Homomorphic Encryption-Based LSB Substitution for High Capacity Data Hiding in the Encrypted Domain

Pauline Puteaux, Manon Vialle, William Puech

Date of publication xxxx 00, 0000, date of current version June 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001385

Homomorphic Encryption-Based LSB Substitution for High Capacity Data Hiding in the Encrypted Domain

PAULINE PUTEAUX, (Student Member, IEEE), MANON VIALLE, and WILLIAM PUECH, (Senior Member, IEEE)

LIRMM, Université de Montpellier, CNRS, Montpellier, France (e-mail: pauline.puteaux@lirmm.fr; manon.vialle@grenoble-inp.org; william.puech@lirmm.fr)

Corresponding author: William Puech (e-mail: william.puech@lirmm.fr).

We would like to thank the financial support of the ANR-16-DEFA-0001 OEIL (statistiques rObustEs pour l'apprentIssage Léger) research project of the French ANR/DGA challenge DEFALS (DEtection de FALSifications dans des images).

ABSTRACT During the last few decades, multimedia security over the cloud has become a major issue. Public-key homomorphism is an efficient approach for data hiding in encrypted images (DHEI). An original image is encrypted using a public key and sent across a network and then processed to embed a secret message directly in the encrypted domain. During the decoding step, a private key is used to obtain a marked reconstructed image, where the secret message can be extracted. In this paper, we propose an efficient method of DHEI based on the Paillier cryptosystem. Using its homomorphic properties, pixel blocks and bits of the message are multiplied in the encrypted domain resulting in an addition in the clear domain. By applying a pre-processing step on the original image before encryption, this addition becomes a least significant bits (LSB) substitution. Experimental results show that using our proposed scheme, we obtain a high payload value (1 bpp) without expanding a lot the original image size contrary to current state-of-the-art methods. Indeed, whatever the key size, the expansion rate is always equal to 2. In addition, the original image and the marked reconstructed image are very similar.

INDEX TERMS Multimedia security, image homomorphic encryption, data hiding, Paillier cryptosystem, signal processing in the encrypted domain.

I. INTRODUCTION

With the constant evolution of the Internet and cloud services, multimedia data exchanged over the networks need to be protected against illegal access and fraudulent usage. The aim of encryption methods is to guarantee data privacy by converting the content of original images into unintelligible ciphertext data [1]. 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, with the increasing demand for information security, the development of signal processing in the encrypted domain techniques has been an issue of great attention in the field of privacy protection and multimedia security [2]–[5].

For the ten past years, data hiding in encrypted images (DHEI) methods have been proposed [6]–[11] to achieve image annotation or authentication in the encrypted domain. They allow data embedding to be performed directly in the encrypted domain, which is secure because no information about the original image content is revealed. The challenge lies in finding the best trade-off between the payload and the reconstructed image quality. State-of-the-art methods can be divided into two categories depending on the order of operations during the encoding phase. These two categories are Reserving Room Before Encryption (RRBE) and Vacating Room After Encryption (VRAE). In RRBE methods, for a content owner, the first step consists to pre-process the original image in the clear domain to release enough space to embed data and then, to encrypt the pre-processed image. A data hider can then embed bits of a secret message into specific positions [9]–[11]. In VRAE methods, the original image content is blindly encrypted by the content owner, and then the data hider can modify the encrypted data in order to hide bits of a secret message [6]–[8]. Moreover, during the decoding phase, image reconstruction and data extraction

are processed at the same time (joint methods) or separately (separable methods).

Recently, many DHEI methods based on public-key homomorphism have been proposed [12]–[18]. Indeed, public-key cryptosystems with probabilistic and homomorphic properties, such as the Paillier cryptosystem [19], are widely used in secure computation. In such methods, from an original image, an encrypted image is generated by using a public key and sent across a network. Then, a secret message can be directly embedded in the encrypted domain using the same public key. On the recipient side, an authorized user can use the private key (associated to the public key) to decrypt the marked encrypted image to reconstruct a marked image in clear and extract the secret message. Moreover, some methods are perfectly reversible, when the original image can be reconstructed without any distortion after the message extraction [14]. Chen *et al.* first designed a scheme where the Paillier cryptosystem is used for encryption [12]. Shiu *et al.* introduced an approach based on difference expansion [13]. Zhang *et al.* described a reversible approach and a lossless data hiding method for images that have been encrypted using a public-key cryptosystem [14]. Xiang and Luo suggested to divide an original image into two parts for self-embedding before encryption [15]. Zhou *et al.* suggested a public-key modulation mechanism, which allows them to embed the data via simple XOR operations. On the decoder side, they proposed to use a powerful two-class SVM classifier to discriminate encrypted and non-encrypted image patches, enabling them to jointly decode the embedded message and the original image signal perfectly [16]. Zheng *et al.* also proposed a lossless data hiding method, this is based on homomorphic cryptosystem which achieves a high embedding rate through efficient mapping and skillful use of expanded pixel values [17]. In addition, Zhou *et al.* designed a separable reversible data hiding scheme in homomorphic encrypted domain based on NTRU [18]. None of the existing methods based on public-key homomorphism succeeds in achieving high embedding capacity without significantly expanding the size of the original data. Indeed, the expansion rate of these previous methods depends on the key size. For instance, for keys of 512 bits, it is between 128 and 256.

