting the PE value to $p_0(i,j)$ and to $p_1(i,j)$. Note that these new values corresponds to $p_k^{[k,7]}(i,j)$ and $inv(i,j)$ during the PE detection and evaluation process described in Section III-B1. In order to discriminate which value is correct, the absolute differences Δ_0 and Δ_1 are evaluated once again, as described in Algorithm 4. This time, based on the fact that the pixel value before adaptation is concerned by a PE, the value of $p_k^{[k,7]}(i,j)$ is determined by the farthest to the predictor $pred(i,j)$:
$$p_k^{[k,7]}(i,j) = \begin{cases} p_0(i,j) & \text{if } \Delta_0 > \Delta_1, \\ p_1(i,j) & \text{otherwise.} \end{cases} \quad (6)$$
  - Conversely, if the absolute value of the PE is equal to $2^{6-k}+1$, then we are in a special case where we cannot determine the correct value of the pixel before adaptation $p_k^{[k,7]}(i,j)$ using prediction. In this case, the sign of the PE value is used to discriminate the correct value of $p_k^{[k,7]}(i,j)$:
$$p_k^{[k,7]}(i,j) = \begin{cases} p_0(i,j) & \text{if the PE is negative,} \\ p_1(i,j) & \text{otherwise.} \end{cases} \quad (7)$$

In a case when the recipient has both the data hiding key and the encryption key, he first extracts bits of the secret message in the same way as a recipient who has only the data hiding key. Then, he can perfectly reconstruct the original image as described previously. In this situation, data extraction and image recovery are still without error.

IV. Experimental results

In this section, we present experimental results obtained by applying our proposed recursive method of very high payload reversible data hiding in encrypted images. Section IV-A illustrates the proposed method, by giving a detailed example. Section IV-B gives performance analysis in terms of payload. These tests are realized on the entire BOWS-2 database of 10,000 grey-level images [2], which are statistically different from each other and present diverse content. In Section IV-C, statistical analysis of the proposed scheme is provided. Finally, in Section IV-D, our method is compared with recent state-of-the-art algorithms.

A. A detailed example for the proposed method

First, we provide an example of the whole process of our proposed method. In Fig. 7, we have applied our method on the *Dolls* original image I with a size of $666\times1,000$ pixels encoded on 256 grey-levels (Fig. 7.a). During the encoding phase, bit-planes are recursively processed from the most significant ($I^{[0]}$) to the least significant ($I^{[7]}$). Therefore, the first step consists of the PE consideration associated to the first bit-plane $I_0^{[0]}$ of the original image. A PE location map, provided in Fig. 7.b, and a PE value list are calculated. Note that the number of PE ($1,628$) is very small and represents only 0.2% of the total number of bits. The size of the PE value list is then analyzed in order to know if the PE values can be embedded in the current bit-plane. As the PE embedding is possible, the image $I_0^{[0,7]}$ is processed to allow PE detection during the reconstruction of the image. In Fig. 7.c, the adapted image $I_1^{[0,7]}$ is then obtained. After this step, the most significant bit-plane is processed separately. It is encrypted and marked by the PE values by the content owner. Finally, part of the secret message is embedded after the PE value list, by bit substitution during the data hiding phase. Fig. 7.d represents the image at the end of the processing of the first bit-plane. It is composed of the marked encrypted most significant bit-plane $I_{\text{E-DH}}^{[0]}$ and the 7 least significant bit-planes of the adapted image, still in the clear domain $I_1^{[1,7]}$. At this step, the total payload of the image $I_{\text{E-DH}}^{[0]} + I_1^{[1,7]}$ is equal to 0.9804 *bpp*, which means that only 0.0196 *bpp* are used to store the PE values in the first bit-plane. After the first bit-plane processing, the second bit-plane $I_1^{[1]}$ of the adapted image is analyzed and processed. Fig. 7.e corresponds to the PE location map computed during the PE consideration associated to $I_1^{[1]}$. Note that the PE number is larger than for the first bit-plane ($29,324$, namely 4.4% of the total number of bits). This is explained by the fact that the less a bit plane is significant, the less the bits are correlated. However, the PE value list is sufficiently small to be embedded in the current bit-plane. $I_1^{[1,7]}$ is therefore adapted to make possible the PE detection and $I_2^{[1,7]}$ is obtained (Fig. 7.f). Fig. 7.g corresponds to the image after encryption and data embedding in the current bit-plane $I_2^{[1]}$. In fact, it is composed of two marked encrypted bit-planes $I_{\text{E-DH}}^{[0,1]}$ and the remaining 6 least significant bit-planes $I_2^{[2,7]}$. The payload of the image $I_{\text{E-DH}}^{[0,1]} + I_2^{[2,7]}$ is equal to 1.6722 *bpp*, which indicates a gain of 0.6918 *bpp* by using the second bit-plane. The same steps are repeated on the third bit-plane $I_2^{[2]}$. Fig. 7.h is the PE location map. This time, $77,094$ bits (11.58%) correspond to a PE, but PE value embedding is still possible. The image $I_2^{[2,7]}$ is adapted, according to the PE consideration associated to the third bit-plane. The adapted image $I_3^{[2,7]}$ is displayed on Fig. 7.i. Moreover, as illustrated in Fig. 7.j, the payload of the image $I_{\text{E-DH}}^{[0,2]} + I_3^{[3,7]}$ is equal to 1.9777 *bpp*, which means that the use of the third bit-plane for data embedding allows a rise of 0.3055 *bpp*. In Fig. 7.k, the PE location map associated to the fourth bit-plane $I_3^{[3]}$ is illustrated. Due to an important number of PE ($144,014$, namely 21.63% of the bits), the size of the PE value list is too large to be embedded in the current bit-plane. The recursive process is thus ended and the current and all the remaining bit-planes in the clear domain are only encrypted. The resulting image, at the end of the encoding process, is provided in Fig. 7.l. The final payload

