Camera Attribution Forensic Analyzer in the Encrypted Domain*

Alberto Pedrouzo-Ulloa[1] Miguel Masciopinto[1] Juan Ramón Troncoso-Pastoriza[2] Fernando Pérez-González[1]
[1] University of Vigo, Vigo, Spain, {apedrouzo,mmasciopinto,fperez}@gts.uvigo.es
[2] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, juan.troncoso-pastoriza@epfl.ch

Abstract

Digital media forensics must deal with constantly increasing volumes of data. In order to efficiently scale up, outsourcing computation becomes an appealing solution. However, due to the highly sensitive nature of forensic data, its privacy must be protected when processed in an untrusted environment. This work proposes a new framework to efficiently perform an outsourced PRNU (Photoresponse Non-Uniformity) fingerprint extraction and detection on encrypted images in a fully unattended way. For this purpose, we rely on lattice-based homomorphic cryptosystems paired with advanced optimization strategies. We evaluate our solutions in terms of efficiency, security and performance for real image datasets, showing the feasibility of camera attribution in the encrypted domain.

1. Introduction

Digital media forensics is rapidly evolving as an answer to societal demands. Besides lively research topic, several commercial applications already exist that are able to (semi-) automatically detect forgeries and tampering, or identify and/or cluster acquisition devices. Although most of these tools have relatively low computational complexity, they must be run on very large and ever increasing databases, with efficiency thus becoming a major concern. On the other hand, the still growing popularity of content-sharing websites such as YouTube, Instagram or Facebook, and the Dark Web [31], leads to rapidly obsolescent forensic analysis platforms, especially in times of budget shortfalls, and quite conspicuously so in the case of law enforcement. An increasingly appealing solution is to buy computing power and database storage as needed, by running software and keeping data on outsourced platforms such as Amazon Web Services, Microsoft Azure or Google Cloud. This approach cuts down maintenance costs and dynamically scales with computing needs. However, outsourcing faces the problem of guaranteeing confidentiality and privacy at the server end, much more so considering that forensic data is highly sensitive.

One salient instance of extremely sensitive data is related to child pornography. Some of the existing tools for camera attribution or device clustering [17, 16, 25, 14] find an immediate application in fighting against crimes involving depictions of minors [1, 37, 26]. To get an estimate of the sheer size of this problem, researchers looked during a one-year period (2010-2011) at two of the then most common peer-to-peer networks, to find more than 2,500,000 peers worldwide sharing child pornography [20]. Obviously, processing this type of files outside of law enforcement's own infrastructure is currently out of the question; encryption alone is not a solution either, because contents must be opened at the server end in order to analyze them.

Opportunely, recent advances in the field of Secure Signal Processing (SSP) [22] hint at a potential solution to cloudify forensic analysis software and forensic data storage in such a privacy-conscious way with zero information leakage. This means that the server does not even learn the outcome of a binary forensic test. Recently, some works have introduced new solutions based on lattice cryptography which are especially adapted to efficiently work with images, covering encrypted operations that range from image filtering [32] and image denoising [33] to more general image processing operations [35].

Most camera-attribution methods rely on the so-called Photoresponse Non-Uniformity (PRNU). The PRNU is a specific noise pattern inherent to digital imaging sensors which represents the difference in response of the sensor array to a uniform light source [19]. It is caused by random imperfections in the manufacturing process and it can be used as a fingerprint of the camera device, serving to determine whether a given test image was taken by a certain camera, by matching a residual obtained from the test image with the fingerprint. Due to its great potential for image forensics, many works have studied the use and properties of the PRNU, from the peculiarities of its mathematical modeling [7, 8] to a wide range of possible applications, including source attribution [4], source-based clustering [25], and tampering detection [21].

This work proposes a new framework for the secure outsourcing of PRNU-based source attribution (including secure PRNU extraction, detection and storage) in a fully unattended way, that is, without the intervention of the secret key owner during the process. To this end, we improve on the efficiency of the state-of-the-art in secure, unattended solutions for image denoising, and we show how filtering, polynomial, denoising and pixel-wise operations

*GPSC is funded by the Agencia Estatal de Investigación (Spain) and the European Regional Development Fund (ERDF) under projects WINTER (TEC2016-76409-C2-2-R) and COMONSENS (TEC2015-69648-REDC). Also funded by the Xunta de Galicia and the European Union (European Regional Development Fund - ERDF) under projects Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019, Grupo de Referencia ED431C2017/53 and Red Temática RedTEIC 2017-2018, and also by the FPI grant (BES-2014-069018). We would like to thank Jean-Claude Bajard, Julien Eynard, M. Anwar Hasan and Vincent Zucca for providing us with their RNS implementation of the FV cryptosystem.

(e.g. element-wise division) can be homomorphically performed in a single round without the need of an interactive protocol.