In this paper, we propose a new efficient method of data hiding in encrypted images using the Paillier cryptosystem. During the encoding phase, an original image is pre-processed and encrypted by blocks of pixels. Homomorphic properties are then exploited to multiply the encrypted image by the encrypted message, which corresponds to substitute all the LSB values in the clear domain by bits of a secret message. The recipient of the marked encrypted image is then capable to recover a marked image in clear which is very similar to the original image and to losslessly extract the secret message. According to our obtained results, we achieve a very interesting trade-off between the data expansion, the payload and the reconstructed image quality. Indeed, the data expansion of our method is equal to 2. Moreover, the Peak Signal to Noise Ratio (PSNR) values between the marked reconstructed images in clear and their corresponding original images are larger than $50\ dB$ and the Structural SIMilarity (SSIM) values are very close to 1 for all test-images. In addition, the obtained payload is very high because each pixel of this image holds a secret message bit in its LSB (payload $1\ bpp$).

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

- We propose an efficient method of DHEI based on the Paillier cryptosystem.
- Because of the homomorphic properties of the Paillier cryptosystem, the multiplication of the encrypted message by the encrypted image becomes a least significant bits (LSB) substitution in the clear domain.
- A pre-processing step is applied on the original image before encryption.
- Compared to recent state-of-the-art methods, our proposed scheme does not significantly increase the original image size (expansion rate of 2).
- We achieve a very interesting trade-off between the data expansion, the payload and the reconstructed image quality.

The rest of the paper is organized as follows. First of all, Section II reports the current state-of-the-art on homomorphic encryption-based data hiding in encrypted images (DHEI). Section III then describes with details the proposed method of DHEI based on the Paillier cryptosystem. Experimental results and comparisons with related work are provided in Section IV. Finally, the conclusion is drawn and future work is proposed in Section V.

II. PREVIOUS WORK ON HOMOMORPHIC ENCRYPTION-BASED DHEI

In the last few years, homomorphic encryption has attracted research attention, and different signal processing methods in the homomorphic encryption domain have been developed [2]–[5]. Since the operations can be performed directly in the encrypted domain, there is no need to decrypt an encrypted image before processing it. By this way, the user privacy and the original image integrity are protected. This property is very important in modern communication systems [4], such as in cloud computing. Most of classic signal processing techniques have been then adapted to be performed in the encrypted domain, such as the Discrete Fourier Transform [2], the Discrete Cosine Transform [3] and the Discrete Wavelet Transform [5]. Moreover, data hiding can be also performed in the homomorphic encryption domain in order to embed a secret message without image privacy invasion.

Public-key encryption-based methods of data hiding in encrypted images (DHEI), which are specific homomorphic encryption-based DHEI methods, are based on two different approaches for encryption. These are Paillier encryption [19] or learning with error (LWE) encryption [20].

On one hand, in 2014, Chen *et al.* have proposed the first Paillier encryption-based DHEI method [12]. In this

scheme, each pixel of an original image is divided into two parts: an even integer (composed by the seven most significant bits) and the least significant bit. Both parts are then encrypted and one bit of a secret message is embedded into a pair of neighboring pixels. During the decoding phase, by comparing all pairs of decrypted pixels, a receiver can obtain the whole secret message and reconstruct the original image. The main drawback of this method is an inherent overflow. Solutions have been then proposed to overcome this problem [13], [21]. As an improvement of the Chen *et al.* method [12], Shiu *et al.* adopted the concept of difference expansion into homomorphic encryption [13]. Furthermore, Wu *et al.* divided each unit of an original image into three parts according to the signal energy transfer principle [21]. These parts are then encrypted using the Paillier cryptosystem and one bit of the secret message can be embedded by processing the three encrypted parts. After decoding, the original image can be perfectly recovered and the embedded secret message is perfectly extracted as well. Note that these last described methods are based on reserving room before encryption (RRBE). The first homomorphic encryption-based approach of DHEI which vacates room after encryption (VRAE) has been proposed by Wu *et al.* [22]. The authors used the self-blinding property of the Paillier cryptosystem and value expansion in two algorithms, which one allows data extraction in the encrypted domain and the other one, in the clear domain (after decryption). Moreover, Zhang *et al.* also developed two different approaches with the same objectives as Wu *et al.* [14]. Indeed, an embedded secret message can be extracted in the encrypted domain in the lossless approach, and from the clear reconstructed image in the reversible and separable approach. Li and Li [23] suggested to exploit the homomorphic addition property of the Paillier cryptosystem and to resort to histogram shifting to perform the data hiding step. Xiang and Luo also proposed a separable homomorphic encryption-based method of DHEI involving the mirroring ciphertext group strategy [15]. Using such a technique, there is no pixel oversaturation in the clear domain after decryption, but the computational cost remains high. In 2019, Zheng *et al.* proposed a lossless data hiding method based on homomorphic cryptosystem which achieves a high embedding rate through efficient mapping and skillful use of expanded pixel values [17]. This method can realize a high embedding rate and low computational complexity, but it needs auxiliary data for the decoding step. Recently, Jiang and Pang have presented an improved implementation of the Paillier cryptosystem based on the use of the Chinese Remainder Theorem [24]. This new implementation significantly improves the rapidity of encryption and decryption operations. After encryption, authors then observed the encrypted pixel pair parity to embed bits of a secret message.

