Image Denoising in the Encrypted Domain

Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González
Signal Theory and Communications Department
University of Vigo
36310 Vigo, Spain
{apedrouzo,troncoso,fperez}@gts.uvigo.es

***Abstract*—The increasing advance of Cloud-based solutions brings about serious privacy problems when outsourcing images for their processing in untrusted environments. One of the fundamental privacy-aware image manipulations that can be outsourced is denoising, an ubiquitous signal processing primitive with a broad set of applications. Traditional Signal Processing in the Encrypted Domain solutions cannot efficiently address this problem, as they require interactive protocols in order to cope with polynomial operations and comparisons at the same time. We propose methods based on 2-RLWE (Ring Learning with Errors) to efficiently perform the whole image denoising operation on encrypted images in a fully non-interactive way; we show how to combine homomorphic polynomial operations and thresholding without involving decryption or interaction, therefore enabling fully unattended encrypted image processing. We evaluate our solutions for real image sizes and strict security parameters, showing their practicality and feasibility.**

***Index Terms*—Image Encryption, Lattice Cryptography, Image Denoising, Homomorphic Processing**

I. Introduction

The field of Secure Signal Processing (SSP), also called Signal Processing in the Encrypted Domain (SPED), was born as a solution to efficiently preserve the privacy on those signal processing scenarios dealing with sensitive signals [1]. In order to address these privacy-preserving problems, homomorphic encryption and, specially, additive schemes like the Paillier cryptosystem [2], have been widely employed for secure signal processing primitives. However, solutions resorting to the Paillier cryptosystem present a very high cipher expansion, and despite the proposal of techniques like packing and unpacking to mitigate this effect [3], the cipher expansion becomes a serious problem when dealing with images. Consequently, recent works have introduced new solutions based on lattice cryptography which can efficiently deal with multidimensional signals, and, in particular, images [4], [5].

The problem of image (or signal) denoising is ubiquitous in signal processing and has a broad set of applications. It appears in any possible scenario looking for the best possible estimate of a signal from a noisy version. Nowadays, outsourced services are increasingly used, so it is not hard to imagine a situation where someone wants to obtain an enhanced version of a noisy signal by relying on a third party to perform the task, therefore incurring in a threat for the privacy of the involved sensitive information. The approaches presented in [5] to deal with images are not enough to tackle the problem non-interactively, requiring interactive secure protocols to obtain a feasible solution. Some current proposals for encrypted domain processing target unattended processing, without resorting to interactive secure protocols [6], but they are limited to polynomial operations.

We can find some recent works dealing with privacy-preserving denoising: Hu *et al.* [7] propose an scheme for performing nonlocal means (NLM) denoising of encrypted images, and Saghaian *et al.* [8] propose a scheme for wavelet denoising resorting to secret sharing. However, the former does not deal with wavelet denoising algorithms (it performs a filtering operation and leaks pixel distances) and the latter is based on interactive protocols (secret sharing).

This work proposes a new solution to the problem of denoising of an image (or a more general multidimensional signal) in the encrypted domain in a fully unattended way. For this purpose, we solve the problem of homomorphically computing both filtering and threshold operations in a sole round without resorting to the intervention of the secret key owner.

Main Contributions: We briefly summarize the main ideas and contributions of our work:

- We introduce a practical scheme for homomorphically denoising images in the encrypted domain. The results can be easily adapted to work with either uni- or multi-dimensional signals.
- The main advantage of our scheme is that it avoids interactive protocols. Therefore, the secret key owner does not need to participate in the middle of the encrypted computation to complete the denoising process.
- We show how to adapt the structure of modern lattice-based cryptosystems to efficiently compute a wavelet transform.
- In the same round, we show how to homomorphically perform the threshold of encrypted values without the need of intermediate decryption or interaction with the secret key owner.

Notation and structure: We represent vectors and matrices by boldface lowercase and uppercase letters respectively. Polynomials are denoted with regular lowercase letters, omitting the polynomial variable (e.g., a instead of $a(z)$) whenever there is no ambiguity. We indicate the variable of the polynomial rings to avoid confusion between univariate and bivariate rings; i.e., $R_q[z] = \mathbb{Z}_q[z]/(z^{n_z}+1)$ denotes the polynomial ring in the variable z modulo $z^{n_z}+1$ with coefficients in $\mathbb{Z}_q$. Analogously, $R_q[x,y] = (R_q[x])[y]/(y^{n_y}+1)$ is the bivariate polynomial ring with coefficients in $\mathbb{Z}_q$ reduced modulo $x^{n_x}+1$ and $y^{n_y}+1$ (n_z, n_x and n_y are powers of two). Finally, $\boldsymbol{A} \otimes \boldsymbol{B}$ is the Kronecker product between the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

The rest of the paper is organized as follows: Section II revisits some relevant concepts related to the used 2-RLWE (Ring Learning with Errors) based cryptosytem and a brief overview of the image denoising problem. Section III introduces the main contributions of this work, including the description of the proposed scheme for encrypted image denoising. Section IV discusses some practical aspects aimed towards an efficient implementation of the proposed scheme, and evaluates its security and efficiency.

II. Preliminaries

This section revises the lattice-based cryptosystem chosen to exemplify our schemes, together with its main parameters and primitives. It also includes a brief explanation of the image denoising problem.

