User Authentication via PRNU-Based Physical Unclonable Functions

Diego Valsesia, *Member, IEEE*, Giulio Coluccia, *Member, IEEE*, Tiziano Bianchi, *Member, IEEE*, and Enrico Magli, *Fellow, IEEE*

***Abstract*— Multifactor user authentication systems enhance security by augmenting passwords with the verification of additional pieces of information such as the possession of a particular device. This paper presents an innovative user authentication scheme that verifies the possession of one's smartphone by uniquely identifying its camera. High-frequency components of the photo-response nonuniformity of the optical sensor are extracted from raw images and used as a weak physical unclonable function. A novel scheme for efficient transmission and server-side verification is also designed based on adaptive random projections and on an innovative fuzzy extractor using polar codes. The security of the system is thoroughly analyzed under different attack scenarios both theoretically and experimentally.**

***Index Terms*— User authentication, PRNU, random projections, fuzzy extractors, polar codes.**

I. Introduction

THE very large diffusion in everyday life of web-based services like social networks, internet banking, cloud-based storage, requires the development of user authentication techniques that are both secure and user friendly [1]. In this sense, the traditional mechanism based on secret passwords shows several shortcomings. Security means that long and unpredictable passwords should be generated and remembered, which is not user friendly. As a consequence, short and easily predictable passwords are commonly reused, which considerably reduces the security of the system.

Recently, several solutions have been proposed for providing an additional level of security in current user authentication systems. A common approach is to resort to a multifactor authentication scheme, in which the knowledge of a secret password is complemented with the possession of one, or more, physical or software tokens [2]. Typical solutions currently implemented on several existing web services are the generation of one-time passwords (OTPs) on a dedicated token, or receiving a OTP by text message on the user's smartphone [3]–[5]. Even if multifactor authentication effectively solves the security problem, the existing solutions typically reduce user friendliness. As an alternative, several authors have proposed authentication systems based on the possession of unique signals that are not easily reproducible. A natural choice is using biometric traits like fingerprints, irises, or faces [6]–[8]. An innovative approach consists in deriving a secret from some physical characteristics of an integrated circuit that are deemed unique, implementing a so-called physical unclonable function (PUF) [9].

In this paper, we propose a novel authentication system that relies on an unclonable physical property of digital image sensors named photo-response non-uniformity (PRNU). The PRNU is a sensor-specific multiplicative noise pattern that has enjoyed great popularity in the last decade because it can be used to solve several forensic problems. Examples of its many applications are: determining which camera has acquired a given photo [10], [11], clustering collections of images by their source camera [12], [13], camera-based image retrieval [14], [15] and detecting and localizing image forgeries [16], [17].

The concept proposed in this paper is to use the PRNU of the camera sensor of the user's smartphone as a *weak PUF* [9], that can be used as a possession factor in a multifactor authentication scheme, or even employed in a single step authentication protocol. Due to the ubiquitous diffusion of smartphones, such a system is potentially much user friendlier than existing solutions, enabling the implementation of an application that automatically acquires pictures, computes a compact code derived from the sensor PRNU and transmits it to a remote verification server requiring minimal or no user interaction. However, turning this idea into a practical authentication system requires to solve several important problems, as well as rigorously show the security of such solutions.

First, the PRNU survives JPEG compression, as well as some image processing operations, and it can be found in photos that are publicly available, e.g., on social networks [18]. Luckily, the PRNU is inevitably degraded by such operations, while in the framework of user authentication, the legitimate user has full control over the camera and could extract the PRNU with an arbitrarily high quality. In the following, we consider extracting the PRNU from RAW images and keeping only its high-frequency components. Since JPEG compression acts as a lowpass filter, the high-frequency components are unavailable or severely degraded in publicly available images and can only be estimated if one has access to the raw data.

Manuscript received October 26, 2016; revised February 13, 2017 and April 11, 2017; accepted April 18, 2017. Date of publication April 24, 2017; date of current version May 10, 2017. This work is supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement n.279848. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Siwei Lyu. *(Corresponding author: Diego Valsesia.)*

The authors are with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy (e-mail: diego.valsesia@polito.it; giulio.coluccia@polito.it; tiziano.bianchi@polito.it; enrico.magli@polito.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2697402

Second, the PRNU has the same size as the image sensor. Sending a complete PRNU signal over a mobile connection could be impractical in several scenarios, as well as storing the reference PRNUs of a large number of users at the server side. In this case, we propose to compress the PRNU using random projections. Recent results show that this technique can reduce the PRNU size by several orders of magnitude, without significantly affecting the matching performance [14]. Moreover, this also provides an additional security layer since the actual PRNU is never disclosed and if a compressed PRNU is compromised this can be revoked and replaced by a freshly generated compression.

Lastly, the server should not store a copy of the PRNU, or its compressed version. This problem can be solved by resorting to techniques used for biometric template protection [19], [20]. Namely, we present an innovative implementation of a secure sketch and a fuzzy extractor based on polar codes, which is specifically tailored to compressed PRNUs. Since in the proposed system an attacker may have a partial knowledge of the PRNU from publicly available photos, the proposed construction incorporates a specific coding technique for the wiretap channel based on polar codes, which effectively prevents the attacker from gaining access to the system.

A. Related Works and Contribution

The idea of using high frequency components of PRNU has been recently introduced in a different context in [21]. The authors considered the case of fingerprint-copy attacks [22], where an attacker wants to plant a fingerprint in an image but only has access to JPEG images of the camera, while the defender has access to RAW data. The user authentication scenario significantly differs from a copy attack and provides unique requirements. Our goal is to show that an attacker that can only access JPEG-compressed images cannot reliably estimate the high-frequency components of the PRNU that the legitimate user employs as fingerprint. In our analysis, the legitimate user has full control over the raw image quality, and the number of images that can be used to generate the reference and test fingerprints. The attacker potentially has access to a large number of high-quality JPEG images and tries to extract a fingerprint that is highly correlated with the legitimate one. In this work, we assume that an attacker can only access public images in JPEG format and we do not consider the possible theft of RAW images. With respect to [21] we also provide a different fingerprint extraction method that is not constrained to work on 8×8 blocks. A significantly larger database with RAW and JPEG images, mostly from smartphone cameras, has been assembled in order to test attacks with hundreds of high-quality JPEG images.

The use of random projections for biometric template protection has been proposed in a number of works [23]–[26], and later extended also to PUFs [27]. With respect to existing papers, we introduce a novel adaptive random projection technique, similar to a technique proposed in [28] and then further expanded and carefully analysed in [29]. Moreover, using the PRNU as a PUF requires an ad-hoc design of the fuzzy extractor, for which we provide an original construction based on polar codes and a rigorous security analysis. Finally, we provide a rigorous security analysis of the whole proposed system under different attack scenarios.

Very recently, the authors of [30] proposed to combine several device sensor features, including PRNU, and apply machine learning for smartphone authentication. The paper provides some interesting insights on the distinctiveness of smartphone sensors, however security issues are not addressed and a complete authentication system is not discussed. The possibility of using PRNU for authentication is also discussed at high level in this recent contribution [31], but no technical solutions are proposed, and a rigorous security analysis is not provided.

B. Paper Organization

The paper is organized as follows. In Section II we provide some background on PRNU, random projections, fuzzy extractors, and polar codes. A high level description of the proposed system is given in Section III, while the technical details regarding fingerprint estimation/compression and user verification are discussed in Section IV and Section V, respectively. In Section VI we provide a rigorous security analysis of the proposed system under different scenarios, while in Section VII we validate the system through extensive experiments. Finally, conclusions are drawn in Section VIII.

II. Background

The following subsections provide some background material to help the reader understanding the rest of the paper. We first (Sec. II-A) present some notation used throughout the paper. Sec. II-B recalls the basics of PRNU of digital imaging sensors. Sec. II-C introduces random projections, a useful dimensionality reduction method. Sec. II-D discusses fuzzy extractors, a set of techniques to extract uniform randomness from a source that is not exactly reproducible. Finally, Sec. II-E reviews polar codes, a channel coding technique.

A. Notations

Lower-case (upper-case) bold symbols denote real-valued vectors (matrices). Lower-case letters indicate scalars or bit strings. Upper-case letters denote random variables. Symbols $\mathbb{P}$ and $\mathbb{E}$ denote the probability and expectation operators, respectively.

The predictability of a random variable A is measured by the min-entropy, defined as $H_\infty(A) = -\log(\max_a \mathbb{P}(A = a))$. A variable whose min-entropy is m bits is as hard to predict as a uniformly random string of m bits.

If the adversary observes a variable B which is correlated with A, the expected predictability of A can be expressed by the average min-entropy of A given B, defined as $\tilde{H}_\infty(A|B) = -\log(\mathbb{E}_b[2^{H_\infty(A|B=b)}])$.

It is also useful to define how much two random variables differ using the statistical distance between variable A and B, defined as $d_S(A, B) = \frac{1}{2}\sum_v |\mathbb{P}(A = v) - \mathbb{P}(B = v)|$.

Table I summarises the main symbols used throughout the paper, along with their description.

TABLE I
SYMBOLS

$\mathbf{k}$	Uncompressed PRNU fingerprint
$\mathbf{O}_a$	Attacker's set of JPEG photos
$\mathbf{\Phi}$	Sensing matrix
$\mathbf{y}$	Real-valued random projections
$\mathbf{l}$	Subsampling locations
m_{pool}, m	No. of random projections before/after subsampling
ρ	Correlation coefficient
w	Compressed fingerprint
x	Secret string
h	Hash of secret
s	Sketch
$Q^{(l)}$, $Q^{(a)}$	Legitimate/Attacker channels
p_l, p_a	Bit error probability for legitimate/attacker channel
r	Random bits for polar code
t	No. of random bits for polar code
k	No. of bits of secret

B. PRNU

PRNU [11], [32] of imaging sensors is a property unique to each sensor array due to the different ability of each individual optical sensor to convert photons to electrons. This difference is mainly caused by impurities in silicon wafers and its effect is a noise pattern affecting every image taken by that specific sensor. Hence, the PRNU can be thought of as a spread–spectrum *fingerprint* of the sensor.