On the other hand, the first LWE-based DHEI method has been proposed in 2016 by Ke *et al.* [25]. Indeed, the authors explained that using a LWE algorithm allows them to obtain a strong security, a controllable redundancy for embedding and a simple computation. They then fixed the parameters for LWE encryption and described their multilevel DHEI approach based on the redundancy recoding in the encrypted domain. The main drawback of this method is that it is not perfectly separable. In [26], Xiong *et al.* dealt with this problem by introducing a modified somewhat LWE encryption, but which preserves correlations between the original image and its associated encrypted version. Theoretically, this makes it vulnerable to cryptanalysis.

III. PROPOSED DHEI METHOD

In this section, we introduce our proposed method of high capacity data hiding in encrypted images based on a homomorphic cryptosystem. An overview of this method is presented in Fig. 1. Using the Paillier cryptosystem and its homomorphic properties, a secret message can be embedded in an encrypted image which corresponds to a substitution of the least significant bits (LSB) of each pixel in the clear domain.

First of all, some explanations about the Paillier cryptosystem are given. We then describe the two steps of the encoding phase of our method: image encryption and message embedding. Finally, the decoding phase is detailed. As shown in Fig. 1, from the marked encrypted image, a marked image in clear can be obtained and the secret message can be losslessly extracted. Note that the marked image in clear is a very high quality version of the original image.

A. PAILLIER CRYPTOSYSTEM

In 1998, Paillier introduced a new homomorphic cryptosystem [19], which the different steps are described as follows. To generate the keys, choose p, q two prime numbers such that:

$$gcd(pq,\ (p-1)(q-1)) = 1. \tag{1}$$

Set n and λ such that:

$$n = pq \text{ and } \lambda = lcm((p-1),(q-1)). \tag{2}$$

Choose $g \in (\mathbb{Z}/n^2\mathbb{Z})^*$ such that:

$$\exists\, \mu \mid \mu\ = (L(g^{\lambda} \bmod (n^2)))^{-1} \bmod (n), \tag{3}$$

where $L(\cdot)$ is defined as:

$$L(x) = \frac{x-1}{n}, \text{ where } x \in \mathbb{N}^*. \tag{4}$$

Hence, the public key is (n, g), while the private key is (λ, μ).

A plaintext m, with $0 \leq m < n$ is then selected. To encrypt it, a random r is generated, with $r \in (\mathbb{Z}/n\mathbb{Z})^*$. Such an r guaranties the non deterministic property of the Paillier encryption scheme. The ciphertext c is:

$$c = \mathcal{E}(m) = g^m \times r^n \bmod (n^2), \tag{5}$$

where $\mathcal{E}(\cdot)$ is the Paillier encryption function.

From the ciphertext c, the initial message m is retrieved:

$$m = \mathcal{D}(c) = L(c^{\lambda} \bmod (n^2)) \times \mu \bmod (n), \tag{6}$$

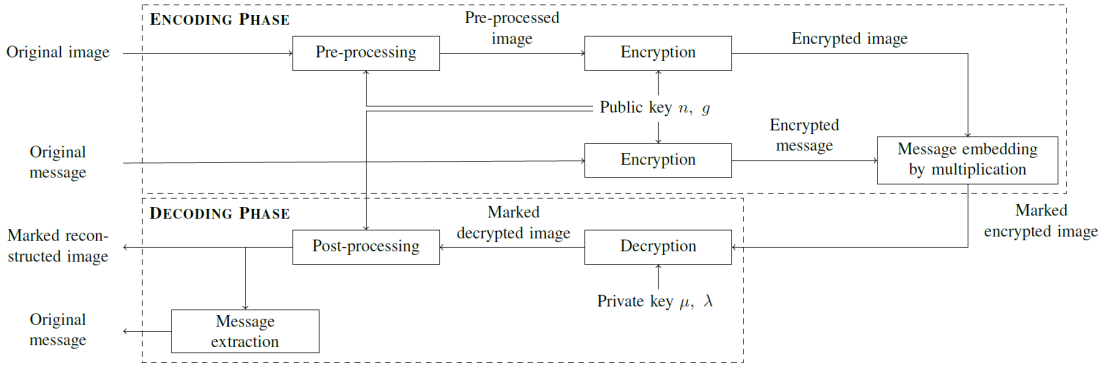

FIGURE 1: Overview of our proposed homomorphic DHEI method.

where $\mathcal{D}(\cdot)$ is the Paillier decryption function.

Moreover, the homomorphic property of the Paillier cryptosystem is such that the product of two ciphertexts decrypts to the sum of their corresponding plaintexts m_1 and m_2:

$$\mathcal{D}(\mathcal{E}(m_1) \times \mathcal{E}(m_2) \bmod (n^2)) = m_1 + m_2 \bmod (n). \quad (7)$$

B. IMAGE ENCRYPTION

During the encoding phase, as shown in Fig. 1, an original image is first separated into blocks of l pixels ($l \geq 1$). The goal is to embed k message bits into each block, with $0 \leq k \leq 8l$ since each pixel is encoded using 8 bits. In particular, to obtain a payload of 1 *bpp*, k has to be equal to l. In order to keep the same image format and size once the image is decrypted, the parameter n of the public key needs to be under the maximum possible value ($mpv = 2^{8l} - 1$) of a pixel block. Therefore, the key size increases as the pixel block size increases. Parameters p and q are thus chosen satisfying $gcd(pq, (p-1)(q-1)) = 1$ such that $n = pq$ remains relatively close but lower than mpv. In order to guarantee a high level of security, it is not always necessary to choose the maximum value n to ensure multiple choices for the keys.