A. 2-RLWE based Cryptosystem

Firstly, we revisit a slightly adapted definition of the m-RLWE problem [4], [5] particularized to our bivariate case:

Definition 1 (2-RLWE problem [4], [5]): Given a polynomial ring $R_q[x,y] = (\mathbb{Z}_q[x,y]/(x^{n_x}+1))/(y^{n_y}+1)$ and an error distribution

978-1-5090-1138-4/16/$31.00 ©2016 IEEE

TABLE I
2-RLWE BASED CRYPTOSYSTEM: PARAMETERS AND PRIMITIVES

Parameters		
Let $R_t[x,y] = (\mathbb{Z}_t[x,y]/(x^{n_x}+1))/y^{n_y}+1$ be the cleartext ring and $R_q[x,y] = (\mathbb{Z}_q[x,y]/(x^{n_x}+1))/y^{n_y}+1$ the ciphertext one. The noise distribution $\chi[x,y]$ in $R_q[x,y]$ takes its coefficients from a spherically-symmetric truncated i.i.d Gaussian $\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})$; q is such that $q \equiv 1 \bmod 2n$ where $n = n_x n_y$, and $t < q$ is relatively prime to q.		
Cryptographic Primitives		
SH.KeyGen	Process	$s, e \leftarrow \chi[x,y]$, $a_1 \leftarrow R_q[x,y]$ $sk = s$ and $pk = (a_0 = -(a_1 s + te), a_1)$
SH.Enc	Input	$pk = (a_0, a_1)$ and $m \in R_t[x,y]$
	Process	$u, f, g \leftarrow \chi[x,y]$ and the fresh ciphertext is $\boldsymbol{c} = (c_0, c_1) = (a_0 u + tg + m, a_1 u + tf)$
SH.Dec	Input	sk and $\boldsymbol{c} = (c_0, c_1, \ldots, c_{\gamma-1})$
	Process	$m = \left(\left(\sum_{i=0}^{\gamma-1} c_i s^i\right) \bmod q\right) \bmod t$
SH.Add	Input	$\boldsymbol{c}_0 = (c_0, \ldots, c_{\beta-1})$ and $\boldsymbol{c}_1 = (c'_0, \ldots, c'_{\gamma-1})$
	Process	$\boldsymbol{c}_{add} = (c_0 + c'_0, \ldots, c_{\max(\beta,\gamma)-1} + c'_{\max(\beta,\gamma)-1})$
SH.Mult	Input	$\boldsymbol{c}_0 = (c_0, \ldots, c_{\beta-1})$ and $\boldsymbol{c}_1 = (c'_0, \ldots, c'_{\gamma-1})$
	Process	Using a symbolic variable v their product is $\left(\sum_{i=0}^{\beta-1} c_i v^i\right) \cdot \left(\sum_{i=0}^{\gamma-1} c'_i v^i\right) = \sum_{i=0}^{\beta+\gamma-2} c''_i v^i$

$\chi[x,y] \in R_q[x,y]$ that generates small-norm random polynomials in $R_q[x,y]$, 2-RLWE relies upon the computational indistinguishability between samples $(a_i, b_i = a_i s + t \cdot e_i)$ and (a_i, u_i), where a_i, $u_i \leftarrow R_q[x,y]$ are chosen uniformly at random from the ring $R_q[x,y]$, while $s, e_i \leftarrow \chi[x,y]$ are drawn from the error distribution, and t is relatively prime to q.

The primitives and parameters of the 2-RLWE cryptosystem are described in Table I. Its ciphertexts are composed of at least 2 polynomial elements from the ring $R_q[x,y]$; the cryptosystem allows for additions (the smallest ciphertext is previously zero-padded) and multiplications on these tuples of polynomials, whose size increases after each multiplication. They can be brought back to the original size by resorting to a relinearization operation.

The security of the cryptosystem is based on the hardness of the 2-RLWE problem, which holds due to the hardness of reducing the n-dimensional lattices ($n = n_x n_y$) generated by the secret key. The applicable security reductions and proofs can be found in [4]. Further details about possible attacks are discussed in Section IV-A.

We choose this cryptosystem as it enables us to encrypt 2-dimensional messages in only one ciphertext, instead of encrypting each coefficient in a different ciphertext. It also enables efficient bivariate negacyclic linear convolutions with only one ciphertext multiplication at the cost of a small overhead (we refer the reader to [5] for a more detailed comparison between homomorphic cryptosystems when dealing with images). This overhead is caused by the use of the ring $\mathbb{Z}_q$ for the polynomial coefficients of the ciphertexts instead of $\mathbb{Z}_t$, where $q > t$. In order to correctly compute D consecutive products and A sums over the same ciphertext, the needed q for correct decryption is lower-bounded by

$$q \geq 4(2t\sigma^2\sqrt{n_x n_y})^{D+1}(2n_x n_y)^{D/2}\sqrt{A}. \quad (1)$$

Our proposed approach involves a multiplication tree with a determined number of levels to achieve a logarithmic complexity. Therefore, we work with a scale-invariant version [9] of the 2-RLWE cryptosystem, where D in eq. (1) represents the number of levels of the multiplication tree.

The relinearization [6], [10] is commonly used to reduce the size of the encryptions after a multiplication, to transform a ciphertext $c = (c_0, c_1, c_2)$ into $c^{relin} = (c_0^{relin}, c_1^{relin})$. This technique can also be leveraged to change the underlying secret key of the ciphertexts, and [6] proposes an extension which enables certain encrypted operations like changes on the sampling rate and element-wise multiplications, at the cost of a small computational overhead. We revisit these techniques for our proposed scheme in Section III-C1, and we refer the reader to [6] for more details on how to implement this extended relinearization.