The literature on camera forensics [11], [33] widely considers the PRNU as unique for each camera since it has very large entropy and therefore the probability of two cameras having the same pattern is negligible. For instance, Bayram et al. [34] estimate the entropy of the PRNU to be 20 bits per pixel, and considering that the PRNU has the same pixel size as the sensor, and the value for each pixel is uncorrelated with the others, the PRNU has very large discriminative power. Being a multiplicative pattern, its strength with respect to other noise sources depends on the brightness of the acquired image.

The PRNU characterizing one sensor can be extracted from a set of images (typically, 20 to 50 smooth images are enough). The procedure to extract the fingerprint $\mathbf{k}$ of a sensor from a set of pictures depends on the model used to characterize the optical sensor. The sensor output $\mathbf{o}$ can be modelled as

$$\mathbf{o} = \mathbf{o}^{\mathsf{id}} + \mathbf{o}^{\mathsf{id}} \cdot \mathbf{k} + \mathbf{e}, \tag{1}$$

where $\mathbf{o}^{\mathsf{id}}$ is the ideal sensor output, $\mathbf{o}^{\mathsf{id}} \cdot \mathbf{k}$ is the PRNU term and $\mathbf{e}$ collects other sources of noise. Assuming to be able to obtain through proper filtering a denoised version of $\mathbf{o}$, referred to as $\mathbf{o}^{\mathsf{dn}}$, then this can be used as an approximation of the ideal sensor output and subtracted from each side of (1) to obtain the so-called *noise residual*, which can be modeled as:

$$\mathbf{r} = \mathbf{o} - \mathbf{o}^{\mathsf{dn}} = \mathbf{o} \cdot \mathbf{k} + \tilde{\mathbf{e}}, \tag{2}$$

where $\tilde{\mathbf{e}}$ accounts for $\mathbf{e}$ and for the non-idealities of the model [11]. Supposing that a certain number C of images is available, the maximum likelihood estimate $\hat{\mathbf{k}}$ can be obtained as

$$\hat{\mathbf{k}} = \sum_{l=1}^{C} \left(\mathbf{r}^{(l)} \cdot \mathbf{o}^{(l)}\right) \Bigg/ \sum_{l=1}^{C} (\mathbf{o}^{(l)})^2. \tag{3}$$

To improve further the quality of the estimation, artifacts shared among cameras of the same brand or model can be removed by subtracting row and column averages. In the case of color images, the estimation must be performed separately on each color channel, and then an RGB–to–gray conversion can be applied.

Finally, a pair of fingerprint vectors $\mathbf{k}_1$, $\mathbf{k}_2$ is typically compared using their correlation coefficient, defined as

$$\rho = \frac{\mathbf{k}_1^T \mathbf{k}_2}{\|\mathbf{k}_1\|_2 \|\mathbf{k}_2\|_2}.$$

C. Random Projections

Random projections (RPs) are a method for dimensionality reduction [35]. A collection $\mathcal{X} \subset \mathbb{R}^n$ of signals living in a high-dimensional space can be embedded with low distortion into low-dimensional representations $\mathcal{Y} \subset \mathbb{R}^m$ (also known as measurements, or random projections, with $m < n$) by computing inner products with random vectors. In matrix form this is written as $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$, for $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$, and where $\mathbf{\Phi}$ is often referred to as sensing matrix. Measurements can also be quantized to achieve more storage-efficient representations. The key property of random projections is that they approximately preserve distances. A classic result is that real-valued random projections, where the sensing matrix is made of independent and identically distributed (i.i.d.) Gaussian entries, are a mapping that satisfies the Johnson-Lindenstrauss (JL) lemma [36], meaning that ℓ_2 distances are nearly preserved. A key property following from the JL lemma is that the number of measurements m depends only on the desired distortion on distances between signals introduced by the embedding, and on the number of signals that are to be embedded but not on the dimensionality of input space n.

Of particular interest are binary random projections that are computed with a sensing matrix made of i.i.d. Gaussian entries, and then quantized to one bit by keeping the sign of the measurement. The Hamming distance between the resulting binary vectors approximately preserves the angle between the signals in the original space [37], i.e.,

$$\mathbb{P}\left(\text{sign}(\boldsymbol{\phi}_i^T \mathbf{u}) = \text{sign}(\boldsymbol{\phi}_i^T \mathbf{v})\right) = 1 - \frac{\theta}{\pi},$$

being $\theta = \cos^{-1}\left(\frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}\right)$, and $\boldsymbol{\phi}_i$ the i-th row of $\mathbf{\Phi}$.

It is often impractical to use a fully random sensing matrix, either because the high dimensionality of the signals requires to generate too many random numbers or because performing the full matrix-vector product is too computationally intensive. Circulant matrices with randomized column signs [38] are an appealing solution because they allow to generate only the first row of the sensing matrix and compute the measurements using the FFT.

In [14], [15], RPs were used to perform dimensionality reduction of PRNU patterns, showing significant gains in terms of storage requirements as well as in the complexity of the match or search in large database operations.

D. Fuzzy Extractors

Fuzzy extractors denote a set of techniques for extracting nearly uniform randomness from sources of information that are neither exactly reproducible nor uniformly distributed [19], [20]. These techniques were originally developed

for generating strong keys from biometric data, however they can be applied to any form of noisy data used for authentication, like PUFs. More precisely, such techniques rely on two primitives: 1) a *fuzzy extractor* that extracts nearly uniform randomness from an input in an error-tolerant way, i.e., close inputs are guaranteed to generate the same randomness; 2) a *secure sketch* producing public information about a secret input w that does not reveal anything about w, yet allows to recover w when combined with another value that is sufficiently close to w.

In our scheme, we will employ a slightly relaxed definition of secure sketches and, in turn, of fuzzy extractors, that accounts for a negligible probability of not recovering the secret input w. This definition applies when the error pattern on w can be modeled by a binary symmetric channel with crossover probability p (BSC-p).

Definition 1: An $(n, m, \tilde{m}, p, \alpha)$-secure sketch *consists in a pair of functions* $\mathsf{SS} : \{0,1\}^n \to \{0,1\}^*$ *and* $\mathsf{Rec} : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ *with the following properties:*

1) Correctness*: if w' is the output of a BSC-p when the input is w, then* $\mathsf{Rec}(w', \mathsf{SS}(w)) = w$ *with probability at least* $1 - \alpha$.
2) Security*: if* $H_\infty(W) = m$ *then* $\tilde{H}_\infty(W|\mathsf{SS}(W)) \geq \tilde{m}$.

Definition 2: An $(n, m, \ell, p, \alpha, \epsilon)$-fuzzy extractor *consists in a pair of functions* $\mathsf{Gen} : \{0,1\}^n \to \{0,1\}^\ell \times \{0,1\}^*$ *and* $\mathsf{Rep} : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^\ell$ *with the following properties:*

1) Correctness*: if* $(x, s) = \mathsf{Gen}(w)$ *and w' is the output of a BSC-p when the input is w, then* $\mathsf{Rep}(w', s) = x$ *with probability at least* $1 - \alpha$.
2) Security*: if* $(x, s) = \mathsf{Gen}(w)$ *and* $H_\infty(W) = m$ *then* $d_S((X, S), (U_\ell, S)) \leq \epsilon$*, where U_ℓ is a uniformly distributed string of ℓ bits.*

From the above definitions, it is evident that a fuzzy extractor can be constructed on top of a secure sketch, provided that one can extract sufficiently uniform randomness from the secret input w [20].

E. Polar Codes

Polar coding is a channel coding technique introduced by Arıkan in [39] that provably achieves the capacity of binary memoryless symmetric (BMS) channels. Let us consider the $2^n \times 2^n$ matrix $\mathbf{G}_n = \mathbf{G}^{\oplus n}$, obtained as the n-fold Kronecker product of the kernel matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Suppose a block of $N = 2^n$ bits are encoded by $\mathbf{G}_n$ and then fed to N independent copies of a BMS channel. If we consider the N equivalent bit channels from the input bits to the output of the corresponding BMS channel, it is shown in [39] that as n grows these bit channels polarize, i.e., if C denotes the capacity of the BMS channels, a fraction nC of the bit channels have a capacity approaching 1, whereas the other bit channels have a capacity close to zero. The rationale of polar coding is to transmit information bits only on the good channels, whereas the other bits are set to zero.

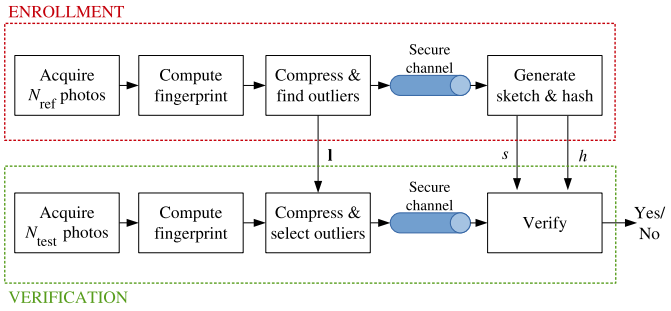


Fig. 1. System block diagram.