For example, for a pixel block of size 1 pixel, mpv is equal to $2^8 - 1 = 255$. Pairs (p, q) that satisfy all the constraints are for example $(3, 83)$ and $(13, 19)$ and give respectively $n = 3 \times 83 = 249$ and $n = 13 \times 19 = 247$.

Each pixel block of the original image is pre-processed following the steps described in Algorithm 1. In Algorithm 1 the function *reo*$(\cdot)$ consists of reorganizing the block's bits. This reorganization is operated on each block, pixel bitplane by pixel bitplane, from the most to the least significant bits. As illustrated in Fig. 2, once the block is reorganized, the l MSB of the pixel block correspond to the MSB of each pixel (in dark blue) in the same order, so on and so forth for each pixel bitplane. Then note that the l LSB of the block are the LSB of each pixel (in red) starting from the right in Fig. 2.

To lose a minimum amount of information, a histogram shrinking function is then applied as a linear function on each pixel block value (considered as a value between 0 and mpv). Indeed, since the encryption scheme takes as an entry a value

Algorithm 1: Image block pre-processing and encryption.

input : Original image block of size l pixels (B_{im_orig}),
Public key (n, g),
Number of bits k to add into the pixel block
output: Encrypted pixel block ($B_{im\_encrypt}$)
$mpv \leftarrow 2^{8l} - 1$;
$B_{im\_process} \leftarrow$ *reo*(B_{im_orig});
$B_{im\_process} \leftarrow$ *rem*($B_{im\_process}, k$);
$B_{im\_process} \leftarrow$ *round*($B_{im\_process} \times \frac{n-1}{mpv}$);
$B_{im\_process} \leftarrow$ *rem*($B_{im\_process}, k$);
$r \leftarrow$ *random_invertible*();
$B_{im\_encrypt} \leftarrow g^{B_{im\_process}} \times r^n \mod (n^2)$;

modulus n, all the pixel block values that are greater than n are lost due to the overflow. After applying the histogram shrinking function, each pixel block value belonging to the range $[0, mpv]$ is then in the range $[0, n-1]$, as presented in Algorithm 1:

$$B_{im\_process} = rem(round(rem(reo(B_{im\_orig}), k) \times \frac{n-1}{mpv}, k), \quad (8)$$

where B_{im_orig} is the original pixel block, $B_{im\_process}$ the pre-processed block to encrypt and *rem*$(\cdot, k)$ the function to remove a given number of LSB (according to parameter k) as presented in Algorithm 1.

Using the Paillier cryptosystem, a multiplication in the encrypted domain corresponds to an addition in the clear domain. Therefore, the block's k LSB are put to zero before encryption to be able to embed the message bits into the pixel block. Thus, an addition of k bits becomes a LSB substitution in the clear domain. The pre-processed image is then encrypted with the public key. The Paillier encryption scheme is used in order to encrypt each pixel block value. Even if the same public key is kept for the whole image, for each pixel block a random $r \in (\mathbb{Z}/n\mathbb{Z})^*$ is generated. Such an r guaranties the non deterministic property of the

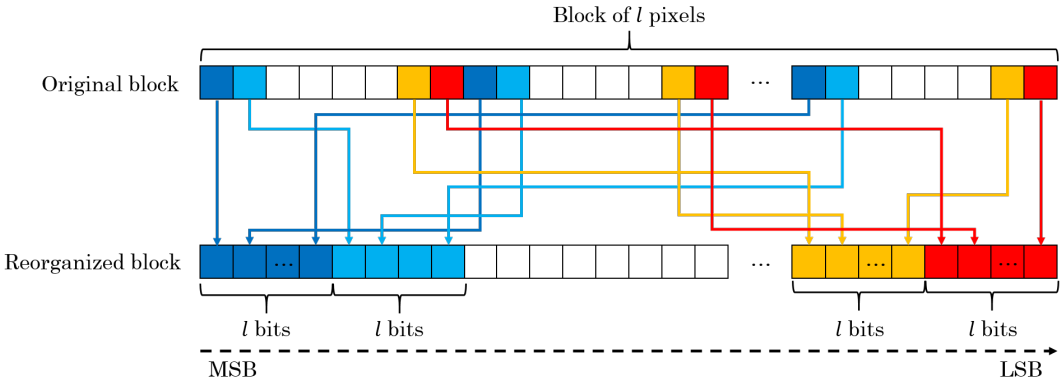

FIGURE 2: Block's bits reorganization by pixel bitplane from the most to the least significant bits.

Paillier encryption scheme, which means that two identical pixel blocks in the clear domain are not identical in the encrypted domain as presented in Algorithm 1, which is based on Eq. (5). In this way, there is no security leak due to the appearance of textured areas in the encrypted image. Each encrypted value is therefore a value modulus n^2, compared to n initially. This doubles the necessary number of bits to encrypt a value, which is why the expansion rate is considered to be 2.