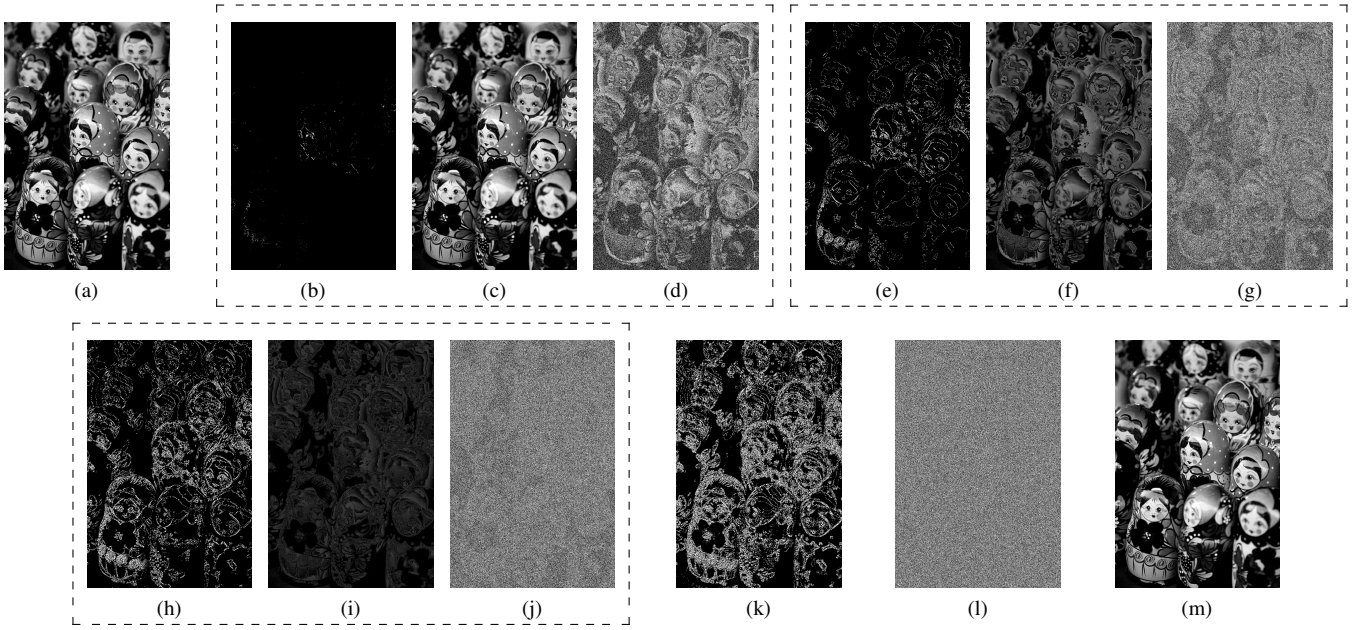


Fig. 7: Illustration of our proposed recursive method: a) Original *Dolls* image I of $666 \times 1,000$ pixels, b) PE location map associated to the 1st bit-plane $I_0^{[0]}$ (MSB-plane), number of errors = 1,628 (0.2%), c) Adapted image $I_1^{[0,7]}$ after PE consideration, d) Image $I_{\text{E-DH}}^{[0]} + I_1^{[1,7]}$ composed of the marked encrypted 1st bit-plane and the 7 least significant bit-planes of (c), payload = 0.9804 *bpp*, e) PE location map associated with the 2nd bit-plane $I_1^{[1]}$, number of errors = 29,324 (4.4%), f) Adapted image $I_2^{[1,7]}$ after PE consideration, g) Image $I_{\text{E-DH}}^{[0,1]} + I_2^{[2,7]}$ composed of the marked encrypted 1st and 2nd bit-planes and the 6 least significant bit-planes of (f), payload = 1.6722 *bpp* (+0.6918 *bpp*), h) PE location map associated with the 3rd bit-plane $I_2^{[2]}$, number of errors = 77,094 (11.58%), i) Adapted image $I_3^{[2,7]}$ after PE consideration, j) Image $I_{\text{E-DH}}^{[0,2]} + I_3^{[3,7]}$ composed of the marked encrypted 1st, 2nd and 3rd bit-planes and the 5 least significant bit-planes of (i), payload = 1.9777 *bpp* (+0.3055 *bpp*), k) PE location map associated with the 4th bit-plane $I_3^{[3]}$, number of errors = 144,014 (21.63%), l) Image $I_{\text{E-DH}}$ after encryption using all bit-planes and data embedding in the 1st, 2nd and 3rd bit-planes, final payload = 1.9777 *bpp*, m) Reconstructed original image I, PSNR $\rightarrow +\infty$, SSIM = 1.

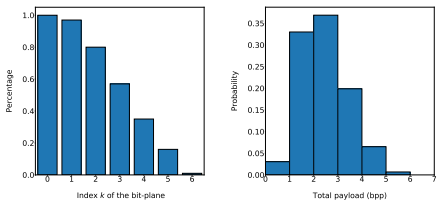
is then 1.9777 *bpp*, after data embedding in the three most-significant bit-planes. During the decoding phase, as presented in Fig. 7.m, the original image I can be perfectly reconstructed, by processing each bit-plane from the least significant to the most significant. Indeed, the least significant bit-planes are necessary to predict the most significant ones. Therefore, the PSNR value between the original image and the recovered image tends to move towards infinity and the SSIM value is equal to 1.

B. Performance analysis on an entire image database

In order to test the efficiency of the proposed recursive method for various image contents, we have applied our scheme to the BOWS-2 database [2].