B. Basic Structure of an Image Denoising Scenario

This section briefly introduces the general scheme of the nonlinear image denoising method which we later perform in the encrypted domain (see Section III).

There are several methods to perform the denoising of one image [11]; we resort here to the use of a wavelet transform to compact the energy of the image in a few values [12]. As the wavelet transform is an orthonormal transformation, the noise distribution is invariant after computing it, and therefore, we have two main components in the transformed domain: a) the signal component, with most of its energy compacted in a few values, and b) the noise distribution component, typically considered Gaussian noise, which is invariant after the transformation.

Hence, in order to separate the two components, a thresholding operation in the transformed domain can preserve the signal information while discarding most of the noise. Afterwards, we can compute the inverse wavelet transform to recover the estimated image. Figure 1 depicts a basic scheme of the clear-text image denoising process.

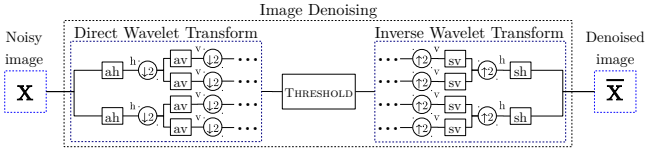

Fig. 1. Basic structure of the image denoising method.

Figure 1 shows the different components of an image denoising method based on wavelet transform. Both direct and inverse wavelet transforms are typically implemented by means of filter banks where a/s and v/h stand respectively for analysis/synthesis and vertical/horizontal filters; and $\downarrow 2$ ($\uparrow 2$) represents downsampling (upsampling) by a factor of two. The threshold operation performs the element-wise threshold of the different transformed coefficients.

III. PROPOSED SCHEME

This section introduces the proposed scheme for encrypted image denoising and details its main blocks. First, we show the general structure of the scheme and the purpose of each component. Afterwards, we focus on the two main parts of the scheme: a) The encrypted wavelet transform (both direct and inverse transforms), and b) the encrypted thresholding in the wavelet domain. We reiterate that we exemplify the scheme with images, but the results can be seamlessly adapted to work with higher dimensional signals [5].

A. General Overview

We exemplify the denoising operation with a typical nonlinear scheme that leverages the properties of the wavelet transform to compact the energy of the signal in a few values while keeping the energy of the noise spread through all the coefficients. This allows for separating noise and signal through a thresholding operation in the wavelet transformed domain. Currently, this problem can only be tackled efficiently in a privacy-preserving manner by resorting to interactive protocols. Our main focus is on an unattended solution which completely avoids interaction, therefore overcoming the need of intervention of the secret key owner during the process.

This paradigm introduces many challenges on the different parts of the process, the hardest one comprising the combination of both

polynomial and thresholding operations in the encrypted domain without the help of the secrey key owner at each step.

Figure 2 depicts the general structure of our proposed solution for encryped image denoising. First, we rely on the cryptosystem presented in [5] to work with encrypted images, and we apply a light-weight pre-/post-processing [6] to enable a homomorphism with the cyclic convolution when multiplying two ciphertexts (see Section III-B1a). The remaining blocks correspond to the homomorphic computation of the bivariate (direct and inverse) wavelet transform and the homomorphic threshold of each coefficient in the transformed domain. The following sections explain the details of these blocks.

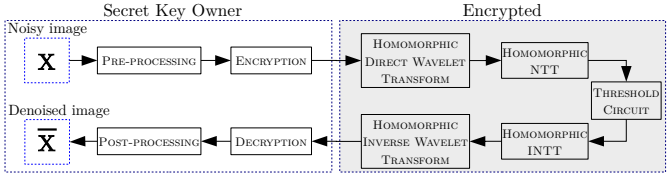

Fig. 2. Structure of the proposed encrypted image denoising method.

B. Homomorphic Wavelet Transform by means of filter banks

This section describes the homomorphic execution of the first and last blocks from Figure 2 (direct and inverse wavelet transforms). For the sake of efficiency, we resort to the filter bank implementation of the wavelet transform, which uses a matrix transformation for the i-th stage of the bivariate case as follows

$$\boldsymbol{W}_i = \boldsymbol{W}_i^{(x)} \otimes \boldsymbol{W}_i^{(y)} = \begin{bmatrix} \boldsymbol{D}_i^{(x)} \boldsymbol{A}_{l_i}^{(x)} \\ \boldsymbol{D}_i^{(x)} \boldsymbol{A}_{h_i}^{(x)} \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{D}_i^{(y)} \boldsymbol{A}_{l_i}^{(y)} \\ \boldsymbol{D}_i^{(y)} \boldsymbol{A}_{h_i}^{(y)} \end{bmatrix},$$

where matrix $\boldsymbol{D}_i^{(z)}$ downsamples the input vectors of the i-th stage by a factor of two in the dimension z, and $\boldsymbol{A}_{l_i}^{(z)}, \boldsymbol{A}_{h_i}^{(z)}$ represent the circulant matrices which correspond, respectively, to the low-pass and high-pass analysis filters of the first stage in dimension z.