Given a BMS channel $Q(y|x)$, its performance can be analyzed using the Bhattacharyya parameter, defined as

$$Z(Q) = \sum_y \sqrt{Q(y|0)Q(y|1)}. \tag{4}$$

The usual design strategy for polar codes is to compute the parameter $Z(Q_i)$ for all the N equivalent bit channels Q_i, $i = 1, \ldots, N$, and use only the $k = Rn$ channels having smaller $Z(Q_i)$ for transmitting information bits, where $R < C$ is the desired rate of the code. This is equivalent to using as generator matrix of the code the submatrix obtained from the corresponding k rows of $\mathbf{G}_n$. An important result for polar codes states that the probability of error of a successive cancellation decoder (SCD) can be upper bounded as [39]

$$P_e \leq \sum_{i \in \mathcal{A}} Z(Q_i) \tag{5}$$

where $\mathcal{A}$ denotes the set of bit channels used by the code.

In practice, the computation of the parameters $Z(Q_i)$ is tractable only for the binary erasure channel, since for every other channel the output alphabet of Q_i grows exponentially. However, useful upper and lower bounds on $Z(Q_i)$ can be computed quite efficiently using channel degrading and upgrading techniques [40], [41].

III. Proposed Technique

The main idea of the proposed technique is to use the PRNU fingerprint of the optical sensor of a user's device, e.g. a smartphone or a tablet, as a PUF for authentication. An overview block diagram is shown in Fig. 1.

In a first phase, the user enrolls into the system by providing a high quality estimate of the device fingerprint, obtained from a certain number of photos acquired in controlled conditions. Instead of directly sending the fingerprint, which usually consists in millions of real numbers, the user first compresses it by means of random projections. The user also stores some side information related to the seed of the pseudorandom number generator and the positions of the entries with largest magnitude (outliers) within those random projections, which will be then used in the authentication phase. The exact algorithm as well as the role of the outliers will be made clear in the following sections. At the server side, the compressed fingerprint is processed by a fuzzy extractor. Namely, the server extracts a uniformly random bit string from the compressed fingerprint and stores a secure hash of this bit string, together with a secure sketch of the fingerprint.

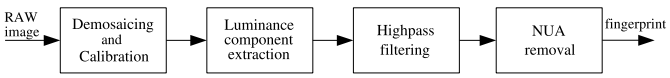


Fig. 2. Fingerprint extraction procedure.

In the authentication phase, the user reproduces a noisy version of the device fingerprint by acquiring a fresh set of photos and compressing the resulting fingerprint according to the stored side information. The server then uses the fuzzy extractor scheme for reproducing the secret bit string from the received compressed fingerprint and the secure sketch, and compares the recovered bit string with the stored secure hash. If the user provides a version of the compressed fingerprint sufficiently close to the enrolled one, then the server can reproduce the same bit string of the enrollment phase and grants access to the system; otherwise, it denies access.

With respect to existing authentication systems based on biometrics/PUFs and fuzzy extractors, the proposed technique introduces two important novelties. First, the actual PRNU-based PUF is obtained by means of a novel compression technique based on adaptive random projections. Besides reducing the size of the transmitted fingerprint, this technique provides an additional security layer, as will be discussed in the following sections. Secondly, the PRNU of a sensor is not a completely private information, since it can be approximated from public photos acquired by that sensor. In order to solve this problem, we introduce a novel fingerprint estimation technique that relies on RAW data acquired by the sensor, which is not usually available from public photos. Moreover, we design the fuzzy extractor in such a way that it is robust with respect to illegitimate fingerprints obtained from public photos. In the following sections, we will discuss the details of both PRNU-based PUF computation and user verification based on the proposed fuzzy extractor.

IV. PRNU-Based PUF

This section describes in detail the client-side functional blocks introduced in the previous section concerning fingerprint extraction and compression.

A. Fingerprint Extraction

In order to devise a PUF for the authentication scheme, we propose to use high frequency components of the PRNU pattern estimated from RAW photos. The motivation is to obtain a fingerprint that is capable of discriminating different sensors and, at the same time, that is uncorrelated with any estimate that can be extracted from JPEG data. In the following we propose an extraction method from RAW images and then model JPEG images to devise an extraction method that better approximates the output of the extraction method from RAW images, in order to study an attack tailored to the proposed system. Since the RAW acquisition process can be controlled and the fingerprint extraction has to run efficiently on a user's smartphone, we suppose that the user acquires approximately flat images to streamline the extraction process.

1) Extracting High-Frequency PRNU From RAW Data: The process described in this section is summarized in Fig. 2. It is important to notice that since the authentication process relies on photos taken at that specific moment rather than using already available photos, the acquisition process can be controlled, i.e., it is possible to select the shooting parameters so to acquire photos that will yield the highest quality estimates of the PRNU. In particular, the exposure should be as high as possible without saturating the pixel values and the content should be uniform and possibly out of focus so that the scene can be well approximated by a constant value. Moreover, we can use a set of fixed values for ISO sensitivity, aperture, and focal length, so that different PRNU estimates will not be affected by those shooting parameters.

The RAW image is first demosaiced and color calibrated to obtain image $\mathbf{o} = [\mathbf{r}, \mathbf{g}, \mathbf{b}]$. The luminance component of such image is then obtained by applying the transformation

$$\boldsymbol{\lambda} = 0.299\mathbf{r} + 0.587\mathbf{g} + 0.114\mathbf{b} \ .$$

It is possible to extract an estimate of the high-frequency components of the PRNU pattern to be used as fingerprint by means of a highpass filter (hereafter denoted as HPF) applied to the luminance component of the demosaiced and color calibrated image. This filter can be implemented as a product in the DCT domain. In Sec. VII we explore two possible solutions where the filtering is performed blockwise (to mimic JPEG), or on the whole image. Hence a first estimate of the fingerprint is:

$$\mathbf{k}^{\mathsf{RAW}} = \mathsf{HPF}\left(\boldsymbol{\lambda}\right) \approx \mathbf{o}^{\mathsf{id}} \cdot \mathsf{HPF}\left(\mathbf{k}\right) + \mathbf{e}'.$$

Since the scene, represented by the term $\mathbf{o}^{\mathsf{id}}$, is flat it is clear that a highpass version of the PRNU pattern is observed. When multiple images $\mathbf{o}^{(l)}$ are available the fingerprint is jointly estimated as

$$\mathbf{k}^{\mathsf{RAW}} = \frac{\sum_l \mathbf{o}^{(l)} \cdot \mathsf{HPF}\left(\boldsymbol{\lambda}^{(l)}\right)}{\sum_l (\mathbf{o}^{(l)})^2}. \tag{6}$$

However, some artifacts may be present, either because of the blockiness introduced by a blockwise highpass filter or because of non-unique artifacts (NUA) [33] such as CFA interpolation, linear pattern, etc.. Such artifacts may introduce ambiguities in the camera detection process and should be removed. Hence, as a post-processing operation we remove row and column means in a checkerboard pattern and perform Wiener filtering to suppress any periodic artifact. Such post-processing operations are well known in the literature to suppress non-unique artifacts. Some cameras may provide corrections for optical distortions, typically involving a resampling step. Such artifacts are notably difficult to remove and lower the detection rate in camera identification applications [42], [43]. However, since we access the RAW data before any kind of post-processing, our PRNU estimates will not contain this kind of artifacts.

2) Extracting High-Frequency PRNU From JPEG Data: The scope of this section is to develop a method to extract a fingerprint from JPEG images in such a way that it achieves the highest possible correlation with the fingerprint extracted from RAW data as described in the previous section. This method is what would be used by an attacker having access to publicly available JPEG images.

JPEG compression uses a quantization table in the discrete cosine transform (DCT) domain to shrink the coefficients in a way that preserves perceived visual quality. This typically results in many high frequency coefficients being set to zero, thus losing all the information associated to high frequencies. If we follow the usual model for the acquired image presented in (1) we can approximate the image after JPEG compression, denoted as $\mathbf{o}^{\mathsf{JPG}}$, as a lowpass filtered version of the original, where the cutoff frequency of the filter essentially depends on the compression quality factor. We denote such lowpass filter with LPF.

$$\mathbf{o}^{\mathsf{JPG}} = \mathsf{LPF}\left(\mathbf{o}^{\mathsf{id}} + \mathbf{o}^{\mathsf{id}}\mathbf{k} + \mathbf{e}\right) = \mathsf{LPF}\left(\mathbf{o}^{\mathsf{id}}\right) + \mathsf{LPF}\left(\mathbf{o}^{\mathsf{id}}\mathbf{k}\right) + \mathbf{e}'.$$

Conventionally, one wants to estimate $\mathbf{k}$ by means of flat images so that $\mathbf{o}^{\mathsf{id}} \approx \mathsf{const.}$, obtaining after denoising the noise residual

$$\mathbf{r} = \mathbf{o}^{\mathsf{id}} \cdot \mathsf{LPF}\left(\mathbf{k}\right) + \mathbf{e}''. \tag{7}$$

It is clear that using flat images one can only observe a lowpass version of the PRNU pattern. However, if the image is not flat, the noise residual is

$$\mathbf{r} = \mathsf{LPF}\left(\mathbf{o}^{\mathsf{id}}\mathbf{k}\right) + \mathbf{e}''. \tag{8}$$

The idea is to replicate the extraction procedure used for RAW data, i.e. highpass filtering, but on the noise residual since the attacker does not have control on the quality of the JPEG images and the flat assumption may or may not hold. First, the luminance noise residual is extracted, then it is filtered with the same highpass filter used to extract the RAW fingerprint and finally a weighted average as in (6) is performed if multiple images are available. Finally, mean removal and Wiener filtering are performed as post-processing operations. Notice that according to (8) the noise residual is a lowpass version of the PRNU modulated by the input image. If highpass filtering is performed one obtains

$$\mathbf{r}' = \mathsf{HPF}\left(\mathsf{LPF}\left(\mathbf{o}^{\mathsf{id}}\mathbf{k}\right)\right) + \tilde{\mathbf{e}} = F\left(\mathbf{o}^{\mathsf{id}}\mathbf{k}\right) + \tilde{\mathbf{e}}.$$

This means that if the highpass filter is properly designed only a very weak signal can be observed due to the leakage of the combination of the two filters, represented by F. The experimental results show that higher correlation values can be achieved by this method instead of using the conventional method that does not include the highpass filter in the extraction chain. Notice that this procedure is not optimal, as the optimal extraction method would retrieve $\mathsf{HPF}\left(\mathbf{k}\right)$. However, this would require solving a challenging deconvolution problem to disentangle the PRNU term from the image content in the observed $\mathsf{LPF}\left(\mathbf{o}^{\mathsf{id}}\mathbf{k}\right)$.