Main Contributions To the best of our knowledge,[1] this is the first work in the literature that proposes a secure implementation of a forensic analyzer. The framework is here epitomized by a PRNU-based extractor/detector, but it embraces many other existing forensic tools. Other main contributions of our work are:

- Rooting in the secure wavelet-based denoising primitive presented in [33], we improve the results therein by means of a new threshold function. Our new procedure enables a considerable reduction in both the depth of the evaluated circuit and the number of effective ciphertext multiplications.
- We discuss the application of our novel homomorphic wavelet-based denoising primitive within a complex use case: PRNU extraction/detection for camera attribution.
- As such application requires many calls to the homomorphic wavelet denoising primitive, we show how to optimize its implementation. The resulting method is able to evaluate the full extraction/detection processes while avoiding execution-time interactions between client and server.

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., $R_t[z] = \mathbb{Z}_t[z]/(z^{n_z}+1)$ denotes the polynomial ring in z modulo $z^{n_z}+1$ with coefficients in $\mathbb{Z}_t$, while $R_t[x,y] = (R_t[x])[y]/(y^{n_y}+1)$ is the bivariate polynomial ring with coefficients in $\mathbb{Z}_t$ reduced modulo $x^{n_x}+1$ and $y^{n_y}+1$ (n_z, n_x and n_y are powers of two). We also represent univariate (bivariate) polynomials as column vectors (resp. matrices) of their coefficients. Finally $\boldsymbol{A}\circ\boldsymbol{B}$ (resp. $\boldsymbol{A}\cdot\boldsymbol{B}$) is the Hadamard (resp. inner) product between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

The rest of the paper is organized as follows: Section 2 briefly revises the used lattice-based cryptosystems and the PRNU matching scenario. Section 3 introduces the main scheme for secure PRNU extraction and detection, and Section 4 evaluates it in terms of security, efficiency and performance.

2. Preliminaries

This section summarizes the main operations performed in a PRNU-based extractor/detector and revisits the lattice-based cryptosystem used in our proposed scheme.

2.1. Basic structure of PRNU extraction/detection

The sensor output model can be approximated by the first two terms of its Taylor series [8], as $\boldsymbol{Y} = (\boldsymbol{1}+\boldsymbol{K})\circ\boldsymbol{X}+\boldsymbol{N}$, where $\boldsymbol{Y}$ is the output matrix of the imaging sensor, $\boldsymbol{K}$ is the PRNU signal, $\boldsymbol{1}$ is a matrix filled with ones, $\boldsymbol{X}$ is the incident light intensity and $\boldsymbol{N}$ represents other noise sources.

[1] Thanks to the anonymous reviewers, we were made aware of a related work by Mohanty et al. [29]. It requires the use of a trusted environment (ARM TrustZone), while our approach can be fully implemented on a general purpose architecture.

It is worth noting that $\boldsymbol{X}$ is unknown in practice, but an estimate $\hat{\boldsymbol{X}}$ can be obtained with a denoising operation over $\boldsymbol{Y}$.

PRNU fingerprint extraction: Let $\{\boldsymbol{Y}^{(l)}\}_{l=1}^{M}$ be a set of M images taken with the same camera device of N_k pixels at native resolution. The PRNU can be estimated by using the maximum likelihood estimator (MLE) derived in [37]:

$$\hat{\boldsymbol{K}} = \left(\sum_{l=1}^{M}\boldsymbol{W}^{(l)}\circ\hat{\boldsymbol{X}}^{(l)}\right)\circ\left(\sum_{l=1}^{M}(\hat{\boldsymbol{X}}^{(l)})^{\circ 2}\right)^{\circ -1}, \quad (1)$$

where $\boldsymbol{W}^{(l)} = \boldsymbol{Y}^{(l)} - \hat{\boldsymbol{X}}^{(l)}$ is the denoising residue of the image $\boldsymbol{Y}^{(l)}$, and $\boldsymbol{A}^{\circ -1}$ (resp. $\boldsymbol{A}^{\circ 2}$) stands for the Hadamard inverse (resp. square) of matrix $\boldsymbol{A}$.

PRNU detection: Given a test image $\boldsymbol{Y}_t$ with residue $\boldsymbol{W}_t = \boldsymbol{Y}_t - \hat{\boldsymbol{X}}_t$ and a PRNU estimate $\hat{\boldsymbol{K}}$, the following hypothesis testing problem can be formulated:

H_0: $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ correspond to different PRNUs.
H_1: $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ correspond to the same PRNU.

As a computationally simpler alternative to the use of the Peak to Correlation Energy (PCE) statistic [17], here we consider

$$u = \boldsymbol{W}_t \cdot \hat{\boldsymbol{K}}, \quad (2)$$

for which an estimate of the variance is

$$\sigma_u^2 = \frac{1}{N_k}(\hat{\boldsymbol{K}}\cdot\hat{\boldsymbol{K}})(\boldsymbol{W}_t\cdot\boldsymbol{W}_t); \quad (3)$$

then, for a given probability of false alarm, the test becomes [37]

$$\frac{u}{\sigma_u} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda, \quad (4)$$

where λ is a fixed threshold that changes depending on the desired false positive probability. In (2) we assume that the signals $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ are aligned; otherwise, the maximum of the cross-correlation for every possible lag must be chosen as u in (4) [16].