Analogously, we can define the inverse transform as $\boldsymbol{W}_i^{-1} = \left(\boldsymbol{W}_i^{(x)}\right)^{-1} \otimes \left(\boldsymbol{W}_i^{(y)}\right)^{-1}$ where $\left(\boldsymbol{W}_i^{(z)}\right)^{-1} = \begin{bmatrix} \boldsymbol{S}_{l_i}^{(z)} \boldsymbol{U}_i^{(z)} & \boldsymbol{S}_{h_i}^{(z)} \boldsymbol{U}_i^{(z)} \end{bmatrix}$ with $\boldsymbol{U}_i^{(z)} = \left(\boldsymbol{D}_i^{(z)}\right)^T$ and the circulant matrices $\boldsymbol{S}_{l_i}^{(z)}, \boldsymbol{S}_{h_i}^{(z)}$ are, respectively, the synthesis low-pass and high-pass filters of the i-th stage for perfect reconstruction (i.e., $\boldsymbol{W}_i^{-1}\boldsymbol{W}_i = \boldsymbol{I}_{N^{(i)}}$ with $N^{(i)} = \frac{N_x N_y}{4^{i-1}}$).

Finally, this process is recursively applied for the four outputs at each stage of the filter bank.

In light of this structure, the main needed homomorphic operations under encryption are a) block-circulant matrix operations (multivariate cyclic convolutions), and b) changes on the sampling rate. The following sections detail the process to achieve these operations by preserving the multivariate structure of the images.

1) Homomorphic Bivariate Cyclic Convolutions: The filter bank implementation of the (direct or inverse) wavelet transform for images involves a total of 4^i filtering operations in the i-th stage. In general, when working with m-dimensional signals, the i-th stage will need a total of 2^{im} filtering operations. In order to securely and efficiently compute these operations we combine two contributions:

- We resort to the multivariate cryptosystem in [5] to encrypt each image in only one ciphertext and to enable encrypted multidimensional linear and negacylic convolutions (see Section II).
- We adapt the techniques from [6] for our multivariate case, in such a way that with a lightweight pre-/post-processing (negligible with respect to the encryption and decryption primitives) of the images before (after) encryption (decryption), we can homomorphically perform multivariate cyclic convolutions.

a) Pre-/Post-processing: In [6], the authors enable homomorphic cyclic convolutions between two one-dimensional signals of length N by performing an element-wise multiplication of both signals with $(-1)^{l/N}$ for $l = \{0, \ldots, N-1\}$ before encryption. The clear-text output of the cyclic convolution can be recovered by multiplying the pre-processed encryptions, decrypting the result and applying an element-wise multiplication with $(-1)^{-l/N}$ for $l = \{0, \ldots, N-1\}$. It is worth noting that, in order for this scheme to be valid, $(-1)^{1/N}$ has to be an element of $\mathbb{Z}_t$, that is, we must be able to find a $2N$-th root of unity in $\mathbb{Z}_t$.

We present a modified version of this pre-/post-processing that transforms the homomorphism on bivariate negacylic convolutions into bivariate cyclic convolutions. Therefore, if we consider two 2-dimensional signals $w[l_x, l_y]$ and $h[l_x, l_y]$ of length N_x and N_y in each dimension (both powers of two), our method works as follows:

- First, we assume the existence of $2N_x$-th and $2N_y$-th roots of unity in $\mathbb{Z}_t$, denoted α_x and α_y (they can be efficiently found).
- We pre-process the signals before encrypting them:

$$w'[l_x, l_y] = w[l_x, l_y] \left(\alpha_x^{l_x} \otimes \alpha_y^{l_y}\right),$$
$$h'[l_x, l_y] = h[l_x, l_y] \left(\alpha_x^{l_x} \otimes \alpha_y^{l_y}\right),$$

  where $l_x = 0, \ldots, N_x - 1$ and $l_y = 0, \ldots, N_y - 1$.
- Analogously, as described in [6], we can compute $v'(x, y)$ under encryption with only one ciphertext product modulo the two functions $x^{N_x} + 1$ and $y^{N_y} + 1$:

$$v'(x, y) = \left(w'(x, y)h'(x, y) \bmod x^{N_x} + 1\right) \bmod y^{N_y} + 1.$$

- Finally, the decrypted signal $v'[l_x, l_y]$ is post-processed:

$$v[l_x, l_y] = v'[l_x, l_y] \left(\alpha_x^{-l_x} \otimes \alpha_y^{-l_y}\right).$$

This approach can be easily extended to the multivariate case. Therefore, considering m-dimensional signals (i.e., $h[l_1, \ldots, l_m]$ where $l_i = 0, \ldots, N_i - 1$) with a length of N_i (all of them powers of two) in each dimension, let α_i be $2N_i$-roots of unity for $i = 1, \ldots, m$; the pre-processing and post-processing vectors are $\left(\bigotimes_{i=1}^m \alpha_i^{l_i}\right)$ and $\left(\bigotimes_{i=1}^m \alpha_i^{-l_i}\right)$ respectively.

2) Homomorphic Downsampling and Upsampling: This section addresses the implementation of downsampling/upsampling steps in the filter bank. For simplicity, we employ here univariate polynomials of n coefficients; we could extend this change of rate to the bivariate case by resorting to the Kronecker product, as done in previous sections. The structure of the filter bank (see Figure 1) requires a change in the sampling rate at each filter: a) one downsampling by a factor of two after each analysis filter ($\boldsymbol{D}_i^{(z)}$), and b) one upsampling by a factor of two before each synthesis filter ($\boldsymbol{U}_i^{(z)}$).