C. MESSAGE ENCRYPTION AND EMBEDDING

In this part, we explain how to embed a secret message in the encrypted image using homomorphic properties of the Paillier cryptosystem. Note that the secret message may have been encrypted before this step (for example by performing a simple XOR operation with a pseudo-random bitstream) to preserve its content confidentiality. This ensures that someone who knows the private key cannot access to the secret message in clear.

The first step of the message embedding phase consists in message encryption, as presented in Algorithm 2, which is mainly based on Eq. (5). To encrypt the message, it is first divided into blocks of k bits, *i.e.* the number of LSB that have been put to zero during the image pre-processing. Each message block is then considered as a decimal value and is encrypted using the Paillier encryption scheme using the same public key as for image encryption. Note that this is not a security issue because the knowledge of the public key does not allow to access to the original image content. Indeed, it does not give any information on the private key, which is required for decryption.

The second step exploits the homomorphic properties of the Paillier cryptosystem for message embedding, as described in Algorithm 3. To embed the encrypted message into the encrypted image, each encrypted block of the message is embedded into one encrypted pixel block of the image by performing a multiplication modulus n^2 between the two:

$$B_{im\_mark\_encrypt} = B_{im\_encrypt} \times B_{msg\_encrypt} \bmod (n^2). \quad (9)$$

At the end of the encoding phase, as developed in Algorithm 3, a marked encrypted image is obtained, as presented in Fig. 1.

Algorithm 2: Message encryption.

input : Message block of size k bits (B_{msg_orig}), Public key (n, g)
output: Encrypted message block ($B_{msg\_encrypt}$)
$r \leftarrow$ *random_invertible()*;
$B_{msg\_encrypt} \leftarrow g^{B_{msg\_orig}} \times r^n \bmod (n^2)$;

Algorithm 3: Message embedding.

input : Encrypted pixel block ($B_{im\_encrypt}$), Encrypted message block ($B_{msg\_encrypt}$)
output: Marked encrypted image block ($B_{im\_mark\_encrypt}$)
$B_{im\_mark\_encrypt} \leftarrow B_{im\_encrypt} \times B_{msg\_encrypt} \bmod (n^2)$;

D. MESSAGE AND IMAGE RECOVERY

The decoding phase consists of two steps. During the first step, using the private key, the Paillier decryption function (Eq. (6)) is applied on each pixel block of the marked encrypted image to obtain a marked decrypted image:

$$B_{im\_mark} = L(B^{\lambda}_{im\_mark\_encrypt} \bmod (n^2)) \times \mu \bmod (n). \quad (10)$$

While the marked decrypted image contains some artifacts due to the original image pre-processing, it is still close to the original image. In order to obtain a very high quality marked image, as presented in Algorithm 4, a post-processing is then applied. With the function *round(·)*, a histogram stretch corresponding to the inverse histogram shrinking function is applied to retrieve the block values after reorganization. Block's bits are then reorganized back into pixels. Finally, the obtained marked reconstructed image has a very high quality compared to the original one and the secret message lays in the pixels' LSB values.

Algorithm 4: Image block decryption and post-processing.

input : Marked encrypted image block of size l pixels ($B_{im\_mark\_encrypt}$),
Private key (λ, μ),
Number of bits k embedded into the block
output: Marked reconstructed pixel block (B_{im_mark}),
Extracted message block (B_{msg})

$mpv \leftarrow 2^{8l} - 1$;
$B_{im\_mark} \leftarrow L(B^{\lambda}_{im\_mark\_encrypt} \bmod (n^2)) \times \mu \bmod (n)$;
$B_{msg} \leftarrow$ *extract_msg*(B_{im_mark}, k);
$B_{im\_mark} \leftarrow$ *round*$(B_{im\_mark} \times \frac{mpv}{n-1})$;
$B_{im\_mark} \leftarrow$ *reo*$^{-1}(B_{\text{im\_mark}})$;
$B_{im\_mark} \leftarrow$ *embed_k_LSB*$(B_{im\_mark}, B_{msg}, k)$;

Statistically half of the reconstructed pixels have the same values as the pixels of the original image. Some are not losslessly reconstructed, due to the round operation in Algorithm 1 and the message embedding phase. However, they are very close to the original values and finally the marked reconstructed image quality is very high.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The different results obtained with our applied method are discussed in this section. Throughout all of our experiments, the number of bits embedded into each pixel block is chosen to be equal to the pixel block size (*i.e.* $k = l$) giving a constant payload of 1 *bpp*. Note that for all the experiments, the messages are randomly generated and are always successfully reconstructed. In Section IV-A our method is applied to the *Lena* image and presented in detail. Section IV-B gives performance analysis on a large image database. Finally, comparisons with related work are provided in Section IV-C.

A. A FULL EXAMPLE

This method is first tested on the *Lena* image, as illustrated in Fig. 3.a. For this experiment, l and k are chosen equal, such that $l = k = 2$, which means that two bits of the secret message are embedded per two pixel blocks. In fact, with $l = 2$, mpv is equal to $2^{16} - 1 = 65535$, we have then to select a n value slightly lower to mpv in order to minimize distortion during the pre-processing step. For this full example, as parameter values of the Paillier cryptosystem, we then chose as public key $(n,\ g) = (64751,\ 120)$ (with $n = p \times q = 73 \times 887$) and as private key $(\lambda,\ \mu) = (31896,\ 63607)$. Note also that a different random r is generated for each block of the image and of the message during the encryption phase. This ensures the non deterministic property of the used cryptosystem. The results of the different steps of our proposed method are illustrated in Fig. 3.