2.2. A 2-RLWE based Cryptosystem

We use univariate and bivariate versions of the FV cryptosystem [13] as the underlying block for our secure forensic analyzer. Due to space constraints, we do not include here a description of all the cryptosystem primitives (we refer the reader to [13] for a detailed description). The plaintext elements belong to the ring $R_t[x,y]$ and ciphertexts are composed of two elements belonging to $R_q[x,y]$. When we work with bivariate polynomials instead of the usual univariate ones, security relies on the indistinguishability assumption of the 2-RLWE problem, defined as follows:

Definition 1 (**2-RLWE problem [34, 32, 10]**) *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 $\chi[x,y] \in R_q[x,y]$ 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.*

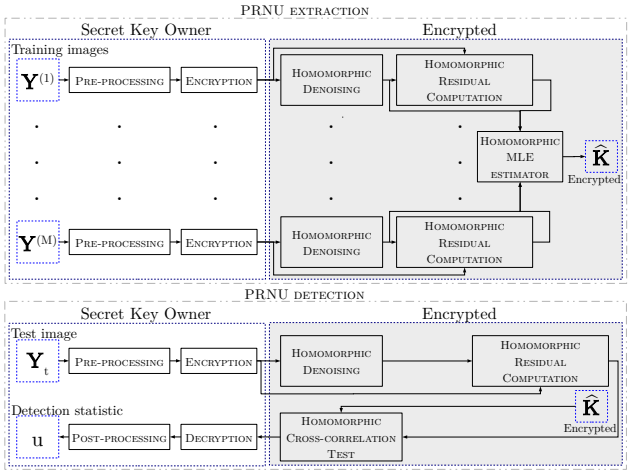


Figure 1: Secure scheme for the PRNU extractor/detector.

The bivariate cryptosystem can encrypt images in only one ciphertext, instead of encrypting each pixel in a different ciphertext. It also enables efficient pixel-wise additions with one ciphertext addition and bivariate linear/cyclic convolutions with only one ciphertext multiplication at the cost of a small overhead (operations are performed over $\mathbb{Z}_q$ instead of $\mathbb{Z}_t$ with $q > t$). We refer the reader to [13, 32, 33] for further details on these homomorphic operations.

To evaluate an arithmetic circuit of multiplicative depth L, we can consider the following condition to have correct decryption (Theorem 1 in [13]) $4n^L(n+1.25)^{L+1}t^{L-1} < \lfloor \frac{q}{B} \rfloor$, where $n = n_x n_y$ and $||\chi|| < B$, that is, χ is a B-bounded distribution.

3. Proposed Scheme

This section describes the proposed scheme for securely evaluating the PRNU extractor/detector. First, we give a general overview of its structure with a brief description of each block. Afterwards, we focus on the secure image denoising block due to its importance for the PRNU extractor/detector. Finally, the two main tasks (PRNU extraction/detection) which form part of the scheme are discussed in more depth.

3.1. General Overview

We establish the following two working hypotheses for the proposed secure solution:

- The adversary model is based on a semi-honest setting, where the party who evaluates the encrypted PRNU extractor/detector tries to gather as much information of the content of the input images as possible, but does not deviate from the protocol.
- We require an unattended solution where the secret key owner does not have to participate in the middle of the process.

Taking into account these constraints, Figure 1 sketches the proposed scheme, which involves the two main attribution stages: 1) The extraction of the PRNU fingerprint given a training set of images from the same camera, and 2) the detection of the PRNU in an input image taking the previously extracted PRNU fingerprint as a template to be matched (see Section 2.1).

Our solution uses the RLWE and 2-RLWE versions [32] of the FV cryptosystem [13] as a means to perform encrypted arithmetic

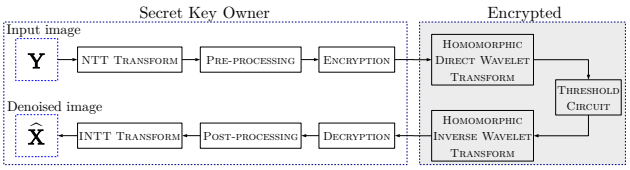


Figure 2: Encrypted Wavelet-based Denosing.

operations. We also make use of some of the techniques described in [33], such as (a) a lightweight pre-/post-processing (for homomorphic cyclic convolutions when multiplying two ciphertexts) and (b) the use of homomorphic NTT/INTTs (Direct/Inverse Number Theoretic Transforms) from [36] (for element-wise additions and multiplications between encrypted vectors).

Whereas the two main stages securely implement two different processes (represented by, respectively, (1) and (2)), both make use of an encrypted image denoising block. In fact, due to the high number of denoising operations, optimizing this common block is especially important for the efficiency of the whole pipeline.

In the following sections we explain in more detail the two main stages in Figure 1, including our optimizations over the state-of-the-art encrypted denoising block proposed in [33].

3.2. Encrypted Image Denoising

We consider as baseline the method for image denoising introduced in [33],[2] which comprises three elements: 1) homomorphic direct/inverse wavelet transform, 2) homomorphic NTT/INTT, and 3) threshold circuit. We considerably improve on the performance of this method by modifying the second and third elements.