The required upsampling operation of a signal $x(z) \bmod z^n + 1$ represented as a polynomial can be seen as a scaling of the independent variable, $x(z^2) \bmod z^{2n} + 1$; conversely, the downsampling operation yields $x(z^{\frac{1}{2}}) \bmod z^{\frac{n}{2}} + 1$ by discarding the coefficients of the non integer exponents of z.

Hence, for a ciphertext $c = (c_0, c_1)$ with the corresponding $((c_0 + c_1 s) \bmod q) \bmod t$ decryption primitive, where s denotes the secret key, the new decryption circuit for the downsampling of c is:

$$((c_0(z^{\frac{1}{2}}) + c_1^{(even)}(z)s^{(even)}(z) + z c_1^{(odd)}(z)s^{(odd)}(z)) \bmod q) \bmod t,$$

where $c_0(z^{\frac{1}{2}})$ denotes the downsampling by a factor of two, and the upperscript denotes the phase (even or odd) of the polynomials.

Therefore, downsampling reduces the number of coefficients of the involved polynomials, but it also increases the number of polynomials of the ciphertexts. We reduce this expansion on the number of polynomial elements after each downsampling by resorting to a relinearization primitive (see Section II-A and [6]).

Interestingly, if our target were to reduce the cipher expansion of the ciphertexts (compressing the signal instead of denoising it), we could skip the relinearization primitive and leverage the encrypted wavelet transform to just discard the detail coefficients, approximating the signal with the (encrypted) approximation coefficients: we would have $\frac{3n}{2}$ coefficients modulo q instead of the $2n$ coefficients of a fresh ciphertext, hence reducing the expansion by a factor of $\frac{4}{3}$.

C. Homomorphic Threshold

After homomorphically computing the wavelet transform, the denoising scheme involves thresholding the encrypted transformed output. Previous approaches [1] to encrypted thresholding resort to the use of Paillier encryptions [2] and an interactive protocol between the secret key owner and the third party, as there is no efficient method proposed so far to deal with homomorphic thresholding and additions/multiplications at the same time. Conversely, our main objective is to reach an unattended solution without intervention of the secret key owner during the process.

Paillier cryptosystem cannot support additions and multiplications between two encrypted messages at the same time. This drawback is severe for our scenario, as our approach to the homomorphic computation of the threshold requires to homomorphically compute both encrypted additions and multiplications. Therefore, an m-RLWE based cryptosystem [5] also allows us to tackle this challenge, at the cost of additional issues derived from its peculiar polynomial structure, which we address in Section III-C1.

Our approach to a homomorphic thresholding block is the following: let $f(x)$ be a function, and consider that we have a set of different points $\{x_0, \ldots, x_l\}$ and their corresponding outputs $\{f(x_0), \ldots, f(x_l)\}$. Now, let us compute the smallest-degree polynomial $p(x) = \sum_{i=0}^{l} a_i x^i$ which satisfies $p(x_i) = f(x_i)$ for $i = 0, \ldots, l$, that is, we find the interpolating polynomial of $f(x)$ for a given set of $l+1$ different points (we refer the reader to [13] for more details on polynomial interpolation).

The solution for the polynomial coefficients a_i can be expressed in matrix form as:

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^l \\ 1 & x_1 & x_1^2 & \ldots & x_1^l \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_l & x_l^2 & \cdots & x_l^l \end{bmatrix}}_{\boldsymbol{X}} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_l) \end{bmatrix},$$

where all the operations are carried out modulo-t (the plaintext domain); it can be easily seen that considering a prime t, $\boldsymbol{X}$ is a nonsingular Vandermonde matrix, as all the x_i are different and t is prime (this implies that there are no zero divisors in $\mathbb{Z}_t$), so the determinant $\det(\boldsymbol{X})$ is not zero. Therefore, the linear system has a unique solution for the coefficients a_i. This interpolating polynomial is typically computed resorting to its Lagrange form:

$$p(x) = \sum_{i=0}^{l} f(x_i) L_i(x) \bmod t, \tag{2}$$

where $L_j(x) = \prod_{i \neq j} (x - x_i) (x_j - x_i)^{-1} \bmod t$.

We leverage this interpolating polynomial $p(x)$ for a threshold computation as follows: a) we consider a function $f(x)$ which encodes the desired threshold function for $x \in \mathbb{Z}_t$, and b) we obtain the interpolating polynomial $p(x)$ for the required inputs.

The polynomial $p(x)$ describes one arithmetic circuit with several layers of additions and products over the same input x; thus, we can homomorphically compute the threshold if the chosen cryptosystem can perform both the addition and multiplication of two encrypted messages up to the depth of such circuit.

It is important to note that the proposed procedure is not limited to threshold functions; in fact, it can be analogously applied to general functions described by any $f(x)$. Additionally, the particular shape of $f(x)$ or the value of the corresponding threshold do not affect our contribution. Therefore, we assume that either the threshold or $f(x)$ are pre-defined in the clear, and we focus on how to homomorphically apply the threshold function as a circuit in the encrypted values.

1) Element-wise threshold: We resort to the use of the m-RLWE based cryptosystem [5] which, as explained in previous sections, allows us to efficiently perform the wavelet transform and to encrypt multidimensional signals. Its main advantage is enabling encrypted cyclic convolutions with only one ciphertext multiplication.