As it can be seen on Fig. 3.b compared with Fig. 3.a, the original image is deteriorated when the histogram shrinking function is applied as described in Section III.2. Indeed, the PSNR value between the original image and the pre-processed image is equal to 32.97 dB and the SSIM is equal to 0.9950. The encrypted marked image, shown in Fig. 3.c, is encoded on 16-bits pixels because of the expansion rate is 2. This is due to the fact that the encrypted value is computed modulus n^2. We recall that the message embedding step does not lead to encrypted image size expansion or any overflow since it is a simple multiplication modulus n^2. We can also see that the marked reconstructed image illustrated Fig. 3.d is indeed very close to the original image (with a PSNR of 51.04 dB and a SSIM of 0.9983) while bits of the secret message lay in the LSB pixel values.

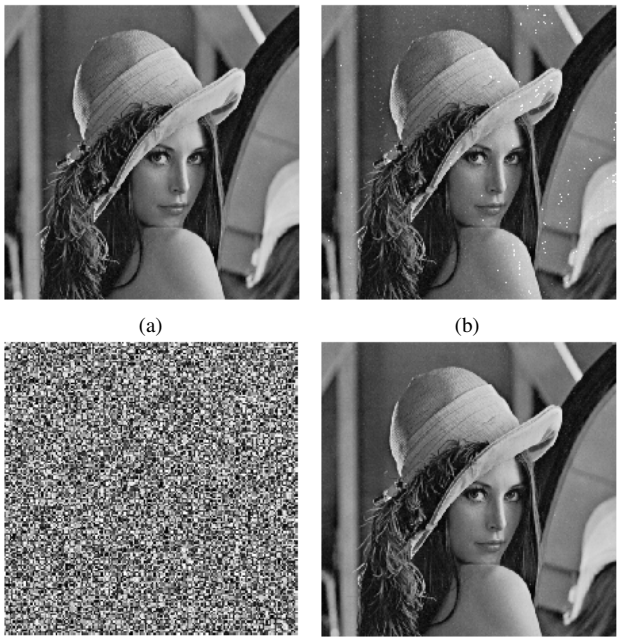

FIGURE 3: Experiment using our method with $l = 2$, with an embedding rate equal to 1 *bpp*: a) Original *Lena* image (512× 512 pixels, 256 greyscale levels), b) Pre-processed *Lena* image, using our method with pixel blocks of size 2 (PSNR with the original image 32.97 dB, SSIM = 0.9950), c) Marked encrypted image with 2 embedded message bits per block, d) Marked reconstructed image, PSNR = 51.04 dB, SSIM = 0.9983.

B. PERFORMANCE ANALYSIS ON A LARGE IMAGE DATABASE

The method is then applied on 1,000 sample images of the BOWS-2 database [27] for different block sizes as seen in Table 1. These data confirm what is observed in Fig. 3.a and Fig. 3.b. Indeed, the pre-processing which deteriorates the most the images is compensated once the post-processing is applied. We can also observe that the message embedding does not modify the image too much. We can notice that the block size (and therefore the key size) does not have a huge impact on the quality of the marked reconstructed image. Indeed, as it can be seen, in Table 1 the PSNR between the

Statistical indicator	I and I_p		I_p and I_{md}		I and I_{md}		I and I_{mr}	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Min.	30.16	0.9849	51.12	0.9870	30.70	0.9910	49.06	0.9892
1st Qu.	33.36	0.9961	51.14	0.9972	34.08	0.9936	50.38	0.9938
Median	34.22	0.9967	51.14	0.9979	35.00	0.9948	50.45	0.9953
Mean	34.56	0.9967	51.14	0.9978	35.38	0.9948	50.45	0.9953
3rd Qu.	35.30	0.9973	51.15	0.9984	36.15	0.9960	50.53	0.9968
Max.	44.72	0.9986	51.17	0.9998	46.84	0.9985	51.10	0.9995

(a) With $l = 1$.

Statistical indicator	I and I_p		I_p and I_{md}		I and I_{md}		I and I_{mr}	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Min.	21.34	0.6769	51.11	0.8187	21.31	0.6749	49.53	0.9855
1st Qu.	30.83	0.8624	51.14	0.9860	30.79	0.8602	51.13	0.9973
Median	32.98	0.9003	51.14	0.9916	32.96	0.8987	51.14	0.9979
Mean	32.82	0.8978	51.14	0.9883	32.80	0.8963	51.14	0.9979
3rd Qu.	35.09	0.9404	51.15	0.9948	35.07	0.9391	51.15	0.9986
Max.	51.13	0.9959	51.17	0.9997	51.15	0.9946	54.64	0.9998

(b) With $l = 2$.