Firstly, our solution moves the homomorphic NTT/INTT to the pre-/post-processing stage, avoiding its costly homomorphic computation and performing most of the operations in this batched setting. Figure 2 details the new structure of this primitive after substituting the homomorphic NTT/INTT block.

Regarding the last element, instead of directly applying a threshold function, we consider a quantization function which, in practice, works similarly to the hard threshold function from [33]. The advantage of this quantization is that it can be implemented by means of the "lowest digit removal" polynomials defined in [18, 6]. Their use allows for a smaller depth on the threshold circuit, hence considerably reducing the runtime of the primitive.

3.2.1 Homomorphic Wavelet Transform

We consider a filter-bank implementation for computing both the homomorphic direct and inverse wavelet transforms of the denoising algorithm. In [33] the authors introduce a light pre-/post-processing which enables the efficient application of low-/high-pass wavelets with cyclic convolutions by means of only one multiplication between a ciphertext and a plaintext encoding the corresponding wavelets.

After each homomorphic filtering operation, a downsampling or upsampling by a factor of 2 has to be applied depending on whether we work with the direct or the inverse transform. This

[2] This choice is mainly motivated by the widespread use of Wavelet denoising and its good tradeoff between cost and performance.

downsampling/upsampling operation is very efficient, but it has to be followed by a costly relinearization.

In this work, we avoid these downsampling/upsampling steps (together with the relinearization) by previously dividing and separately encrypting the original image into as many polyphase components as required in the last level of the homomorphic wavelet transform (see [36]). Restricting the wavelet transform to Haar wavelets, their particular structure enables to express the transform as very efficient additions among the polyphase components.[3]

3.2.2 Homomorphic Threshold

The approach considered in [33] for the homomorphic threshold (see Figure 2) directly interpolates the desired function (together with a normalization factor corresponding to the wavelet transform) over the plaintext. However, as the plaintext cardinality increases after each stage of the filter bank,[4] the complexity of the threshold circuit also increases. Hence, the results from [33] do not scale well when working with a high number of stages (in [33] the authors evaluate a denoising algorithm with only 2 stages).

This section introduces our quantization method to homomorphically evaluate both the normalization and the threshold. By choosing the plaintext modulo t as a prime power p^2 (where p is roughly equal to the number of possible input values for the images, e.g., $p=257$), we can evaluate the quantization step with a polynomial whose maximum degree is equal to the cardinality of the plaintext before applying the wavelet transform, which considerably improves its performance with respect to [33].

This technique is based on the use of the "lowest digit removal" polynomials recently introduced in [18, 6] as a means to enhance the performance of bootstrapping for FHE (Fully Homomorphic Encryption) schemes. Here we leverage their properties for a different purpose: the homomorphic quantization of the plaintext.

We first present these polynomials (Lemma 3 from [6]) and how to construct them for our particular scenario:

Lemma 1 ([6]) *Let p be a prime and $e \geq 1$. Then there exists a polynomial f of degree at most $(e-1)(p-1)+1$ such that for every integer $0 \leq x < p^e$,*

$$f(x) \equiv (x - (x \bmod p)) \bmod p^e, \tag{5}$$

where $|x \bmod p| \leq \frac{p-1}{2}$ when p is odd.

For $e=2$, $f(x) = -x(x-1)...(x-p+1)$ (Example 4 in [6]).

In our case, the quantization function which we want to evaluate is $\lfloor \frac{x}{Q} \rfloor$ for positive x and $\lceil \frac{x}{Q} \rceil$ for negative x. To have this functionality, and considering $e=2$, we can define $f(x) = -(x + \frac{p-1}{2})...(x+1)x(x-1)...(x-\frac{p-1}{2})$, which implements the desired function for a quantization step $Q=p$. Once we have $f(x) \bmod p^e$ we can directly divide by p to have $\frac{f(x)}{p} \bmod p^{e-1}$. When working with the FV cryptosystem (see Section 2.2), after homomorphically evaluating $f(x)$, this division can be done for free, only introducing a slight increase in the ciphertext's noise (see [6]).

[3] A total of $i4^i$ ciphertext additions for i levels, where each ciphertext encrypts a polynomial of size $\frac{n}{4^i}$ and n is the size of the original image.

[4] For example, considering a Haar wavelet the range of plaintext values is increased by a factor of 4 after each stage.

3.3. Homomorphic Cross-correlation Test

To securely perform the detection test, we have to homomorphically evaluate (2) (the general flow is depicted in Figure 1). After the encrypted denoising block (see Section 3.2), computing the residuals is straightforward by means of a homomorphic subtraction. Afterwards, as the test image may have been cropped, depending on whether it is aligned or not with the PRNU estimate (see Section 2.1), an encrypted scalar product or an encrypted cross-correlation operation is required.

Aligned case: To calculate the scalar product, we take advantage of the fact that the first coefficient of the NTT is the addition of all the coefficients in the time domain. Therefore, the server divides the encrypted PRNU into blocks and obtains the homomorphic NTT transform of each block, multiplies each PRNU block with the corresponding encrypted polyphase component of the residual, and finally adds all the encrypted polyphase components. This method encodes the scalar product in the first coefficient of the encrypted result. [5]

Non-aligned case: Here we want to calculate the full cross-correlation between the encrypted residuals and the reference PRNU. To do this, the client applies a pre-/post-processing over the plaintexts before/after encryption/decryption, and works with a cryptosystem based on the 2-RLWE problem. Then, the server can exploit the cyclic convolution property of the bivariate homomorphic INTT from [33] with the purpose of obtaining the time domain representation of the encrypted polyphase components (we refer the reader to [33] for details on this operation).