In Fig. 8.a, we present the repartition of the database images, according to the possibility to embed data in each bit-plane. First of all, we can see that, for this database, in all cases, data embedding can be achieved in the first bit-plane (MSB-plane, $k = 0$). This is due to the very strong correlation between neighboring most significant bits in each image. Until the third bit-plane, this correlation remains significant and so, data embedding is possible in the second bit-plane ($k = 1$) for 97% of the images and in the third bit-plane ($k = 2$) for 80% of the images. After the third bit-plane, the amount of images where it is possible to mark the remaining bit-planes decreases. Indeed, data embedding can be achieved until the 4th bit-plane ($k = 3$) for 57% of the images, until the 5th bit-plane ($k = 4$) for 35% of the images, and until the 6th bit-plane ($k = 5$) for 16% of the images. Moreover, only a few images (0.1%) can be marked from the MSB-plane to the last bit-plane which can be predicted ($k = 6$, corresponding to the second LSB-plane). These images are not textured and seem quite homogeneous, which explains the predictability of each bit-plane. Note that the least significant bit-plane (LSB-plane, $k = 7$) is never marked, this is because its values cannot be predicted with the help of other bit-planes. Fig. 8.b is the distribution of the database images according to the payload. The obtained results are consistent with those of the Fig. 8.a. Indeed, only 3% of images have a payload of less than 1 *bpp*. These images are the most textured ones from the database. As there are a lot of edges, neighboring pixels are not well correlated. For this reason, only the first bit-plane is generally marked in these images and the maximal payload is therefore 1 *bpp*. As shown previously, most images are marked until the third plane. In this case, the maximal possible payload is of 3 *bpp*. Consequently, in Fig. 8.b, we can see that the payload value is between 1 *bpp* and 3 *bpp* for 70% of images. This also means that, for 27% of images in the database, the payload is very high and larger than 3 *bpp*. For 20% of images, it is between 3 *bpp* and 4 *bpp*. For the remaining 7%, it is higher than 4 *bpp*, when the data embedding can be achieved in the least significant bit-planes and, sometimes, until the 7th bit-plane. The maximal possible

payload is therefore 7 *bpp*. Furthermore, we can see that there is less than 1% of images which have a payload between 6 *bpp* and 7 *bpp*. This is explained by the fact that, even if data embedding is possible, due to the significant size of the PE value list, the amount of embedded bits of the secret message is relatively small in the least significant bit-planes.