We remark that the existence of methods that improve the estimation of the high-frequency PRNU components beyond what we proposed in this section does not compromise the overall authentication scheme described in this paper. In fact, the legitimate user has full access to the RAW data provided by the device and can increase the difficulty of an attack by increasing the cutoff frequency of the filter or increasing the number of acquired photos to achieve arbitrarily high fingerprint quality levels.

B. Fingerprint Compression

Since the fingerprint must be sent to a server for verification purposes, it is of paramount importance to compress it to a size that makes transmission over bandlimited channels manageable. The objective of the compression step is to transform the real-valued, high-dimensional fingerprint into a short binary code. Correlated fingerprints must be mapped into similar binary codes.

In Sec.II-C we presented binary-quantized random projections, characterized by the property that their Hamming distance concentrates around the angle between the original uncompressed fingerprints. One can therefore use them to obtain compact binary codes. Since the fingerprints are high-dimensional objects, a complexity issue arises in the calculation of the random projections. This can be solved by using circulant random matrices with randomized column signs, as shown in [14]. For such matrices, only the first row must be generated at random and the matrix-vector product can be efficiently performed using the FFT.

In this paper, however, we propose to use a modified version of such random projections, that we call adaptive random projections [29]. The key property of adaptive random projections is that some randomness is traded for a better (more compact) representation of signals correlated with a particular signal of interest. This solution has three main advantages in the context of the proposed user authentication system:

- more compact codes allow to save transmission time;
- more compact codes allow a more efficient and easier design of the fuzzy extractor at server side;
- adaptivity allows to preserve as much as possible of the inter-class correlation gap between fingerprints extracted from JPEG data and fingerprints extracted from RAW data; this also simplifies the design of the channel code in the fuzzy extractor because it maximizes the margin between the bit-error probability observed by a legitimate user and that observer by an attacker.

During the registration phase, a high-quality version of the fingerprint $\mathbf{k} \in \mathbb{R}^n$ is available. A vector $\boldsymbol{\phi}$ with n i.i.d. Gaussian entries is generated and circularly convolved with $\mathbf{k}$ using the FFT to implement a circulant sensing matrix. The result of this operation is first subsampled to keep the first m_{pool} values. The $m < m_{\text{pool}}$ entries with largest magnitude are identified and their locations $\mathbf{l}$ stored locally on the user device as side information. Finally, the sign of the entries at those locations is saved as compressed fingerprint w of m bits. During the verification phase, a test fingerprint $\mathbf{k}'$ is presented for compression, and its projections are computed by keeping only the sign of the entries indexed by $\mathbf{l}$.

The value of m_{pool} determines the storage overhead required for the location information. Choosing m outliers from a larger pool improves the adaptivity to the reference signal but increases the storage overhead. The effect of adaptivity is shown in Fig. 3 where the expected value of the Hamming distance between the binary codes is plotted against the correlation coefficient between the original uncompressed fingerprints. Notice that the adaptive method allows to achieve smaller values for the Hamming distance and maximize the

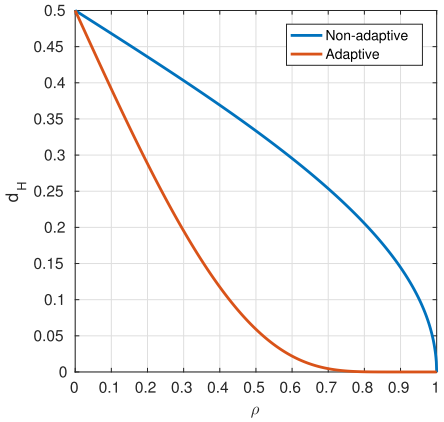


Fig. 3. Adaptive random projections. $m_{\text{pool}} = 2^{20}$, $m = 2^{15}$.

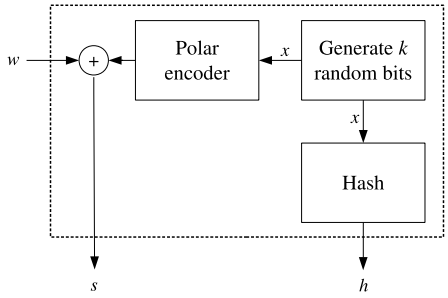


Fig. 4. Generation of sketch and hash.

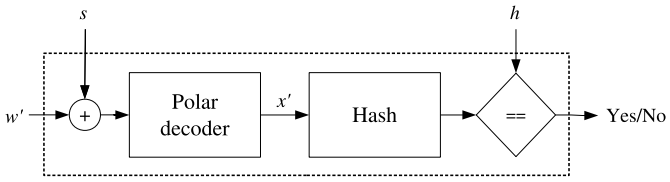


Fig. 5. User verification.

margin between the class of invalid fingerprints having very low correlation values and the class of valid fingerprints having higher correlation values.

V. User Verification

Due to the non-exact repeatability of the PRNU fingerprint estimation procedure, during the verification phase the user will produce a compressed fingerprint that contains some bit errors with respect to the enrolled fingerprint. Moreover, an attacker having access to a certain number of publicly available JPEG photos acquired by the user's device may also be able to provide a noisy version of the enrolled fingerprint, albeit with a much higher number of bit errors.

In order to cope with this scenario, we design a novel fuzzy extractor scheme. The proposed solution is based on the fuzzy commitment scheme proposed in [44] and a coding scheme for the wiretap channel that uses polar codes [45]. The proposed scheme is based on a generation function and a verification function, whose block diagrams are depicted in Fig. 4 and Fig. 5, respectively.

During the enrollment phase, the server generates a uniformly random string x of k bits. From this secret string, the server computes a hash $h = \mathsf{SH}(x)$, where $\mathsf{SH}(\cdot)$ denotes a secure hashing function, and a secure sketch $s = w \oplus C(x)$, where w is the compressed fingerprint received from the user and C denotes a (m, k) error correcting code based on polar codes. The server then discards x and stores h and s.

During the verification phase, the server computes the k-bit string $x' = D(w' \oplus s)$, where w' is the noisy fingerprint and D denotes the decoding algorithm of the error correcting code, and authenticates the user only if $\mathsf{SH}(x') = h$.

The error correcting code is not a standard (m, k) polar code, but is constructed according to the scheme in [45]. Let us assume a BSC-p_l for the legitimate channel and a BSC-p_a for the attacker channel, and denote them as $Q^{(l)}$ and $Q^{(a)}$, respectively. The code construction requires choosing a security parameter $t > m(1 - H_2(p_a))$, where $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ denotes the binary entropy function, and verifying that $k + t < m(1 - H_2(p_l))$. Then, we define two subsets $\mathcal{A}_l$ and $\mathcal{A}_a \subset \mathcal{A}_l$ of the indices $i = 1, \ldots, N$ satisfying

$$|\mathcal{A}_l| \;\; = k + t, \;\; \forall i \in \mathcal{A}_l, j \notin \mathcal{A}_l, Z(Q_i^{(l)}) \leq Z(Q_j^{(l)})$$
$$|\mathcal{A}_a| = t, \;\; \forall i \in \mathcal{A}_a, j \in \mathcal{A}_l \setminus \mathcal{A}_a, Z(Q_i^{(a)}) \leq Z(Q_j^{(a)})$$

The encoder generates t uniformly random bits r, assigns them to the bit channels in $\mathcal{A}_a$, and maps the k message bits x onto the remaining channels in $\mathcal{A}_l \setminus \mathcal{A}_a$. The code is then generated by using the corresponding rows in $\mathbf{G}_m$. In order to take into account the randomization in the encoding process, in the following the encoder function will be denoted as $C(x, r)$. The decoder simply applies the SCD to the received codeword and discards the t bits corresponding to $\mathcal{A}_a$. It can be checked that the above construction verifies Definition 2:

Proposition 3: If $H_\infty(W) = m$, then the above construction is an $(m, m, k, p_l, \alpha, 0)$-fuzzy extractor, where $\alpha = \sum_{i \in \mathcal{A}_l} Z(Q_i^{(l)})$. Moreover, there exist m^* and $\beta < 1/2$ such that, for $m \geq m^*$, we have $\alpha \leq 2^{-m^\beta}$.

Proof: The correctness property follows from the fact that, if $w' = w \oplus e$, then $w' \oplus s = C(x, r) \oplus e$. Hence, the scheme is equivalent to transmitting $C(x, r)$ through $Q^{(l)}$ and, according to (5), a polar code designed for $Q^{(l)}$ has a probability of block error upper bounded by α. The security property derives from the fact that X is uniformly random and, for $H_\infty(W) = m$, X and S are independent. The last claim is a direct consequence of Theorem 1 in [45]. ■

VI. Security Analysis

The security of the proposed authentication scheme depends on the probability that an attacker gains access to the system. In the following, we will make two important security assumptions: *i*) the attacker does not have access to any RAW photos of the user's device; *ii*) the attacker can access only a finite number N of JPEG photos of the user's device.