Once this is done, as the test image is encrypted in different blocks with a cyclic convolution property, the server can resort to the traditional "overlap-save" method [30] for calculating the linear convolution between the PRNU template and the encrypted polyphase components of the test image.

It must be noted that overlap-save discards part of the computed values, so the server has to generate enough space inside the ciphertexts. To achieve it, the server breaks the content of each encrypted polyphase component into four new ciphertexts before applying the homomorphic INTT, where each one has a quarter of the original polyphase component (for simplicity we consider that we are working with square images) and the rest are zero values. This increases the number of ciphertexts by a factor of 4, yielding a total of 4^{i+1} when working with an i-level wavelet denoising. The computational cost of the mentioned operation is equivalent to applying 4^{i+1} times an overlap-save algorithm over a filter encoded in the ciphertext and a PRNU 4^i times smaller.

Variance normalization: The statistic presented in (2) is normalized by $\sigma_w \sigma_k \sqrt{N_t}$ where $N_t \sigma_w^2 = \boldsymbol{W}_t \cdot \boldsymbol{W}_t$ and $N_k \sigma_k^2 = \hat{\boldsymbol{K}} \cdot \hat{\boldsymbol{K}}$ (N_t and N_k are the number of elements in $\boldsymbol{W}_t$ and $\hat{\boldsymbol{K}}$ respectively). For efficiency reasons, the server calculates the desired λ and returns the encrypted result of the scalar product

[5] For this scalar product we do not take advantage of the bivariate structure of the image, so we could consider an RLWE based-cryptosystem.

or cross-correlation together with an encryption of λ scaled by this normalizing factor. The server could also homomorphically evaluate the division as we describe next for the PRNU extraction.

To compute these normalizing factors, the server can homomorphically evaluate the square of the residuals and PRNU, and add for both the polyphase components of their results. The desired values are stored in the first coefficient of the NTTs (see [38]).

3.4. Secure PRNU extraction

The secure PRNU extraction involves the computation of (1) in an encrypted way. The encrypted denoising block for the input images and the pixel-wise operations on the encrypted image and residuals are analogous to those in (1), which are explained in Section 3.2, so we do not repeat them again here.

Finally, several strategies can be considered to implement the encrypted division needed to fully realize (1) under encryption; we briefly describe them here.

Approximate division: We can consider the methods for encrypted division used in [5, 11, 10], with which we can approximate the result of the division with a predefined bit precision.

For example, the server can approximate the inverse of a number b with 2^r bits of precision with the expression:

$$\rho^{-2^r}\prod_{j=0}^{r-1}\left(\rho^{2^j}+(\rho-b)^{2^j}\right) \text{ where } \frac{\rho}{2}\leq|b|\leq\frac{3\rho}{2}. \qquad (6)$$

This approximation can be applied by adding an adequate value to the denoised images in (1) (for both numerator and denominator), such that all the pixels lie in the right range for convergence (for example, if $p=257$ and pixel values are in the interval $[-128,128]$, the server could add 256).

The server can also use a gradient descent algorithm (previously shifting the negative values to the positive side) as the Newton's root finding algorithm proposed in [5], where the inverse of a number b can be calculated through an iteration of the form

$$a_{i+1}=a_i(2\rho^{2^i}-ba_i), \qquad (7)$$

with $b\in[0,2^k]$ scaled by $\rho^{2^{\mu-1}}$ (that is, $(\rho^{2^{\mu-1}}/b)$), and being μ the number of iterations, $a_0=1$ and $\rho=2^{k-1}$.

4. Security and Performance Evaluation

This section provides a complete evaluation of the proposed scheme, in terms of security, efficiency and performance.

For such evaluation, and due to space constraints, we assume that the client has control over the content of the images. This scenario could arise when the police have seized the camera of a suspect and wants to verify whether a certain image was taken from that camera, but due to legal constraints it cannot be directly outsourced in the clear. In this setting, we can safely assume that the client can gather a set of non-sensitive training images from the same camera (e.g., flatfield images); we can then perform the extraction in the clear. Once the PRNU has been extracted, we do consider that the test images may have a very high sensitive content, which requires the client to encrypt them before outsourcing.

In an extended version of this paper we will also include a complete evaluation of the extraction in the encrypted domain.

Alternatives for PRNU extraction: As can be seen in (1), the extraction is more costly than detection due to its high number of denoising operations. However, we can consider other approaches more amenable to the allowed encrypted operations. For example, instead of separately denoising each image from the training set, we could previously add them and apply (1) to the resulting image. This computation is very similar to the PRNU detection, and the homomorphic addition of all the images can be done very efficiently with the used cryptosystem (see Section 2.2).