(a) (b)

Fig. 8: a) Repartition of the images from the BOWS-2 database [2] ordered according to the possibility of embedding data in the most significant bit-plane of index k, b) Distribution of the images from the BOWS-2 database [2] according to the payload.

Moreover, Table I and Table II present respectively the percentage of PE and payload value measured in each image of the database, according to the considered index k of the bit-plane. Note that we only consider the images for which the data embedding is possible in the previous bit-plane of index $k-1$ for the PE calculation, and in the current bit-plane of index k for the payload evaluation. The results are given in terms of quartiles (Q1, median and Q3) and average. First, we can see that the number of errors is very small for the MSB-plane and close to 0%. As a result, the payload in this bit-plane is very high and close to the maximal value of 1 *bpp*. The second bit-plane is also highly predictable. Indeed, the amount of PE is smaller than 10% in all cases. The payload value is still high: it is larger than 0.6105 *bpp* for 75% of images, and even larger than 0.8881 *bpp* for 25% of images. If we consider the third bit-plane, the PE number is a little more important and of the order of 10%. Consequently, payload decreases, but still remains high: for more than 50% of images, it is higher than 0.5 *bpp*. In the fourth bit-plane, the PE number is larger than 15% on average, and so, the associated median payload value for this bit-plane is of 0.3574 *bpp*. However, for 25% of images, it is still larger than 0.5 *bpp*. As a reminder, less than 50% of the total number of images in the database can be marked after the fourth bit-plane. In the fifth bit-plane, the amount of PE is larger than 20% in most cases, and the associated payload value is generally smaller than 0.4 *bpp*. It is worth noting that in the sixth bit-plane, where there are more than 30% of PE, and a payload smaller than 0.2 *bpp* for most images. In the seventh bit-plane, the number of PE is very important (almost 60%). For this reason, this bit-plane cannot be marked in almost all cases. Nevertheless, 0.1% of the total number of images of the database is very homogeneous. Hence, the PE value list remains short due to a small amount of PE and because only two bits are required to code each of them (without compression). Due to that, data embedding is possible and, in these special cases, the payload can be quite high, with a median value equal to 0.3875 *bpp* and an average of 0.4338 *bpp*. Finally, in the last column of Table II, we can see the values of the final payload, after embedding data in all bit-planes when it is possible. The results indicate a very high payload for all images in the database. In fact, 75% of images have a payload value larger than 1.7224 *bpp*. Moreover, the median value is 2.3209 *bpp* and the average value is 2.4586 *bpp*. In addition, for 25% of the images, we achieve a very high value, larger than 3.0759 *bpp*.

% of PE	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
Q1 (25%)	0.0092	1.6373	5.4136	10.6075	17.2287	28.7792	57.2205
Median (50%)	0.0469	3.2806	9.0668	15.6956	22.7856	33.9851	60.8387
Q3 (75%)	0.1545	5.8624	13.8999	20.9782	27.4303	38.2093	63.4789
Average	0.1507	4.3247	10.3718	15.9634	22.1918	33.2427	59.4856

TABLE I: Percentage of PE on images from the BOWS-2 database [2], according to the considered k^{th} bit-plane, with $0 \leq k \leq 6$.

Payload (*bpp*)	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=0$ to 6
Q1 (25%)	0.9876	0.6105	0.3315	0.1834	0.1181	0.0722	0.3711	1.7224
Median (50%)	0.9962	0.7770	0.5289	0.3574	0.2564	0.1518	0.3875	2.3209
Q3 (75%)	0.9993	0.8881	0.7089	0.5495	0.4233	0.2734	0.4536	3.0759
Average	0.9879	0.7221	0.5146	0.3765	0.2867	0.1913	0.4338	2.4586

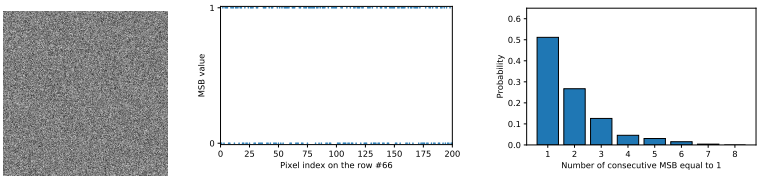
TABLE II: Payload measurements (in *bpp*) on images from the BOWS-2 database [2], according to the considered k^{th} bit-plane, with $0 \leq k \leq 6$.