Regarding the other system parameters, we will consider four different security scenarios, depending on whether the attacker can access the parameters stored on the server, i.e., the sketch s and the secure hash h, or the parameters stored by the client, i.e., the seed for generating $\mathbf{\Phi}$ and the location vector $\mathbf{l}$. We will show that the proposed system is secure in all the scenarios, although with different security levels.

A. Scenario 1: Server and Client Are Both Safe

This is the best case scenario, when the attacker can only access N JPEG photos. Let us denote these photos as $\mathbf{O}_a$. It is easy to see that these photos give no information about the enrolled compressed fingerprint w:

Lemma 4: If $\mathbf{\Phi}$ is drawn from a Gaussian ensemble and $w = \text{sign}(\mathbf{\Phi_l k})$, where $\mathbf{\Phi_l}$ denotes the submatrix of $\mathbf{\Phi}$ formed by the rows indexed by $\mathbf{l}$, then $\tilde{H}_\infty(W|\mathbf{O}_a) = m$.

Proof: We note that $I(W; \mathbf{O}_a) \leq I(W; \mathbf{k}) = 0$. The first inequality holds since $\mathbf{O}_a \rightarrow \mathbf{k} \rightarrow W$ is a Markov chain, whereas the last equality is due to the fact that W depends only on the spherical angle of $\mathbf{\Phi k}$ that, for $\mathbf{\Phi}$ drawn from a Gaussian ensemble, is independent from $\mathbf{k}$ and uniformly distributed on the unit sphere (see Lemma 1 in [46]). Hence, $\tilde{H}_\infty(W|\mathbf{O}_a) = H_\infty(W) = m$. ∎

In this scenario, the best an attacker can do is to draw w uniformly at random. The probability of success of this attack can be upper bounded as follows:

Theorem 5: Under Scenario 1, the probability of success of the attacker verifies $P_a \leq 2^{-k}$.

Proof: Under a fixed x, the attacker succeeds if he/she chooses $w \in \mathcal{C}_x$, where $\mathcal{C}_x = \{w | \mathsf{SCD}_k(w) = x\}$ and $\mathsf{SCD}_k(\cdot)$ denotes the output of the successive cancellation decoder on the k bits corresponding to x. Since x is uniformly random and not under the attacker's control, the probability of success is obtained as

$$P_a = \mathbb{E}_x \left[\mathbb{P}\left(w \in \mathcal{C}_x\right)\right] = \mathbb{E}_x \left[\frac{|\mathcal{C}_x|}{2^m}\right] = \frac{1}{2^{m+k}} \sum_x |\mathcal{C}_x| \leq \frac{1}{2^k}$$

where the last inequality holds since $\mathcal{C}_x$ are disjoint sets. ∎

A possible remark is that Lemma 4 does not hold if $\mathbf{\Phi}$ is circulant. In such cases, we can have the same result by modifying fingerprint compression as $w = \text{sign}(\mathbf{\Phi_l k}) \oplus \mathbf{b}$, where $\mathbf{b}$ is a uniformly random m-bit vector that the client stores along with $\mathbf{\Phi}$ and $\mathbf{l}$.

Notice that in this scenario the security is guaranteed by the secrecy of the projection matrix and an attacker may even have access to RAW photos without compromising the system.

B. Scenario 2: Server Is Compromised

Under this scenario, the attacker can see $\mathbf{O}_a$, s, and h. Let us consider an intermediate scenario in which the attacker observes only s. Since $H_\infty(W) = m$, according to Proposition 3 the fuzzy extractor verifies the security properties with $\epsilon = 0$, i.e., the secret x is indistinguishable from a uniformly random k-bit vector, even when observing s. This can be equivalently stated as $\tilde{H}_\infty(W|S) = k$, i.e., s is a (m, m, k, p_l, α)-secure sketch (this result can be proved using Lemma 4.5 in [20]).

Thanks to Lemma 4, the above result holds also when the attacker observes $\mathbf{O}_a$ and s, since $\tilde{H}_\infty(W|S, \mathbf{O}_a) = \tilde{H}_\infty(W|S)$. This follows from the fact that $\mathbf{O}_a \rightarrow W \rightarrow S$ is a Markov chain and W is independent from $\mathbf{O}_a$. In both scenarios, the attacker can only guess x', pick a random r, and try whether $w' = C(x', r) \oplus s$ is accepted by the system, which has a success probability $P_a = 2^{-k}$.

When the attacker also observes h, the system does not satisfy any more the above statistical security definition, since it is easy to verify $\tilde{H}_\infty(W|S, H = h) = \log N_C(h)$, where $N_C(h)$ is the number of collisions yielding value h in the secure hashing function, when computed over all possible x. In this scenario, the attacker is able to verify any feasible w until he/she finds a w' satisfying $w' = s \oplus C(x', r)$, with $\mathsf{SH}(x') = h$. Nevertheless, with a proper secure hashing function the system is computationally secure:

Proposition 6: If $\mathsf{SH}(\cdot)$ is ideal, then under Scenario 2 the expected complexity of an attack is $N_a = \Omega\left(2^{\min\{k, n_{hash}\}}\right)$ operations, where n_{hash} is the hash length in bits.

Proof: If $k > n_{\text{hash}}$, a pre-image attack on a n_{hash}-bit ideal secure hash requires $2^{n_{\text{hash}}}$ guesses on average. If $k < n_{\text{hash}}$, finding the right x requires $\frac{2^k+1}{2}$ guesses on average. ∎

C. Scenario 3: Client Is Compromised

In this scenario, the attacker sees $\mathbf{O}_a$, $\mathbf{\Phi}$, and $\mathbf{l}$. Intuitively, the attacker can exploit $\mathbf{O}_a$ to estimate a degraded version of the reference fingerprint $\mathbf{k}$ and then compress it with $\mathbf{\Phi_l}$ in order to produce a noisy estimate of w. The relationship between $\mathbf{O}_a$ and $\mathbf{k}$ is very difficult to characterize in a rigorous way, since it depends on both the content of the images in $\mathbf{O}_a$ and the acquisition and compression pipeline of the device. Hence, in this case it is extremely difficult, if not impossible, finding a useful characterization of $\tilde{H}_\infty(W|\mathbf{O}_a, \mathbf{\Phi}, \mathbf{l})$.

We will introduce a further security assumption to have a tractable analysis of this scenario, i.e., we will assume that the channel described by $\mathbb{P}(\mathbf{O}_a, \mathbf{\Phi}, \mathbf{l}|w)$ is a degraded version of m independent BSC-p_a channels. More formally, we assume that there exists a channel $P(\mathbf{O}_a, \mathbf{\Phi}, \mathbf{l}|w')$ such that

$$\mathbb{P}(\mathbf{O}_a, \mathbf{\Phi}, \mathbf{l}|w) = \sum_{w'} P(\mathbf{O}_a, \mathbf{\Phi}, \mathbf{l}|w') \prod_i Q^{(a)}(w'_i|w_i). \quad (9)$$

The above property is equivalent to assuming that an attacker using maximum likelihood estimation cannot obtain a better estimate of w than that obtained observing the output of m independent BSC-p_a channels when the input is w.

Under this assumption, it is possible to exploit the properties of the wiretap channel coding method of Sec. V for proving the following upper bound on the attacker's probability of success:

Theorem 7: There exist $c < 1$ and m^* such that, for $m > m^*$, the probability of success of the attacker verifies

$$P_a \leq \left(1 - H_2^{-1}(c)\right)^k. \quad (10)$$

Proof: Since $t > m(1 - H_2(p_a)) = mI(Q^{(a)})$, where $I(Q^{(a)})$ denotes the symmetric capacity of $Q^{(a)}$, according to Proposition 20 in [45] there exist $c < 1$ and m^* such that, for $m > m^*$ and for $i \in \mathcal{A}_l \setminus \mathcal{A}_a$, we have $I(Q_i^{(a)}) \leq 1 - c$. Using Fano's inequality, the probability of error of the SCD when decoding x_i can be lower bounded as

$$\begin{aligned} p_{e,i} &\geq H_2^{-1}\left(H(X_i|W, X^{(i-1)}, R)\right) \\ &= H_2^{-1}\left(1 - I(Q_i^{(a)})\right) \geq H_2^{-1}(c). \end{aligned}$$

Hence, the proof follows from $P_a = \prod_{i \in \mathcal{A}_l}(1 - p_{e,i}) \leq \prod_{i \in \mathcal{A}_l \setminus \mathcal{A}_a}(1 - p_{e,i})$ and $|\mathcal{A}_l \setminus \mathcal{A}_a| = k$. ∎

Actually, if one is able to provide an upper bound on $I(Q_i^{(a)})$ for every $i \in \mathcal{A}_l \setminus \mathcal{A}_a$, the bound in (10) can be tightened. For example, using the technique described in [41] upgraded versions of $Q_i^{(a)}$ can be computed, from which it is possible to find values δ_i such that $p_{e,i} \geq \delta_i$ and obtain $P_a \leq \prod_{i \in \mathcal{A}_l \setminus \mathcal{A}_a} (1 - \delta_i)$.