4.1. Security of the Proposed Scheme

The security of the proposed scheme is based on the semantic security of the used cryptosystem, which relies on the indistinguishability of the RLWE and 2-RLWE distributions (see Definition 1). In this work, we consider distinguishing attacks [27] (although the considered values of n also resist the decoding attacks described in [24]), aimed at breaking the indistinguishability assumption through basis reduction algorithms (such as BKZ [9]). The runtime of basis reduction attacks is parameterized by the root-Hermite factor $\delta>1$ (for details on how to calculate δ see [23, 36]) as approximately $e^{k/\log\delta}$ with a constant k; hence, a lower δ gives higher attack runtimes. To estimate the bit security, we use the lower bound estimate[6] for BKZ $t_{BKZ}(\delta)$ given in [24]:

$$t_{BKZ}(\delta)=\frac{1.8}{\log_2\delta}-110. \qquad (8)$$

Bit security estimates (together with execution times) for our proposed scheme are included in Tables 1 and 2.

4.2. Implementation and execution times

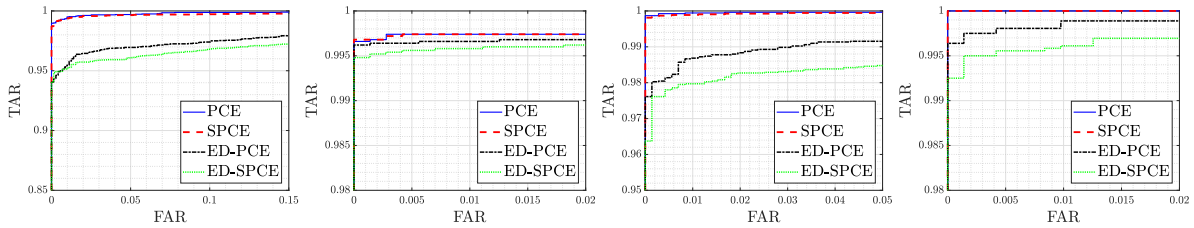
We have implemented our scheme making use of the RNS variant of the FV cryptosystem [3].[7] Table 1 compares the runtimes of our proposed encrypted denoising (with the new threshold circuit) and the original algorithm from [33], which we already optimized by applying the NTT/INTT before/after the pre-/post-processing, to fairly compare the raw performance of the denoising primitive.

The runtimes substantially improve those from [33] (the improvement would be even more significant if we did not include our optimized NTT/INTT in our implementation of [33]). First, we avoid the heavy homomorphic INTT/NTT computation. Secondly, the use of a new threshold function considerably reduces the ciphertext size and the depth of the evaluated circuit, resulting in a much faster computation.

Table 2 reports the runtimes for the detection scenario of our proposed scheme for PRNU extraction/detection. For efficiency reasons we separately compute the detection statistic u and the normalizing factor in two different ciphertexts (avoiding the costly encrypted division, which can be computed by the client as post-processing). The additional process of division does not add an important overhead to the secret key owner (see Table 2). Moreover, due to the highly parallelizable structure of the operations (they can be seamlessly parallelized even with a factor of 256), we include the runtimes considering different levels of parallelization.

[6] This estimate is more pessimistic than the security estimator recently developed by Albrecht et al. [2].

[7] Execution times were measured on an Intel Xeon E5-2667v3 at 3.2 GHz using one core for the non-parallelized option.



(a) Nikon D7000. (b) Nikon D90. (c) Canon 1100D. (d) Nikon D3000.

Figure 3: True Acceptance Rate (TAR) vs. False Alarm Rate (FAR) for 4 different target camera devices. PCE represents the result obtained with the denoising in [28] and the PCE statistic [17], SPCE is the simplified detector in (4) applying the denoising in [28], ED-PCE is the PCE statistic using the encrypted image denoising described in Section 3.2, and ED-SPCE stands for the simplified detector discussed in Section 3.

Table 1: Performance of Encrypted Image Denoising ($\sigma = 8$)

Encrypted Denoising with RLWE cryptosystem (bit security > 110)				
Denoising with 2 stages	Optimized from [33]		Our denoising	
N (size image $N \times N$)	1024	2048	1024	2048
t	65537	65537	257^2	257^2
Cipher Exp. (ratio)	200.6250	210.0000	134.6250	134.6250
δ	1.00561	1.00294	1.00374	1.00374
Bit security (Eq.(8))	≈ 112	≈ 315	≈ 223	≈ 223
L (multiplicative depth)	12	12	8	8
Encrypt. + Pre-proc. (ms)	308.5	1333.4	211.2	844.9
Decrypt. + Post-proc. (ms)	591.4	2518.2	392.0	1568.2
Enc. Denoising (*min*)	17.42	74.21	2.79	11.19
Denoising with 3 stages	Optimized from [33]		Our denoising	
N (size image $N \times N$)	1024	2048	1024	2048
t	65537	65537	257^2	257^2
Cipher Exp. (ratio)	652.0000	326.0000	179.5000	179.5000
δ	1.00342	1.00342	1.00374	1.00374
Bit security (Eq.(8))	≈ 255	≈ 255	≈ 223	≈ 223
L (multipplicative depth)	14	14	8	8
Encrypt. + Pre-proc. (ms)	797.2	1594.4	211.2	844.9
Decrypt. + Post-proc. (ms)	2311.7	4623.5	588.1	2352.3
Enc. Denoising (*min*)	98.12	196.25	2.80	11.20