C. Statistical analysis

In this section, we analyze the security of the proposed method from a statistical point of view.

In Fig. 9, we evaluate the visual security level of a marked encrypted image obtained with our RDHEI method. Fig. 9.a is the marked encrypted image associated to the original *Baboon* image. Although both PE value list and secret message are embedded in the encrypted domain, there is no visual artifact. In order to perform this analysis, we also focus on one row of the marked encrypted image. As an example, in Fig. 9.b, MSB values of the first 200 pixels of the row $\#66$ are plotted for the marked encrypted image illustrated in Fig. 9.a. We can see that there are no long sequences of bits with the same value. These results are consolidated in Fig. 9.c which represents the distribution of sequences of consecutive MSB equal to 1 depending on their length in the entire image. Statistical properties for a marked encrypted image should follow the Geometric law with parameter $p = 0.5$, which is actually the case for the results obtained with our proposed method, as displayed in Fig. 9.c.

Table III and Fig. 10 present a comparison of statistical analysis between the original *Baboon* image and the marked encrypted version obtained with our method illustrated in Fig. 9.a. The used statistical metrics are horizontal and vertical correlations, histogram representation, Shannon entropy and χ^2 test (square root). Number of changing pixel rate (NPCR), unified averaged changed intensity (UACI) and peak-signal-to-noise ratio (PSNR) between the original and the marked encrypted images are also analyzed. According to Fig. 10.a and the first row of the Table III, the correlation between neighboring pixels is very high in the original *Baboon* image. In both horizontal and vertical directions, values are close to 1 (0.8611 and 0.7666). By contrast, as shown in Fig. 10.c and the second row of the Table III, this correlation is very low in the marked encrypted image. Adjacent pixels are very different: both horizontal and vertical correlation values are close to 0 (0.0005 and 0.0007). Fig. 10.b and Fig. 10.d correspond to

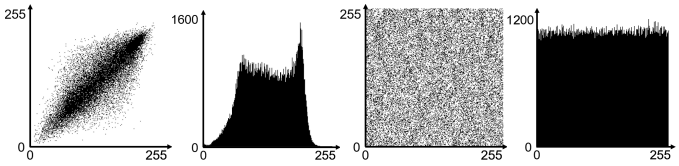


(a) (b) (c)

Fig. 9: Visual security level evaluation of our proposed method: a) Marked encrypted image of *Baboon* obtained with the proposed method, b) MSB values of the first 200 pixels of row #66 of the image in (a), c) Distribution of sequences of consecutive MSB equal to 1 depending on their length in the entire image in (a).

Image	Horizontal correlation	Vertical correlation	Entropy (*bpp*)	χ^2 test		NPCR (%)	UACI (%)	PSNR (dB)
				score	p-value			
Original image	0.8611	0.7666	7.4744	142.81×10^3	0	/	/	/
Marked encrypted image (Fig. 9.a)	0.0005	0.0007	7.9994	228.65	0.8810	99.6037	28.6723	9.2245

TABLE III: Quality evaluation of the marked encrypted image using the proposed method.



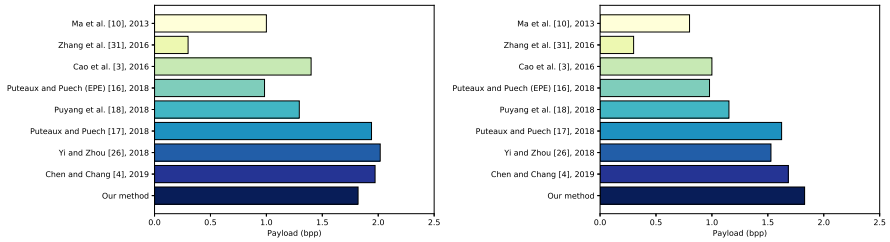
(a) (b) (c) (d)

Fig. 10: Statistical representations (correlation and histogram) for the original *Baboon* image and the associated marked encrypted image obtained with our method illustrated Fig. 9.a: a) Horizontal correlation in the original image, b) Histogram of the original image, c) Horizontal correlation in the marked encrypted image (Fig. 9.a), d) Histogram of the marked encrypted image (Fig. 9.a).