However, the threshold circuit has to be independently computed for each coefficient, so we need element-wise operations, which are not supported by the homomorphism. Consequently, the advantage of having the signal encoded with a polynomial structure becomes a problem for applying the threshold. To address this problem, we introduce an unattended homomorphic NTT (Number Theoretic Transform) [6] of the encrypted signal. The NTT has a convolution property (similar to that of the Fourier Transform), such that the convolutions in the transformed domain get translated into component-wise products in the original domain. We proceed as follows:

- Compute the homomorphic NTT of the encrypted signal.
- The encryped NTT of the signal is the input to the arithmetic threshold circuit.
- After the threshold circuit, we perform a homomorphic INTT.

As each ciphertext addition performs the addition of two NTTs and the ciphertext multiplication is equivalent to the cyclic convolution between two NTTs, we are homomorphically performing the element-wise multiplication between the values of the encrypted signal. Hence, when we consider the NTT of the encrypted signal as the input of the threshold circuit (see Eq. (2)), we are actually homomorphically computing the threshold for all the signal values.

a) Optimization for square images: In our proposed scheme, we perform a bidimensional NTT of the image. As the NTT is a separable transform, this can be easily realized by concatenating two homomorphic univariate NTTs (horizontal and vertical). For this purpose, a direct application of the methods proposed in [6] is not the optimal procedure, as they would be considering more relinearization matrices than needed. Therefore, we propose an optimization on the additional information required to perform the bivariate NTT for a square image (or in general, the multivariate NTT of any multidimensional signal with the same length in each dimension).

The general algorithm presented in [6] for performing our two NTTs (one for each dimension) would need one relinearization matrix for each NTT. However, when working with square images, it can be seen that one of the matrices can be replaced by a basic relinearization (see Section II-A), hence reducing in half the additional information with respect to the direct application of the original method in [6].

Our optimization reuses the relinearization matrix of one of the NTTs by performing two changes of variables $x \to y$ and $y \to x$. This procedure allows to apply the homomorphic NTT to the second variable, but it also introduces a change on the considered secret key, which now has its variables reversed. This problem can be solved

with a basic relinearization for performing the switching key (see Section II-A) which has a size negligible compared to the original relinearization matrix.

2) Efficient computation of the threshold circuit: This section evaluates the computational cost of the threshold circuit and proposes methods for efficiently computing it in the encrypted domain.

In the worst case scenario, the maximum number of different points that our threshold circuit can have as input is t, which is the modulo considered for the plaintext (see Section II). Therefore, we can find an interpolating polynomial whose maximum possible degree is $t-1$.

It is also known that there exist algorithms for computing general polynomials of degree $t-1$ with as many multiplications as the degree of the polynomial [14], for example, resorting to Horner's rule [15] we can easily compute a polynomial of degree $t-1$ with $t-1$ multiplications. However, dealing with a homomorphic cryptosystem brings about two important points:

- Horner's rule considers that all the multiplications have the same cost; hence, it does not take into account our special case dealing with a homomorphic cryptosystem, where multiplications between a ciphertext and a known scalar value are negligible with respect to the product between two encrypted values.
- Horner's rule does not take into account that a somewhat homomorphic cryptosystem bounds the number of allowed multiplications over the same encryped value "x" (in our case it is bounded by D; see Section II).

Hence, in order to deal with these constraints, we resort to the algorithms for polynomial evaluation proposed by Paterson and Stockmeyer [16], which only count non-scalar multiplications, i.e., those multiplications involving the variable of the polynomial on both sides. Therefore, if we adapt their algorithms for bounding the number of multiplications over the same encrypted value, we can compute an arithmetic circuit of an l-degree polynomial with an order of $\mathcal{O}(\sqrt{l})$ non-scalar multiplications (ciphertext multiplications).

The smallest number of multiplications can be achieved with the algorithm C from [16], which has a computional cost equivalent to $\sqrt{2l} + \log_2 l + \mathcal{O}(1)$ ciphertext multiplications:

- It assumes $l = k2^{m-1}$. If this is not the case, we decompose l in smaller pieces of length $k2^{i-1}$, evaluate them separately and subsequently join them using the powers $\{x^{2k}, \ldots, x^{2k\lceil \log_2 \frac{l}{k} \rceil}\}$. This implies an additional cost of $\log_2 l/k$ multiplications.
- Compute the powers $\{x^2, x^3, \ldots, x^k\}$.
- Compute the powers $\{x^{2k}, x^{4k}, \ldots, x^{2^{m-1}k}\}$.
- After computing these powers, we can evaluate the polynomial with a total of $\sqrt{2l} + \log_2 l + \mathcal{O}(1)$ nonscalar multiplications if we consider $k \approx \sqrt{\frac{l}{2}}$.

IV. Security and Performance Evaluation

This section evaluates both the performance and security of our proposed scheme. First, we briefly revisit and discuss some important concepts regarding the security of lattice-based cryptosystems. Afterwards, we show which are the changes that we can apply to the scheme so as to improve its efficiency when working in practical applications. Finally, we present the achieved runtimes together with the corresponding security parameters.

A. Security of Lattice Cryptosystems

All the proposed methods are noninteractive, and their security is entirely based on the semantic security of the used cryptosystem and the hardness assumptions on which it is grounded (m-RLWE problem). We can analyze the security of lattice-based cryptosystems by following the same procedures of prior works [5], [6], [10]. Hence, we focus on distinguishing attacks [17], which aim at breaking the indistinguishability assumption resorting to basis reduction algorithms.

We do not specifically deal with decoding attacks, which are aimed at obtaining the secret key, but we consider minimum values for $n = n_x n_y$ similar to those used in [10], so we can achieve protection against the decoding attacks described in [18].