TABLE 1: Comparison of the different states of our DHEI method using PSNR in dB and SSIM (I = the original image, I_p = the pre-processed image, I_{md} = the marked decrypted image, I_{mr} = the marked reconstructed image) applied on 1,000 sample images from the BOWS-2 database [27].

original image (I) and the marked reconstructed image (I_r) remains around $51\ dB$. Even better, the reconstructed image quality increases with the key size. **This means that the more secure the method is, the better the quality of the reconstructed image is.**

C. COMPARISONS WITH PREVIOUS WORK

In Table 2, we compare our method with the most pertinent DHEI methods based on the Paillier cryptosystem and using a n public key of 512 bits [12]–[15]. With our proposed method, this gives us 64 pixel block size. We can first notice that our expansion rate is really small compared to the other methods. Indeed, no matter what the key size is, the expansion rate of our method is always 2. Therefore, with our method, it is possible to use a larger key size such as $1,024$ bits or $2,048$ bits, as recommended in security requirements for Paillier cryptosystem. The larger the public key n, the larger the number of pixels per block, whereas the other methods encrypt pixel per pixel no matter what the key size is. This is very useful for multimedia data because their initial size is already significant and so data expansion has a huge impact on the size, especially when the encrypted image is loaded onto a cloud. Moreover, we can also see that the payload of our method is better than the ones for other methods. Indeed, a payload of 1 *bpp* is achieved with our method whereas the payload obtained using Chen *et al.* [12], Shiu *et al.* [13] or Xiang and Luo [15] DHEI schemes is always less or equal to 0.5 *bpp*. Although the payload of the method of Zhang *et al.* [14] is comparable to ours, the expansion rate with this scheme is huge (factor 128). On top of that the payload of our method does not depend on the original image content since our scheme relies on LSB substitution and does not need to make more free space to store additional data (such as a number of shifted pixels).

In terms of quantitative comparisons (payload and PSNR) with [12]–[15], we present the results on the well known images of *Lena* and *Airplane* in Table 3 and display the average results on larger image databases in Fig. 4. These average results were calculated according to the results presented in each article on databases between 6 and 50 images and according to a privileged criterion (payload or PNSR). First of all, we can see that with our method, the achieved payload is larger than the others in all cases, as previously presented in Table 2. In addition, the marked reconstructed image quality, in terms of PSNR, is also increased compared to other methods. Indeed the marked reconstructed image is very close to a LSB substitution-based marked image. Therefore, the PSNR remains around the average PSNR of these kinds of methods which is $51\ dB$ with a payload of 1 *bpp*. None of the other methods obtain such results because the secret message is embedded in a way that it does not only involve LSB values. Indeed, as presented in Table 3 and in Fig. 4, the PSNR values are smaller than $40\ dB$ no matter what the used approach is.

Moreover, some papers of other state-of-the-art methods – for example Zheng et al. [17] – have provided a full computational complexity analysis of their proposed algorithm. Note that with the method we propose, as the image is processed by block and there is a size expansion rate of 2, the computational complexity is of the order of $O(N^2)$.

V. CONCLUSION

In this paper, we described a new efficient method of data hiding in encrypted images using the Paillier cryptosystem. After the original image pre-processing and encryption, a secret message can be embedded performing a multiplication in the encrypted domain, which results in a LSB substitution in the clear domain. Experimental results put in evidence that unlike most recent state-of-the-art methods, our proposed scheme **does not expand a lot the original image size**.

Method	Message recovery in the ED	Message recovery in the CD	Max. payload (*bpp*)	Expansion rate
Chen *et al.* [12]	no	yes	0.5	256
Shiu *et al.* [13]	no	yes	0.5	128
Zhang *et al.* [14]	yes	yes	0.97	128
Xiang and Luo [15]	yes	yes	0.25	128
Ours	**no**	**yes**	**1**	**2**

TABLE 2: Comparison of the different DHEI methods using a n public key of 512 bits on 256 greyscale pixels images (ED = Encrypted Domain, CD = Clear Domain).

Image	Method	Payload (*bpp*)	PSNR (dB)
Lena	Chen *et al.* [12]	0.5	39.83
	Shiu *et al.* [13]	0.5	35.69
	Zhang *et al.* [14]	0.3	36.30
	Xiang and Luo [15]	0.25	32.63
	Ours	**1**	**51.04**
Airplane	Chen *et al.* [12]	0.5	39.84
	Shiu *et al.* [13]	0.5	32.54
	Zhang *et al.* [14]	0.3	32.00
	Xiang and Luo [15]	0.25	34.90
	Ours	**1**	**51.16**

TABLE 3: Performance comparison between our proposed method with $l = 2$ and previous work for *Lena* and *Airplane* images.

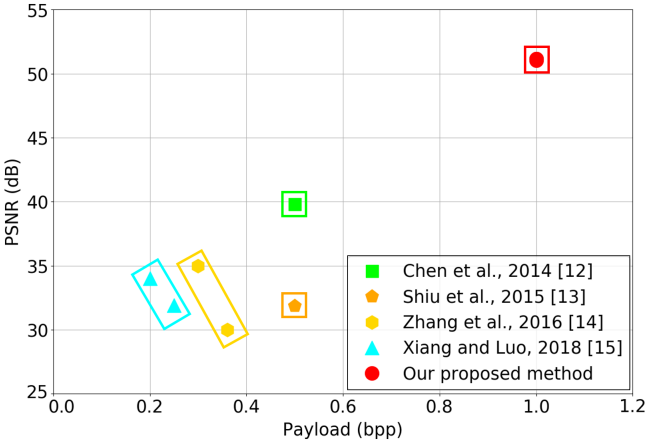

FIGURE 4: Performance comparison – in terms of payload and marked image quality in terms of PSNR – between our proposed method and previous work (average results based on images presented in each article on databases between 6 and 50 images and according to a privileged criterion (payload or PNSR)).