the histograms of the pixels of the original image and its marked encrypted version respectively. Contrary to the pixel distribution of the original image, pixel distribution associated to the marked encrypted image tends to be uniform. This means that it is not possible to obtain statistical information about the original image content from the histogram of the marked encrypted image. These two histogram representations are consistent with the two Shannon entropy values. For the marked encrypted image, it is close to the maximal value of 8 *bpp* (7.9994 *bpp*), whereas it is equal to 7.4744 *bpp* for the original image. The uniformity of the pixel distribution in the marked encrypted image is also ensured by applying the χ^2 test. For this test, we have $L = 256$ possible grey-levels and then, $L - 1 = 255$ degrees of freedom, and we consider an error risk $\alpha = 0.05$. By referring to the χ^2 table, the value $\chi^2(255, 0.05)$ is equal to 293.25. Moreover, the obtained χ^2 score is very high for the original image (142.81×10^3) and much smaller for the marked encrypted image (228.65). The associated p-values are 0 and 0.8810 respectively. As expected, the original image distribution is not uniform, according to the obtained results. Conversely, for the marked encrypted image, we have a score which is smaller than $\chi^2(255, 0.05)$ and a p-value larger than 0.1. This means that there is no presumption against the null hypothesis: the pixel distribution in the marked encrypted image seems to be uniform. Therefore, the marked encrypted image is not vulnerable to statistical attacks based on histogram analysis. NPCR value is very high and close to the maximal value of 100% (99.6037%), UACI is equal to 28.6723% and PSNR is low (9.2245 dB). These different measurements attest to the strong differences between the original and marked encrypted image contents. They also highlight that embedding the PE values and the secret message has no impact on the security of the encryption scheme. In conclusion, we note that our proposed method seems to be statistically secure.

D. Comparisons with related methods and discussion

In Fig. 11, we compare our proposed method with recent state-of-the-art algorithms: the methods of Ma *et al.* [10], Zhang *et al.* [31], Cao *et al.* [3], Puyang *et al.* [18], Yi *et al.* [26], Chen and Chang [4] and two other approaches by Puteaux and Puech, which are EPE-HCRDH [16] and its extension using all the possible bit-planes [17]. The comparison was made with the payload (expressed in *bpp*). Note that the payload calculated for our method corresponds to the true payload value, which means the amount of embedded bits of the secret message only, *i.e.* excluding the embedding PE value list and the flag EOL for each marked bit-plane. For these comparisons, we used the two well known grey-level images of *Lena* and *Man*. We do not compare the quality of the reconstructed image because, whatever the used method, the original image can be losslessly recovered during the decoding phase (PSNR$\rightarrow +\infty$ and SSIM $= 1$). This can be completed with the encryption key only [17], [18], [26], or using both encryption and data hiding keys [3], [4], [10], [31]. We can see that our proposed method obtains very good results. For the two images, the payload value obtained by Zhang *et al.* [31] is smaller than 0.5 *bpp*. Using the methods of Ma *et al.* [10], Cao *et al.* [3], Puyang *et al.* [18] or the EPE approach of Puteaux and Puech [16], the payload is higher, but it is close to 1 *bpp* and does not exceed 1.5 *bpp*. Puteaux and Puech [17], Yi *et al.* [26] and Chen and Chang [4] methods are the three most recent methods achieving a very high payload. Consequently, all of them allow us to embed a large amount of information for the two images. For *Lena*, the obtained payload is close to 2 *bpp* and, for *Man*, it is higher than 1.5 *bpp*, but does not exceed 1.75 *bpp*. Using our proposed method in this paper, it is possible to embed secret message data in the three most-significant bit-planes of the two test images. Consequently, payload values are very high: 1.8100 *bpp* for *Lena* and 1.8289 *bpp* for *Man*. Note that these



(a) Test image: *Lena*. (b) Test image: *Man*.

Fig. 11: Performance comparisons between our method and recent state-of-the-art methods [3], [4], [10], [16]–[18], [26], [31].

similar payload values are consistent with the compression rates computed between these two original images and their compressed versions using JPEG-LS (1.7415 for *Lena* and 1.7067 for *Man*). As a result, we can see that our results are comparable with those obtained by Puteaux and Puech [17], Yi *et al.* [26] and Chen and Chang [4].