D. Scenario 4: Server and Client Are Both Compromised

This is the worst case scenario, in which only the high quality fingerprint $\mathbf{k}$ remains hidden from the attacker. As in Scenario 2, let us consider an intermediate scenario in which the attacker observes $\mathbf{O}_a$, $\mathbf{\Phi}$, $\mathbf{l}$, and the sketch s. Even if we rely on the assumption in (9), in this scenario the attacker is not constrained by the performance of the SCD. Since $w \oplus s$ is a valid codeword of the polar code, a computationally unbounded attacker could apply maximum likelihood (ML) decoding in order to obtain the most likely codeword $C(x')$ and attack the system with $C(x') \oplus s$. Let us define $\mathbf{\Theta}_a = (\mathbf{O}_a, \mathbf{\Phi}, \mathbf{l})$. The success probability of this attack is easily obtained as

$$P_a = \mathbb{E}_{\mathbf{\Theta}_a, s}\left[\max_x \mathbb{P}(x|\mathbf{\Theta}_a, s)\right] = 2^{-\tilde{H}_\infty(X|\mathbf{\Theta}_a, S)}. \quad (11)$$

The expectation over $\mathbf{\Theta}_a, s$ is justified since these quantities are not under the attacker's control.

Proposition 8: Under the assumption in (9)*, the success probability p_a is upper bounded by the expected ML decoding performance of the polar code on the attacker's channel $\tilde{Q}^{(a)}(z|w) = \prod_i Q^{(a)}(z_i|w_i)$, i.e.,*

$$P_a \leq \frac{1}{2^{k+t}} \sum_{z,r} \max_x \tilde{Q}^{(a)}(z|C(x,r)). \quad (12)$$

Proof: The proof follows from the chain of inequalities:

$$\begin{aligned}
P_a &= \mathbb{E}_{\mathbf{\Theta}_a, s}\left[\max_x \mathbb{P}(x|\mathbf{\Theta}_a, s)\right] = \sum_{\mathbf{\Theta}_a, s} \max_x \mathbb{P}(x, \mathbf{\Theta}_a, s) \\
&= \sum_{\mathbf{\Theta}_a, s} \max_x \mathbb{P}(\mathbf{\Theta}_a|x, s)\mathbb{P}(x, s) \\
&= \frac{1}{2^{k+m}} \sum_{\mathbf{\Theta}_a, s} \max_x \mathbb{P}(\mathbf{\Theta}_a|x, s) \quad (13) \\
&= \frac{1}{2^{k+m+t}} \sum_{\mathbf{\Theta}_a, s, r} \max_x \mathbb{P}(\mathbf{\Theta}_a|x, s, r) \quad (14) \\
&= \frac{1}{2^{k+m+t}} \sum_{\mathbf{\Theta}_a, s, r} \max_x \mathbb{P}(\mathbf{\Theta}_a|w = C(x,r) \oplus s) \quad (15) \\
&= \frac{1}{2^{k+m+t}} \sum_{\mathbf{\Theta}_a, s, r} \max_x \sum_z P(\mathbf{\Theta}_a|z)\tilde{Q}^{(a)}(z|C(x,r) \oplus s) \\
&\leq \frac{1}{2^{k+m+t}} \sum_{\mathbf{\Theta}_a, s, r} \sum_z P(\mathbf{\Theta}_a|z) \max_x \tilde{Q}^{(a)}(z|C(x,r) \oplus s) \\
&= \frac{1}{2^{k+m+t}} \sum_{z, s, r} \max_x \tilde{Q}^{(a)}(z|C(x,r) \oplus s) \\
&= \frac{1}{2^{k+t}} \sum_{z,r} \max_x \tilde{Q}^{(a)}(z|C(x,r)) \quad (16)
\end{aligned}$$

where (13) holds since x, s are independent and are uniformly distributed, (14) makes explicit the marginalization over r, (15) follows since w is completely determined by x, r, and s, and (16) is due to the symmetry of $Q^{(a)}$. ∎

Unfortunately, computing the right hand side of (12) given $Q^{(a)}$ has exponential complexity in m, so we cannot come up with a tractable upper bound. However, a reasonable conjecture is that ML decoding would not dramatically improve over the SCD and an exponentially decreasing bound as in (10) is still valid.

Even if we cannot obtain a formal proof that the system achieves statistical security in this scenario, we can prove that the system verifies Wyner's weak security definition [47]:

Theorem 9: Under the assumption in (9)*, the system satisfies*

$$\lim_{k \to \infty} \frac{I(X; \mathbf{\Theta}_a, S)}{k} = 0.$$

Proof: We have the following chain of mutual information inequalities

$$\begin{aligned}
I(X; \mathbf{\Theta}_a, S) &= I(X; S) + I(X; \mathbf{\Theta}_a|S) \quad (17) \\
&= I(X; \mathbf{\Theta}_a|S) \quad (18) \\
&\leq I(X; W'|S) \quad (19) \\
&= I(X; W', S) \quad (20) \\
&= I(X; W' \oplus S) \quad (21)
\end{aligned}$$

where (17) is the chain rule for mutual information, (18) is true since X and S are independent, (19) comes from the data processing inequality, (20) is again due to the independence of X, S, and (21) holds since $X \to W' \oplus S \to (W', S)$ is a Markov chain. For the last claim, it suffices to notice that $W' = W \oplus E$, where E is the error pattern of the BSC-p_a, $S = W \oplus C(X, R)$, and W and E are both independent of X, so X does not say anything more about the joint distribution of (W', S) once $W' \oplus S = C(X, R) \oplus E$ is disclosed.

Now, let $U = W' \oplus S$. It is easy to see that U is the output of a BSC-p_a when the input is $C(X, R)$. Let us consider the bit channels used by the polar code, i.e., $i \in \mathcal{A}_l$. Since $|\mathcal{A}_l| < mI(Q^{(l)})$, according to Theorem 1 in [45], there exist m^* and $\beta < 1/2$ such that, for $m > m^*$, $Z(Q_i^{(l)}) \leq 2^{-m^\beta}/m$. If we define the set

$$\mathcal{A}_a^{(\beta)} = \left\{ i | Z(Q_i^{(a)}) \leq \frac{2^{-m^\beta}}{m} \right\}$$

since $|\mathcal{A}_a| > mI(Q^{(a)})$, there exists m^{**} such that, for $m > m^{**}$, $|\mathcal{A}_a^{(\beta)}| < |\mathcal{A}_a|$. Moreover, thanks to Lemma 4 in [45], for $m > \max\{m^*, m^{**}\}$, we have $\mathcal{A}_a^{(\beta)} \subset \mathcal{A}_l$ which implies $\mathcal{A}_a^{(\beta)} \subset \mathcal{A}_a$.

Let V denote the message transmitted on the bit channels $i \in \mathcal{A}_l \setminus \mathcal{A}_a^{(\beta)}$. It is immediate to verify $V = X \| R_{\mathcal{A}_a \setminus \mathcal{A}_a^{(\beta)}}$, i.e., V is the concatenation of X and the random bits in R that do not belong to $\mathcal{A}_a^{(\beta)}$. Hence, the proof follows from

$$\begin{aligned}
I(X; U) &= I(X \| R_{\mathcal{A}_a \setminus \mathcal{A}_a^{(\beta)}}; U) - I(R_{\mathcal{A}_a \setminus \mathcal{A}_a^{(\beta)}}; U) \\
&\leq I(X \| R_{\mathcal{A}_a \setminus \mathcal{A}_a^{(\beta)}}; U) = I(V; U)
\end{aligned}$$

and Theorem 8 in [45], which states $\lim_{k\to\infty} I(V;U)/k = 0$. ∎

The above claims do not hold anymore when the attacker observes also h, since a computationally unbounded adversary could try every possible x until $h = \mathsf{SH}(x)$. Nevertheless, we can lower bound the expected complexity of an attack according to the success probability when h is not known.

Proposition 10: If $\mathsf{SH}(\cdot)$ *is ideal, then under Scenario 4 the expected complexity of an attack is* $N_a = \Omega\left(2^{\min\{\tilde{H}_\infty(X|\Theta_a,S),n_{hash}\}}\right)$.

Proof: If $k > n_{\text{hash}}$, see Prop. 6. If $k < n_{\text{hash}}$, the expected number of random guesses is

$$N_a = \sum_{i=1}^{2^k} i\,P_{a,i} \geq \sum_{1=1}^{P_a^{-1}} i\,P_a = \frac{P_a^{-1}+1}{2}$$

where $P_{a,i}$ is the probability of success of the ith guess and P_a is the probability of success of the most likely guess, which is given in (11). ∎

Actually, the above bound is a bit pessimistic, since it assumes that the adversary knows the likelihood of each guess. A computationally bounded adversary will not be able to evaluate such probabilities exactly, since this requires ML decoding. However, finding a practical bound on the complexity of the attack remains an open problem.

VII. Experimental Results

The following experiments are aimed at proving the functionality of all the system blocks presented in Fig.1. In particular, the experiments aim at verifying that: i) the fingerprint extraction procedure generates fingerprints that can discriminate cameras; ii) fingerprint extraction from RAW images allows to achieve significantly higher correlation than from JPEG images, thus clearly separating the two classes; iii) compression via adaptive random projections still allows such separation; iv) the fuzzy extractor based on polar codes introduces a negligible probability of rejection of legitimate users.

First (Sec. VII-A), the fingerprint extraction technique is validated by analyzing the correlation values obtained with the fingerprints extracted from RAW and JPEG images. In order to perform such tests, we created a database of 14 cameras (13 Android smartphones and 1 DSLR) able to acquire RAW images. The devices are reported in Table II, which also shows the ordering in which they appear throughout the experiments. Notice that we have multiple devices of the same model for some of the smartphones. This allows us to check that the fingerprint is actually able to discriminate the specific device. The smartphones use Android as it allows to acquire RAW images through the Camera 2 API [48] supported by version 5.0 or higher of the operating system. We remark that no Apple devices have been used since at the time of the experiment we did not find a reliable way to get RAW data.[1] Following the assumptions detailed in Section III we acquired RAW images in controlled conditions, setting a suitable exposure, minimal ISO sensitivity and shooting out-of-focus pictures of walls in order to have a uniform content. Each camera has at least 150 RAW photos that have been then partitioned into training (20 photos) and test sets (at least 130 photos). JPEG images were acquired with the default camera application at maximum resolution and maximum JPEG quality. Whenever further manual options were available we forced the lowest ISO setting and controlled the exposure as well. Most of the devices use the standard JPEG quantization matrix with quality factor QF=95, one device uses QF=97 and the DSLR has a custom higher quality matrix. We acquired at least 200 flat images and 200 images with content. For the DSLR camera we used 1016 flat images and 2978 natural images, all well exposed and unprocessed, to test an attack with a significant number of high quality images.