Indeed, whatever the key size, the expansion rate is 2. In addition, we achieve a strong security level and an interesting payload value (1 *bpp*). Note also that, even with this high payload value, the marked reconstructed image is almost the same as the original image, as indicated by a PSNR value larger than 50 dB and a SSIM close to 1. Finally, as bits of the secret message are embedded into each pixel block in the scanning order, it is possible to perform multiple messages embedding. As an application example, we can consider an encrypted image sent across a network. The IP addresses of each node during the routing can be embedded independently from each other.

Recently published papers proposed to embed a secret message using homomorphic encryption in multimedia files other than images, such as HEVC videos for example [28]. In our case, in future work, we are interested in applying the proposed method to 3D meshes. Indeed, we think that the pixel block pre-processing step can be easily extended to 3D floating point vertices using 3D selective encryption. To the best of our knowledge, it would be one of the first uses of homomorphic encryption for data hiding in encrypted 3D meshes.

ACKNOWLEDGMENT

We would like to thank the financial support of the ANR-16-DEFA-0001 OEIL (statistiques rObustEs pour l'apprentIssage Léger) research project of the French ANR/DGA challenge DEFALS (DEtection de FALSifications dans des images).

REFERENCES

[1] W. Trappe and L. C. Washington, Introduction to cryptography with coding theory. Pearson Education India, 2006.
[2] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," IEEE Transactions on Information Forensics and Security, vol. 4, no. 1, pp. 86–97, 2009.
[3] ——, "Composite signal representation for fast and storage-efficient processing of encrypted signals," IEEE Transactions on Information Forensics and Security, vol. 5, no. 1, pp. 180–187, 2009.
[4] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 82–105, 2012.
[5] P. Zheng and J. Huang, "Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain," IEEE Transactions on Image Processing, vol. 22, no. 6, pp. 2455–2468, 2013.
[6] 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.

[7] X. Zhang, "Reversible data hiding with optimal value transfer," IEEE Transactions on Multimedia, vol. 15, no. 2, pp. 316–325, 2012.
[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] 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.
[10] 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.
[11] 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.
[12] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," Journal of Visual Communication and Image Representation, vol. 25, no. 5, pp. 1164–1170, 2014.
[13] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," Signal Processing: Image Communication, vol. 39, pp. 226–233, 2015.
[14] 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.
[15] S. Xiang and X. Luo, "Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 11, pp. 3099–3110, 2018.
[16] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 441–452, 2016.
[17] S. Zheng, Y. Wang, and D. Hu, "Lossless data hiding based on homomorphic cryptosystem," IEEE Transactions on Dependable and Secure Computing, 2019.
[18] N. Zhou, M. Zhang, H. Wang, Y. Ke, and F. Di, "Separable reversible data hiding scheme in homomorphic encrypted domain based on NTRU," IEEE Access, pp. 1–1, 2020.
[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 1999, pp. 223–238.
[20] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," Journal of the ACM (JACM), vol. 56, no. 6, pp. 1–40, 2009.
[21] X. Wu, B. Chen, and J. Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer," Journal of Visual Communication and Image Representation, vol. 41, pp. 58–64, 2016.
[22] H.-T. Wu, Y.-M. Cheung, and J. Huang, "Reversible data hiding in Paillier cryptosystem," Journal of Visual Communication and Image Representation, vol. 40, pp. 765–771, 2016.
[23] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," Signal Processing, vol. 130, pp. 190–196, 2017.
[24] C. Jiang and Y. Pang, "Encrypted images-based reversible data hiding in Paillier cryptosystem," Multimedia Tools and Applications, vol. 79, no. 1-2, pp. 693–711, 2020.
[25] Y. Ke, M. Zhang, and J. Liu, "Separable multiple bits reversible data hiding in encrypted domain," in International Workshop on Digital Watermarking. Springer, 2016, pp. 470–484.
[26] L. Xiong, D. Dong, Z. Xia, and X. Chen, "High-capacity reversible data hiding for encrypted multimedia data with somewhat homomorphic encryption," IEEE Access, vol. 6, pp. 60 635–60 644, 2018.
[27] P. Bas and T. Furon, "Image database of BOWS-2," http://bows2.ec-lille.fr/, accessed: 2019-06-22.
[28] B. Guan, D. Xu, and Q. Li, "An efficient commutative encryption and data hiding scheme for HEVC video," IEEE Access, vol. 8, pp. 60 232–60 245, 2020.

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 & Systems for Video Technology and IEEE Trans. on Dependable and Secure Computing.

MANON VIALLE received her B.S. of engineering from Grenoble INP ENSIMAG, France in 2018. She is currently persuing both a M.S in engineering and a M.S. in applied mathematics with a specialty in Modeling, Scientific computing and Image Analysis from Grenoble INP ENSIMAG, France. After she graduates, she is interested in persuing a Ph. D. in the same field.

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.