In order to illustrate the efficiency of our proposed scheme, in Fig. 12, we analyze the results on the full BOWS-2 database [2] obtained by our approach and methods by Puyang *et al.* [18], Puteaux and Puech [16], [17], Yi *et al.* [26] and Chen and Chang [4]. As already noticed previously in Fig. 11, payload values obtained by Puteaux and Puech [16] and Puyang *et al.* [18] approaches allow for a high payload, but average associated payload values do not exceed 1.5 *bpp*. Moreover, even if the methods in [17], [26] and [4] achieve better results for relatively smooth images (as shown in Fig. 11), average payload values are less important than using our proposed method. Indeed, the average payload on the database with our new scheme is equal to 2.4586 *bpp*. Note that there are some reasons for this payload increase compared to other methods based on MSB prediction in [18] and [17]. First, in these two last methods, whatever the processed bit-plane, nearly 24 bits are lost in case of a prediction error (PE) and cannot be used for data embedding. This is because bit-planes are processed by sequences of 8 bits and two flags are used to highlight a sequence with PE. With our new proposed approach, each bit-plane is processed bit by bit. If a PE occurs, the value of the PE is stored in the PE value list. Each PE value is then encoded using $8-k$ bits (considering the k^{th} bit-plane as the current bit-plane). This means that even if there are more PE in the least significant bit-planes than in the most significant one, a smaller number of bits is used to encode each PE value. As less bits are used to encode a PE in this new method than in the approaches described in [18] and [17], the obtained payload is more significant. It is even higher using a more efficient predictor (Median Edge Detector). Using such a predictor, the number of PE decreases, which improves the payload value.

To conclude, through various experiments, we note that our proposed recursive data hiding method in the encrypted domain is perfectly reversible (PSNR$\rightarrow +\infty$, SSIM $= 1$) and allows for a very high payload, in addition it is error-free during the extraction of the secret message. The obtained results with our proposed method outperform those in recent state-of-the-art methods by obtaining a median payload value equal to 2.3209 *bpp* and an average of 2.4586 *bpp*, while

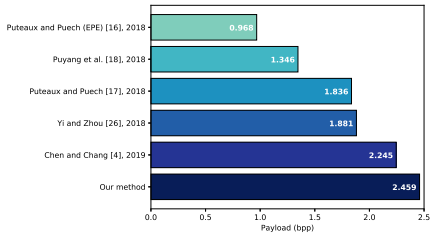


Fig. 12: Performance comparisons between our proposed method and recent high payload, state-of-the-art methods [4], [16]–[18], [26] on images from the BOWS-2 database [2].

other methods based on MSB prediction in [18] and [17] achieve 1.3460 *bpp* and 1.8360 *bpp* for the average value respectively. In fact, only the two first MSB-planes are used to embed information in [18] and it is also often the case in [17]. Conversely, with our proposed method, in most cases, at least the three most significant bit-planes of an image are predictable and can therefore be used for data hiding. Note also that only the encryption key is needed to reconstruct the original image from the marked encrypted image, which means that the proposed method is fully separable. Moreover, the statistical analysis shows that our scheme offers a good level of security, because there is no information about the original image content analyzing the marked encrypted image. Finally, the proposed scheme offers a very good trade-off between reconstructed image quality and payload, while being statistically secure.

V. Conclusion

In this paper, a new method of data hiding in encrypted images, which is fully reversible and allows for a very high payload, is presented. With an average payload of 2.4586 *bpp* and a median value of 2.3209 *bpp*, it outperforms the latest modern state-of-the-art methods, where the maximal payload is often in the order of 1 *bpp*. In this new approach, the bit-planes of the original image are processed recursively, from the most significant one to the least significant. During the encoding phase, for each bit-plane, the clear content is analyzed and the PE are considered by computing a PE location map and a PE value list. A test is then realized in order to know if the PE value list can be embedded in the current bit-plane. If it is possible, the image is adapted to highlight the location of all incorrectly predicted pixels. Note that this adaptation step is perfectly reversible: it does not prevent the lossless recovery of the original image. The current bit-plane is then processed separately. It is encrypted and marked by bits of the

PE value list and of the secret message. If the embedding is not possible in the current bit-plane then the recursive process is ended and in this case, the current and the remaining bit-planes are only encrypted. During the decoding phase, a smart reconstruction is performed. Each bit-plane is recursively reconstructed from the LSB-plane to the MSB-plane. In fact, all the least significant bit-planes are necessary to reconstruct a most significant one. Moreover, all bit-planes of the original image can be perfectly recovered (PSNR $\rightarrow +\infty$, SSIM = 1) using prediction. Indeed, all adapted pixels – corresponding to incorrectly predicted pixels – can be located by identifying special cases, and corrected using the inserted PE value list. In addition to the excellent trade-off between the reconstructed image quality and the possible payload, the marked encrypted image is statistically secure and there is no loss of information from the original image content. Therefore, this RDHEI method can be used to provide image confidentiality, whilst allowing for authenticity or integrity checks with the help of the embedded message.