Table 2: Performance of Encrypted PRNU detection (2048×2048 image, PRNU of 16 Mpixels, $L = 11$, $t = 257^5$, $\sigma = 8$, denoising with 2 stages)

Aligned detection with RLWE cryptosystem (bit security > 128)				
Parallelization	1	8	16	20
Cipher Exp. (ratio)	379.95			
δ	1.00396			
Bit security (Eq.(8))	≈ 205			
Encrypt. + Pre-proc. (ms)	3642.62			
Decrypt. + Post-proc. (ms)	26.87			
Enc. Detection (*min*)	128.33	16.05	8.03	6.53
Non-aligned detection with 2-RLWE cryptosystem (bit security > 128)				
Parallelization	1	8	16	20
Cipher Exp. (ratio)	113.13			
δ	1.00396			
Bit security (Eq.(8))	≈ 205			
Encrypt. + Pre-proc. (ms)	3642.62			
Decrypt. + Post-proc. (ms)	6878.50			
Enc. Detection (*min*)	1140.10	142.50	71.30	57.90

4.3. Performance of the PRNU extraction/detection

In order to evaluate the feasibility of the proposed scheme in terms of detection probabilities, we securely perform the PRNU detection test proposed in Sect. 2.1 as described in Sect. 3.3. To do so, we employed images from a database containing 2639 TIFF images from 16 digital single lens reflex camera devices [15, 12].

For a given target camera device, the fingerprint is extracted from $M = 50$ randomly chosen TIFF images, while crops of the JPEG-compressed version of the TIFF images with size 1536×1536 and quality factor 95 are considered for detection purposes. To test the H_1 hypothesis, after discarding the M images used for extraction, 20 different crops per image with random origins are considered on the images from the target camera, while H_0 hypothesis is tested by considering one crop per image for all images from each remaining camera device.

Figure 3 compares the performance of the detector in (2) with the Peak to Correlation Energy (PCE) state of the art detector [17], both when the widely used image denoising in [28] and when the proposed encrypted denoising filter with 2 stages (see Section 3) are used to obtain the residue of the test images. Notice that the denoising procedure from [28] is used for extraction ($\hat{\mathbf{K}}$ estimation) in all experiments, since the fingerprint is estimated in the clear.

The performance loss in the encrypted domain is mainly due to: 1) The simpler encrypted denoising algorithm, and 2) the simpler variance estimation on the detector in (2).

In spite of this slight loss in performance, the source attribution problem based on PRNU detection is feasible in the encrypted domain, achieving high true detection rates with low false alarm rates on JPEG test images, as shown in Fig. 3.

5. Conclusions

We have proposed a novel framework for secure outsourced camera attribution in a fully unattended way. As a fundamental block for both PRNU extraction and detection, we also present a new image denoising algorithm which improves the efficiency of the state of the art. Our solutions focus on unattended processing, where no interaction with the client is needed during the outsourced computation. This work opens up a new set of secure forensic applications, showing that suboptimal choices can be more adequate for homomorphic operations. Finally, we also evaluate our proposed scheme in terms of security, efficiency and performance, showing the feasibility of secure camera attribution in the encrypted domain.

References

[1] Nifty website. http://research.ncl.ac.uk/nifty/. Accessed: 15 September 2018. 1

[2] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of Learning with Errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015. 5

[3] J. Bajard, J. Eynard, M. A. Hasan, and V. Zucca. A Full RNS Variant of FV Like Somewhat Homomorphic Encryption Schemes. In *SAC 2016*, pages 423–442, 2016. 5

[4] L. Bondi, F. Pérez-González, P. Bestagini, and S. Tubaro. Design of Projection Matrices for PRNU Compression. In *IEEE WIFS 2017*, pages 1–6, Dec 2017. 1

[5] G. S. Çetin, Y. Doröz, B. Sunar, and W. J. Martin. An Investigation of Complex Operations with Word-Size Homomorphic Encryption. Cryptology ePrint Archive, Report 2015/1195, 2015. http://eprint.iacr.org/. 5

[6] H. Chen and K. Han. Homomorphic Lower Digits Removal and Improved FHE Bootstrapping. In *Advances in Cryptology - EUROCRYPT 2018*, pages 315–337, 2018. 3, 4

[7] M. Chen, J. Fridrich, and M. Goljan. Digital imaging sensor identification (further study). In *Proc. SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Content X*, volume 6505, Feb. 2007. 1

[8] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, Mar. 2008. 1, 2

[9] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *ASIACRYPT'11*, volume 7073 of *LNCS*, pages 1–20. Springer, 2011. 5

[10] J. H. Cheon and A. Kim. Homomorphic Encryption for Approximate Matrix Arithmetic. Cryptology ePrint Archive, Report 2018/565, 2018. https://eprint.iacr.org/2018/565. 2, 5

[11] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology - ASIACRYPT 2017*, pages 409–437, 2017. 5