[1] iOS 10 released in September 2016 now supports RAW acquisition.

TABLE II
List of Devices Used in the Experiments

Brand	Model	Camera ID
LG	Nexus 5	1
Nikon	D3100	2
LG	Nexus 5X	3
LG	Nexus 5	4
Motorola	Nexus 6	5
Motorola	Nexus 6	6
LG	Nexus 5	7
LG	G4 H815	8
LG	Nexus 5X	9
LG	Nexus 5X	10
Huawei	Nexus 6P	11
LG	Nexus 5	12
Samsung	Galaxy S7	13
LG	Nexus 5X	14

Then (Sec. VII-B) compression by means of adaptive random projections is introduced. The results are then used to determine the parameters of a polar code that allows for the decoding of legitimate fingerprints while blocking illegitimate ones.

Sec. VII-C shows the performance of the designed polar code using synthetic data and an ideal BSC channel.

Sec. VII-D clarifies the error bounds for an attacker when the client is compromised introduced in VI-C, by explicitly calculating them for the designed polar code.

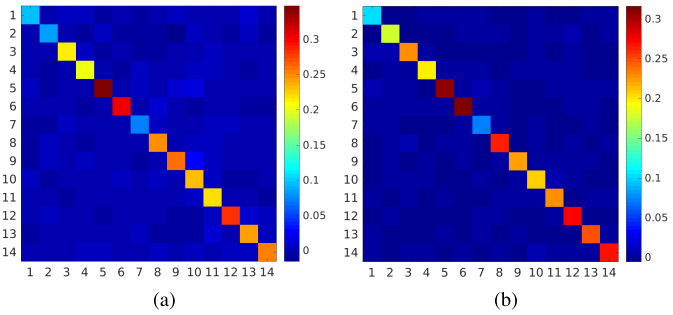
Finally, VII-E uses the polar code on the actual fingerprints generated from real photos.

A. Performance of Uncompressed Fingerprints

We tested the following fingerprint extraction methods from RAW images:

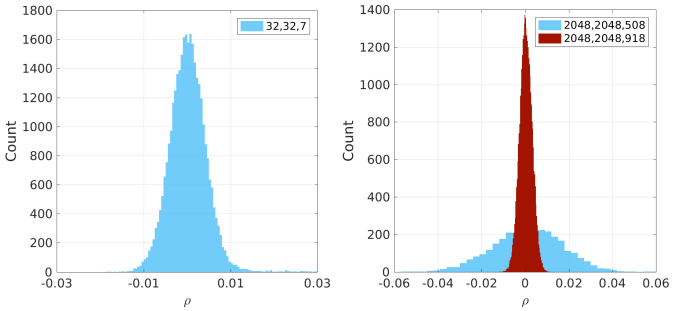
- *blockwise*: the image is partitioned into blocks of size 32×32, highpass filtered blockwise with a $(32, 32, 7)$ filter, meaning that only the DCT coefficients on the bottom-right 7 antidiagonals are kept. Postprocessing mean removal and Wiener filtering operations are global;
- *full*: the image is cropped to a 2048×2048 area, and this area is filtered with a $(2048, 2048, 918)$ filter. Postprocessing mean removal and Wiener filtering are applied after highpass filtering.

We generated the reference fingerprint of a camera averaging the PRNU estimates obtained from N_{ref} images according to Eq.(6). We fixed this value to $N_{\text{ref}} = 20$ as a good compromise between obtaining the highest possible quality



(a) (b)

Fig. 6. Correlation matrix. Each camera in the dataset has a reference fingerprint extracted from RAW images and a test fingerprint also extracted from RAW images is tested against the references of all devices ($N_{\text{ref}} = 20$, $N_{\text{test}} = 2$). (a) Blockwise. (b) Full.



(a) (b)

Fig. 7. Histogram of all off-diagonal correlations. With reference to the experiment of Fig. 6 all the off-diagonal values are collected. Additionally, a second filter (2048,2048,508) is tested for the full method to show the effect of fewer independent coefficients ($N_{\text{ref}} = 20$, $N_{\text{test}} = 2$). (a) Blockwise. (b) Full.

for the reference fingerprint and computational effort required by the user's smartphone in the enrollment phase. Similarly, test fingerprints are also obtained averaging multiple images, although fewer in order to have shorter delays during the verification phase. N_{test} represents the number of RAW images whose PRNU estimate is averaged to obtain the test fingerprint. We experimented with using from 1 up to 5 images.

1) Inter/Intra-Camera Detection: First, it is necessary to study whether such extraction method is discriminative enough, i.e., if it can correctly discriminate different camera sensors, and in particular suppress any artifact common to the same camera model. Fig. 6 shows the correlation coefficients between a test fingerprint (row) and the reference (column) in matrix form for the blockwise and full methods (warmer color means higher correlation). Visual inspection shows that the system seems able to discriminate between different cameras. Fig. 7 aggregates the results over the test sets for all the cameras to report histograms for the off-diagonal entries, i.e., the correlation achieved by a test fingerprint estimated from N_{test} RAW photos of a certain camera against the reference fingerprints of the other cameras (inter-camera detection). Instead, Table III reports the worst-case value for the entries on the diagonal, i.e. when a test fingerprint is correlated with the reference of its own camera (intra-camera detection). The worst-case value is the minimum such correlation that we observed. By comparing the values reached by the tails of 7 and Table III, it can be noticed that intra-camera correlations are always significantly larger than inter-camera correlations, allowing perfect discrimination between the devices, even if the model of the device is the same.

Notice that Fig. 7 also tests a filter with a different cutoff frequency to show that the distribution for non-matching cameras has a variance that is mostly determined by the cutoff frequency of the highpass filter. The higher the cutoff, the fewer coefficients will be maintained, thus losing discriminative power. As a first rough approximation the non-matching correlation coefficients are Gaussian with zero mean and variance $2/((c+1)c)$ for the full method and a $(\cdot,\cdot,c)$ filter. Thus a first tradeoff emerges in the choice of the filter that should have a high enough cutoff to ensure a low correlation with JPEG images, but at the same time low enough for the fingerprint to be discriminative.

TABLE III

MINIMUM INTRA-CAMERA CORRELATIONS WITH TEST FINGERPRINTS COMPUTED FROM RAW PICTURES

Camera ID	Method	Min $\lvert\rho\rvert$ (1 pic)	Min $\lvert\rho\rvert$ (2 pics)	Min $\lvert\rho\rvert$ (3 pics)	Min $\lvert\rho\rvert$ (4 pics)	Min $\lvert\rho\rvert$ (5 pics)
1	block	0.0709	0.1244	0.1589	0.1792	0.1911
	full	0.0803	0.1339	0.1631	0.1868	0.2071
2	block	0.0589	0.2625	0.2609	0.3198	0.3647
	full	0.1191	0.4833	0.4835	0.5624	0.6177
3	block	0.2042	0.2826	0.3363	0.3752	0.4079
	full	0.2164	0.2956	0.3507	0.3946	0.4274
4	block	0.1925	0.2652	0.3132	0.3563	0.3863
	full	0.1889	0.2522	0.3043	0.3425	0.3773
5	block	0.1963	0.1688	0.1788	0.2354	0.3291
	full	0.1296	0.1595	0.1495	0.2074	0.2848
6	block	0.2762	0.3685	0.4344	0.4803	0.5170
	full	0.2802	0.3739	0.4388	0.4849	0.5123
7	block	0.0539	0.0725	0.1134	0.1307	0.1397
	full	0.0593	0.0884	0.1238	0.1416	0.1498
8	block	0.2345	0.3214	0.3809	0.4256	0.4592
	full	0.2522	0.3433	0.4046	0.4482	0.4837
9	block	0.2089	0.2617	0.3196	0.3702	0.3978
	full	0.1931	0.2631	0.3164	0.3550	0.3852
10	block	0.1840	0.2084	0.2295	0.2956	0.3324
	full	0.1359	0.1806	0.1854	0.2455	0.2653
11	block	0.2132	0.2914	0.3465	0.3900	0.4209
	full	0.2185	0.2944	0.3543	0.3962	0.4287
12	block	0.0901	0.0682	0.0899	0.1286	0.1310
	full	0.1014	0.0974	0.1176	0.1522	0.1628
13	block	0.2333	0.3161	0.3753	0.4192	0.4528
	full	0.2357	0.3073	0.3576	0.3958	0.4484
14	block	0.2280	0.3094	0.3698	0.4112	0.4489
	full	0.2441	0.3319	0.3925	0.4356	0.4733

2) RAW vs JPEG Detection: In this section we are interested in intra-camera detection only, and, in particular, in determining if the correlation achieved by fingerprints extracted from JPEG images is substantially lower than that achieved by fingerprints extracted from RAW images. The fingerprint generated from JPEG images is only tested against the reference of the corresponding camera because this is the attack model we are interested in for the user authentication scenario, i.e. someone that tries to use the publicly available JPEG photos of someone to authenticate as her.