1) Distinguishing attacks: The best attacks against lattice-based cryptosystems rely on basis reduction algorithms, being BKZ [19] one of the most efficient ones. The parameter which establishes the complexity of reduction atacks on the lattice is the root Hermite factor $\delta > 1$, such that for a constant k the runtime of an attack is approximately proportional to $e^{k/\log \delta}$ (see [6] for more details on how δ is obtained). In order to calculate the corresponding bit security (and be able to compare our chosen cryptosystem with other "traditional" cryptosystems), we resort to the accepted pessimistic lower bound estimate $t_{BKZ}(\delta)$ of [18]:

$$t_{BKZ}(\delta) = \log_2 \left(T_{BKZ}(\delta)\right) = \frac{1.8}{\log_2(\delta)} - 110. \tag{3}$$

B. Performance Evaluation

This section discusses some additional implementation challenges that can appear when realizing our proposed scheme in a practical scenario. We also bring about some approaches which can help to considerably improve the efficiency and cipher expansion of the proposed solutions for these practical situations. Additionally, we also include different runtimes together with the corresponding bit security (Eq. (3)) for several image sizes when performing our image denoising in the encrypted domain.

1) Practical considerations: Carefully looking at all the stages of our proposed encrypted image denoising process, it can be seen that the most costly operation is the element-wise threshold circuit, whose worst-case degree is highly dependent on the input cardinality.

For practical input images, their pixel values vary in range, therefore determining the degree of the threshold circuit, together with the corresponding computational cost for its execution.

In order to alleviate the computational cost of the threshold circuit, we can reduce the maximum value that the image coefficients can achieve as a result of the homomorphic wavelet transform.

Hence, for a practical implementation of our encrypted image denoising, we resort to the use of the Haar wavelet. Its use allows to easily analyze how the encrypted image coefficients increase after each stage, yielding a factor of 4^k after k stages. So, for a practical range for images like $[0, 255]$, by mapping $[0, 255] \rightarrow [-127, 128]$ before encryption, we have that the output of the k-stage belongs to the possible interval $4^k[-127, 128] = [-2^{7+2k} + 2^{2k}, 2^{7+2k}]$ for the coefficients. Now, we can take the number of values of this interval minus one as the considered maximum degree for the threshold circuit (in practical cases this degree would be much lower), therefore obtaining a clear improvement comparing with the case of using $t-1$ as the maximum degree. Additionally, the structure of the Haar wavelet allows us to express the computational cost of the wavelet transform as very efficient additions among shifted polynomials.

2) Implementation and execution times: We have implemented the 2-RLWE cryptosystem in C++ using the GMP,[1] MPFR[2] and NFLlib [20] libraries. Table II compares the performance for encrypted image denoising for a range of four different sizes of images and for two

[1] www.gmplib.org
[2] www.mpfr.org

different lower bounds on the bit security (above 128 and above 256 bits of security), when running on an Intel Core-i5 2500 at 3.3 GHz using only one core (but the code is very amenable to parallelization). For all the cases, we consider a Haar wavelet and two stages for the filter-bank implementation. Additionally, the range of values for the pixels is $[0, 255]$, mapped to $[-127, 128]$ before pre-processing the input images. The possible interval for the values of the (clear-text) coefficients at the input of the threshold circuit is $[-2032, 2048]$; hence, for preserving correctness in decryption, we consider $D = \lceil \log_2 4081 \rceil = 12$ for obtaining the bound on q (see Eq. (1)). This value for D yields a conservative pessimistic q, as the optimizations of [6] allow to consider ciphertext multiplications with polynomials of less than n coefficients. In any case, we take into account this fact for the estimation of δ and the calculation of the equivalent bit security. We report here the achieved performance when denoising is used as a standalone block, but it is possible to perform further homomorphic operations supported by the cryptosystem before or after the denoising, being the only requirement to increase D to account for the rest of the processes in the chain.

We include the corresponding runtimes for each of the operations in the pipeline: the pre-/post-processing together with encryption/decryption, and the homomorphic image denoising. Additionally, we have included the root Hermite factor δ, the bit security (see eq. (3)) for each scenario and the ratio in bits between the size of the encrypted image and the size of the image in clear (cipher expansion).

TABLE II
PERFORMANCE OF IMAGE DENOISING ($D = 12$, $A = 1$, $t = 65537$, $\sigma = 1$, 2 STAGES)

2-RLWE cryptosystem (bit security > 128)				
Image size	128×128	256×256	512×512	1024×1024
Cipher Exp. (ratio)	101.25	107.5	113.75	120
δ	1.0043	1.0045	1.0048	1.0051
Bit security (Eq.(3))	≈ 182	≈ 165	≈ 150	≈ 136
Encrypt. + Pre-proc. (*m.s*)	9	41	199	939
Decrypt. + Post-proc. (*m.s*)	10	42	211	1428
Enc. Denoising (*min*)	1.46	6.06	25.74	106.77
2-RLWE cryptosystem (bit security > 256)				
Image size	128×128	256×256	512×512	1024×1024
Cipher Exp. (ratio)	104.25	110.5	116.75	123
δ	1.0022	1.0023	1.0025	1.0026
Bit security (Eq.(3))	≈ 456	≈ 424	≈ 396	≈ 370
Encrypt. + Pre-proc. (*m.s*)	19	97	417	1973
Decrypt. + Post-proc. (*m.s*)	20	101	441	2998
Enc. Denoising (*min*)	4.26	17.85	76.49	316.69