[12] D.T. Dang-Nguyen, C. Pasquini, V. Conotter and G. Boato. RAISE – A raw images dataset for digital image forensics. In *Proc. 6th ACM Multimedia Systems Conference*, pages 219–224, Mar. 2015. 6

[13] J. Fan and F. Vercauteren. Somewhat Practical Fully Homomorphic Encryption. Crypt. ePrint Archive, Report 2012/144, 2012. 2, 3

[14] F. Gisolf, P. Barens, E. Snel, A. Malgoezar, M. Vos, A. Mieremet, and Z. Geradts. Common source identification of images in large databases. *Forensic science international*, 244:222–230, November 2014. 1

[15] T. Gloe and R. Böhme. The 'Dresden Image Database' for benchmarking digital image forensics. In *Proc. of the 25th Symp. On Applied Computing (ACM SAC 2010)*, volume 2, pages 1585–1591, Mar. 2010. 6

[16] M. Goljan and J. Fridrich. Camera identification from cropped and scaled images. In *Proc. SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Content X*, volume 6819, Mar. 2008. 1, 2

[17] M. Goljan, J. Fridrich, and T. Filler. Large scale test of sensor fingerprint camera identification. In *Proc. SPIE, Electronic Imaging, Media Forensics and Security XI*, volume 7254, pages 0I 1-0I 12, Feb. 2009. 1, 2, 6

[18] M. Griffin. Lowest degree of polynomial that removes the first digit of an integer in base p. https://mathoverflow.net/q/269282. Accessed: 15 September 2018. 3, 4

[19] G. C. Holst. *CCD Arrays, Cameras, and Displays*. SPIE Optical Engineering Press Bellingham, WA, 2nd edition, 1998. 1

[20] R. Hurley, S. Prusty, H. Soroush, R. J. Walls, J. Albrecht, E. Cecchet, B. N. Levine, M. Liberatore, B. Lynn, and J. Wolak. Measurement and analysis of child pornography trafficking on p2p networks. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 631–642, New York, NY, USA, 2013. ACM. 1

[21] P. Korus and J. Huang. Multi-scale analysis strategies in prnu-based tampering localization. *IEEE Transactions on Information Forensics and Security*, 12(4):809–824, April 2017. 1

[22] R. L. Lagendijk, Z. Erkin, and M. Barni. Encrypted Signal Processing for Privacy Protection. *IEEE SP Mag.*, 30(1):82–105, 2013. 1

[23] K. Lauter, M. Naehrig, and V. Vaikuntanathan. Can Homomorphic Encryption be Practical? Crypt. ePrint Archive, Report 2011/405, 2011. 5

[24] R. Lindner and C. Peikert. Better Key Sizes (and Attacks) for LWE-based Encryption. In *CT-RSA'11*, pages 319–339. Springer, 2011. 5

[25] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva. Blind PRNU-based image clustering for source identification. *IEEE Transactions on Information Forensics and Security*, 12(9):2197–2211, September 2017. 1

[26] C. Meij and Z. Geradts. Source camera identification using Photo Response Non-Uniformity on WhatsApp. *Digital Investigation*, 24:142 – 154, 2018. 1

[27] D. Micciancio and O. Regev. Lattice-based Cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009. 5

[28] M. K. Mihcak, I. Kozintsev, and K. Ramchandran. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. In *IEEE ICASSP 1999*, volume 6, pages 3253–3256, March 1999. 6

[29] M. Mohanty, M. Zhang, M. R. Asghar, and G. Russello. PANDORA: Preserving Privacy in PRNU-Based Source Camera Attribution. In *IEEE TrustCom/BigDataSE*, pages 1202–1207, Aug 2018. 2

[30] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Pearson, 1975. 4

[31] G. H. Owenson and N. J. Savage. The Tor Dark Net, 9 2015. 1

[32] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González. Multivariate Lattices for Encrypted Image Processing. In *IEEE ICASSP 2015*, pages 1707–1711, April 2015. 1, 2, 3

[33] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González. Image Denoising in the Encrypted Domain. In *IEEE WIFS 2016*, pages 1–6, Dec 2016. 1, 2, 3, 4, 5, 6

[34] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González. On Ring Learning with Errors over the Tensor Product of Number Fields. *CoRR*, abs/1607.05244, 2016. 2

[35] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González. Multivariate Cryptosystems for Secure Processing of Multidimensional Signals. *CoRR*, abs/1712.00848, 2017. 1

[36] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González. Number Theoretic Transforms for Secure Signal Processing. *IEEE Transactions on Information Forensics and Security*, 12(5):1125–1140, May 2017. 3, 4, 5

[37] F. Pérez-González, M. Masciopinto, I. González-Iglesias, and P. Comesaña. Fast sequential forensic detection of camera fingerprint. In *IEEE ICIP 2016*, pages 3902–3906, Sep. 2016. 1, 2

[38] J. R. Troncoso-Pastoriza, A. Pedrouzo-Ulloa, and F. Pérez-González. Secure genomic susceptibility testing based on lattice encryption. In *IEEE ICASSP 2017*, pages 2067–2071, 2017. 5