In future work, we are interested in finding an effective way to allow the decryption of the marked encrypted image obtained with the proposed method while preserving the embedded secret message. In this context, we are also involved in the design of a format-compliant encryption method which would be homomorphic to data embedding. Moreover, we investigate how embedding a large amount of information in JPEG crypto-compressed images in order to achieve high payload reversible data hiding in JPEG crypto-compressed images.

REFERENCES

[1] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

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

[3] 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*, vol. 46, no. 5, pp. 1132–1143, 2016.

[4] K. Chen and C.-C. Chang, "High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 334–344, 2019.

[5] 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*, vol. 2007, p. 17, 2007.

[6] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Security and Watermarking of Multimedia contents*, vol. 4314. International Society for Optics and Photonics, 2001, pp. 197–208.

[7] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1061–1070, 2011.

[8] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[9] Y. Hu, H.-K. Lee, K. Chen, and J. Li, "Difference expansion based reversible data hiding using two embedding directions," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1500–1512, 2008.

[10] 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*, vol. 8, no. 3, pp. 553–562, 2013.

[11] N. D. Memon, X. Wu, V. Sippy, and G. Miller, "Interband coding extension of the new lossless JPEG standard," in *Visual Communications and Image Processing*, vol. 3024. International Society for Optics and Photonics, 1997, pp. 47–59.

[12] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[13] 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*, vol. 22, no. 12, pp. 5010–5021, 2013.

[14] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Processing*, vol. 92, no. 1, pp. 54–62, 2012.

[15] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, vol. X. International Society for Optics and Photonics, 2008, pp. 68 191E–68 191E.

[16] 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*, vol. 13, no. 7, pp. 1670–1681, 2018.

[17] ——, "EPE-based huge-capacity reversible data hiding in encrypted images," in *IEEE Workshop on Information Forensics and Security*. IEEE, 2018, pp. 1–7.

[18] Y. Puyang, Z. Yin, and Z. Qian, "Reversible data hiding in encrypted images with two-MSB prediction," in *IEEE Workshop on Information Forensics and Security*. IEEE, 2018, pp. 1–7.

[19] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1486–1491, 2014.

[20] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted JPEG bitstreams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 351–362, 2018.

[21] 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*, vol. 19, no. 7, pp. 989–999, 2009.

[22] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.

[23] J. Tian, "Reversible watermarking by difference expansion," in *Proceedings of Workshop on Multimedia and Security*, vol. 19, 2002.

[24] ——, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[25] W. Trappe and L. C. Washington, *Introduction to cryptography with coding theory*. Pearson Education India, 2006.

[26] S. Yi and Y. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 51–64, 2019.

[27] W. Zhang, H. Wang, D. Hou, and N. Yu, "Reversible data hiding in encrypted images by reversible image transformation," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1469–1479, 2016.

[28] X. Zhang, "Reversible data hiding with optimal value transfer," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 316–325, 2012.

[29] ——, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[30] ——, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[31] 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*, vol. 26, no. 9, pp. 1622–1631, 2016.

Pauline Puteaux received her M.S. degree in Computer Science and Applied Mathematics, with specialization in Cybersecurity, from the University of Grenoble, France, in 2017. She is currently pursuing her Ph.D. degree with the Laboratory of Informatics, Robotics and Microelectronics of Montpellier, France. Her work has focused on multimedia security, and in particular, image analysis and processing in the encrypted domain. Since 2016, she has published 4 journal papers and 7 conference papers. She is a reviewer for Signal Processing (Elsevier), J. of Visual Communication and Image Representation (Elsevier), IEEE Trans. on Circuits and Systems for Video Technology and IEEE Trans. on Dependable and Secure Computing.

William Puech received the diploma of Electrical Engineering from the Univ. Montpellier, France (1991) and a Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France (1997) with research activities in image processing and computer vision. He served as a Visiting Research Associate to the University of Thessaloniki, Greece. From 1997 to 2008, he has been an Associate Professor at the Univ. Montpellier, France. Since 2009, he is full Professor in image processing at the Univ. Montpellier, France. His current interests are in the areas of image forensics and security for safe transfer, storage and visualization by combining data hiding, compression, cryptography and machine learning. He is head of the ICAR team (Image and Interaction) in the LIRMM, has published more than 40 journal papers and 120 conference papers and is associate editor for 5 journals (JASP, SPIC, SP, JVCIR and IEEE TDSC) in the areas of image forensics and security. Since 2017 he is the general chair of the IEEE Signal Processing French Chapter and since 2018 he is a member of the IEEE Information Forensics and Security TC.