The performance of the proposed methods shown in Table II proves the practicality of the scheme, requiring a few minutes (using just one core) to process an entire image of moderate size with a bit-security over 128 bits (mid-term security), and few milliseconds for encryption/decryption. The denoising runtime shows a quasi-linear behavior in terms of the image size, which is basically caused by the computational cost of the polynomial operations. This is much more efficient than using a comparison protocol with Paillier; e.g., [5] shows that for a basic filtering operation of a 1024×1024 image size, an RLWE-based solution provides runtimes 3 orders of magnitude faster than Paillier. Even a fully interactive secret sharing solution like [8] (which claims to be more efficient than a garbled-circuit based solution) needs over 16 minutes for a two-level denoising of a 128×128 image; considering a very favourable case with a communication cost of a LAN. For this case, our solution, besides not requiring any interaction, performs one order of magnitude faster.

V. CONCLUSIONS

We have proposed non-interactive methods based on 2-RLWE (Ring Learning with Errors) that overcome the limitations of previous Signal Processing in the Encrypted Domain solutions to efficiently perform encrypted image denoising. We have shown how to combine homomorphic polynomial operations and thresholding without involving decryption or interaction, therefore enabling fully unattended encrypted image denoising.

The performance of our proposed methods for mid-term and long-term security (128 and more than 256 bits) proves their practicality, improving on the usage of interactive comparison protocols with Paillier, and also comparing favorably with respect to fully interactive secret sharing solutions, even when we do not require any interaction.

ACKNOWLEDGMENTS

This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (ERDF) under projects COMPASS (TEC2013-47020-C2-1-R) and COMONSENS (TEC2015-69648-REDC), by the Galician Regional Government and ERDF under projects "Consolidation of Research Units" (GRC2013/009) and AtlantTIC, and by the EU H2020 Framework Programme under project WITDOM (project no. 644371).

REFERENCES

[1] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted Signal Processing for Privacy Protection," *IEEE SP Mag.*, vol. 30, no. 1, pp. 82–105, 2013.
[2] P. Paillier, "Public-key Cryptosystems Based on Composite Degree Residuosity Classes," in *EUROCRYPT'99*. Springer, 1999, pp. 223–238.
[3] T. Bianchi, A. Piva, and M. Barni, "Composite Signal Representation for Fast and Storage-Efficient Processing of Encrypted Signals," *IEEE Trans. on Inf. Forensics & Sec.*, vol. 5, no. 1, pp. 180–187, March 2010.
[4] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, "On Ideal Lattices over the Tensor Product of Number Fields and Ring Learning with Errors over Multivariate Rings," *ArXiv e-prints*, Jul. 2016.
[5] ——, "Multivariate Lattices for Encrypted Image Processing," in *IEEE ICASSP'15*, April 2015, pp. 1707–1711.
[6] ——, "Number Theoretic Transforms for Secure Signal Processing," *ArXiv e-prints*, Jul. 2016.
[7] X. Hu, W. Zhang, K. Li, H. Hu, and N. Yu, "Secure Nonlocal Denoising in Outsourced Images," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 3, pp. 40:1–40:23, Mar. 2016.
[8] S. M. SaghaianNejadEsfahani, Y. Luo, and S. c. S. Cheung, "Privacy protected image denoising with secret shares," in *IEEE ICIP'12*, Sept 2012, pp. 253–256.
[9] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," Crypt. ePrint Archive, Report 2012/144, 2012.
[10] K. Lauter, M. Naehrig, and V. Vaikuntanathan, "Can Homomorphic Encryption be Practical?" Crypt. ePrint Archive, Report 2011/405, 2011.
[11] A. Buades, B. Coll, and J. M. Morel, "A Review of Image Denoising Algorithms, with a New One," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
[12] I. Atkinson, F. Kamalabadi, S. Mohan, and D. L. Jones, "Asymptotically Optimal Blind Estimation of Multichannel Images," *IEEE Trans. on Image Proc.*, vol. 15, no. 4, pp. 992–1007, April 2006.
[13] G. Phillips, *Interpolation and Approximation by Polynomials*, ser. CMS books in mathematics. New York: Springer, 2003.
[14] S. Winograd, "On the Number of Multiplications Required to Compute Certain Functions," *PNAS*, vol. 58, no. 5, pp. 1840–1842, 1967.
[15] P. Borwein and T. Erdélyi, *Polynomials and Polynomial Inequalities*, ser. Graduate texts in mathematics. New York: Springer, 1995.
[16] M. S. Paterson and L. J. Stockmeyer, "On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials," *SIAM J. Comput.*, vol. 2, no. 1, pp. 60–66, 1973.
[17] D. Micciancio and O. Regev, "Lattice-based Cryptography," in *Post-Quantum Cryptography*. Springer, 2009, pp. 147–191.
[18] R. Lindner and C. Peikert, "Better Key Sizes (and Attacks) for LWE-based Encryption," in *CT-RSA'11*. Springer, 2011, pp. 319–339.
[19] Y. Chen and P. Q. Nguyen, "BKZ 2.0: Better Lattice Security Estimates," in *ASIACRYPT'11*, ser. LNCS. Springer, 2011, vol. 7073, pp. 1–20.
[20] C. Aguilar-Melchor, J. Barrier, S. Guelton, A. Guinet, M.-O. Killijian, and T. Lepoint, "NFLlib: NTT-based Fast Lattice Library," in *CT-RSA'16*, San Francisco, United States, Feb. 2016.