The experiments with JPEG images have been performed in the following way. Flat images and natural images are not mixed but used separately in order to check if content has any effect on the quality of the fingerprint. Out of all the available images, 20 random subsets of 100 photos each are selected and a fingerprint is estimated from each subset. Furthermore,

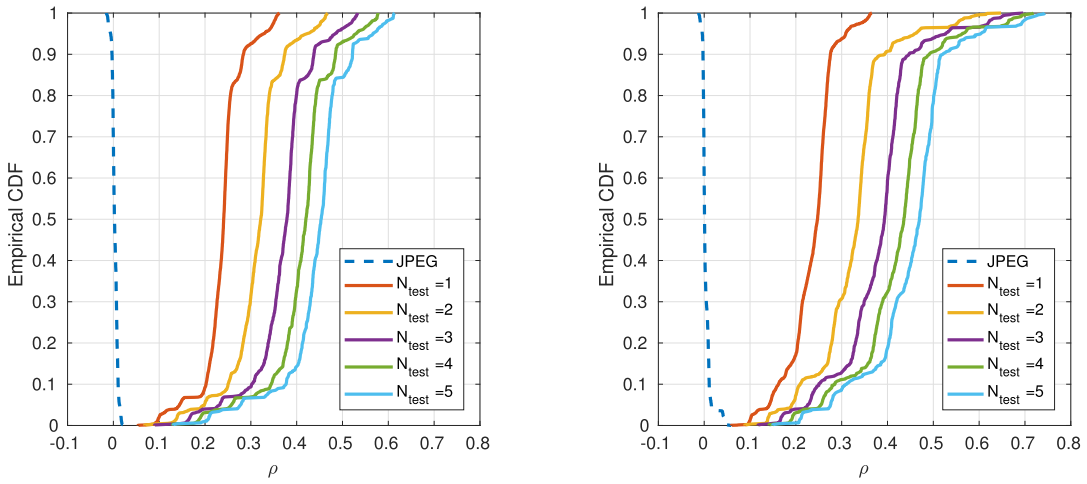


Fig. 8. Correlation cumulative distributions. Cumulative distribution of correlation values between the reference fingerprint extracted from RAW images and a test fingerprint extracted from RAW images of the same camera, for all available test fingerprints and cameras. Complementary cumulative distribution of correlation values between the reference fingerprint extracted from RAW images and a test fingerprint extracted from several JPEG images of the same camera, for all available test fingerprints and cameras, and all JPEG tests (20 attacks with 100 photos each and best attack with all the photos). (a) Blockwise. (b) Full.

a "best" attack is simulated using all the available images for that camera (typically at least 200).

Fig. 8 reports the results of the experiment over the whole dataset by plotting the cumulative distribution of all correlation values achieved for all the cameras. The reference fingerprint of each camera is generated from $N_{\text{ref}} = 20$ RAW images. The tests from RAW images use the images from the test sets (which are different from those used to generate the reference) and multiple configurations are tested by varying N_{test} from 1 to 5 pictures. Since there are at least 130 photos in test set for each camera, the $N_{\text{test}} = 1$ trace is composed of more than $130 \times 14 = 1820$ correlation values, while the $N_{\text{test}} = 5$ trace has more than $130/5 \times 14 = 364$ values. The decreasing curve in the same figure also shows the percentage of tests with JPEG images exhibiting a correlation lower than the value on the abscissa (i.e., the curve is the complementary CDF, i.e., 1 - CDF). This curve includes all the JPEG tests, i.e., both the 20 attacks with 100 photos each and the best attack with all the photos. It can be noticed that with the chosen system parameters there indeed exists a gap between the highest correlation achieved from JPEG data and lowest achieved from RAW data, appearing graphically where the tails of the empirical CDF reach 0. Notice that the lowest values of correlation appearing in the RAW data are due to cameras no.1 and no.8, which have rather dark reference RAW images. Despite such non-ideal conditions, they still display correlation values that are significantly higher than those obtained by using hundreds of JPEG images. Indeed, the performance of the RAW data could be further improved by increasing the exposure.

For reference, Table III reports the *minimum* correlation achieved by RAW data and the number of test photos used, while Table IV reports the *maximum* correlation achieved by JPEG data and the number of photos used.

Finally, Fig. 9 shows how the correlation value achieved by the fingerprint extracted from JPEG data varies as a function of the number of photos used in its estimation. It can be noticed that the curve quickly saturates and that a large number of photos is then needed to achieve minimal gains.

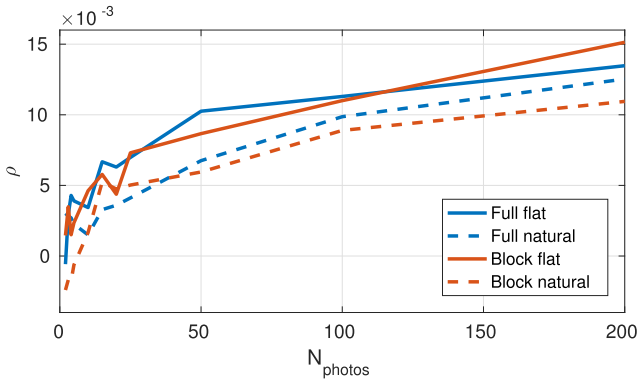


Fig. 9. JPEG correlation against number of used photos for camera no.7.

B. Performance Under Adaptive Random Projections

The experiment in this section is the same as the one on intra-camera detection on the whole dataset presented in the previous subsection on RAW vs JPEG detection. The difference is that it now uses fingerprints compressed with the adaptive random projection method instead of uncompressed fingerprints. The correlation coefficient previously used to measure similarity between uncompressed fingerprints is replaced by the Hamming distance between the binary strings representing compressed fingerprints. The purpose of the experiment is to determine the number m of adaptive random projections to be used to compress the fingerprints. A sufficient number of random projection is needed in order to have low embedding variance. i.e. the Hamming distance between compressed fingerprints tightly concentrates around the expected value of the embedding. The experiments also provide the parameters p_l and p_a, which are used to model the

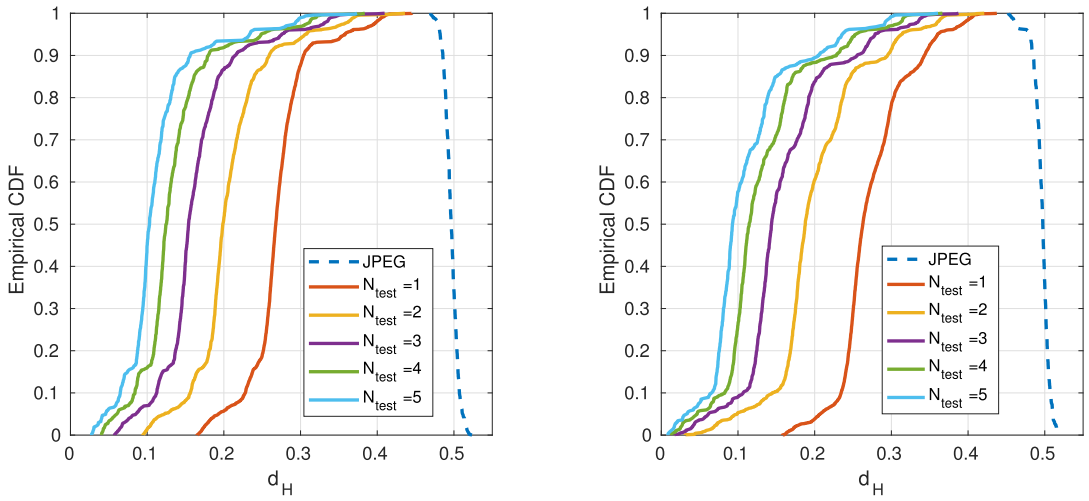


Fig. 10. Hamming distance cumulative distributions. ($m = 2^{15}$, $m_{\text{pool}} = 2^{20}$). Equivalent of Fig. 8 after compression. (a) Blockwise. (b) Full.

TABLE IV
MAXIMUM INTRA-CAMERA CORRELATIONS WITH TEST FINGERPRINTS COMPUTED FROM JPEG PICTURES

Camera ID	Photo type	No. of photos	Max $\|\rho\|$ (blockwise)	Max $\|\rho\|$ (full)
1	flat	210	0.0099	0.0046
	natural	228	0.0070	0.0033
2	flat	1016	0.0029	0.0022
	natural	2978	0.0024	0.0021
3	flat	215	0.0035	0.0039
	natural	257	0.0113	0.0119
4	flat	251	0.0115	0.0040
5	flat	204	0.0043	0.0053
	natural	243	0.0031	0.0037
6	flat	225	0.0043	0.0028
	natural	216	0.0115	0.0031
7	flat	401	0.0176	0.0153
	natural	221	0.0110	0.0125
8	flat	333	0.0151	0.0119
	natural	204	0.0037	0.0094
9	flat	228	0.0044	0.0060
	natural	218	0.0092	0.0140
10	flat	343	0.0055	0.0067
	natural	247	0.0113	0.0159
11	flat	230	0.0071	0.0066
	natural	228	0.0151	0.0145
12	flat	198	0.0122	0.0042
	natural	219	0.0122	0.0039
13	flat	300	0.0060	0.0200
	natural	209	0.0249	0.0590
14	flat	228	0.0055	0.0033
	natural	220	0.0125	0.0135

crossover probability for the worst-case BSC of the legitimate user and the best-case BSC of the attacker, respectively. Fig. 10 is the analogue of Fig. 8 after compression. It reports the empirical cumulative distribution of Hamming distances between the test fingerprints extracted from RAW images of the correct camera as well as one minus the cumulative distribution of Hamming distance when the fingerprints are extracted from JPEG images. It can be noticed that the curve follows the mapping from uncompressed correlations to Hamming distances shown in Fig. 3 since the chosen parameters are $m_{\text{pool}} = 2^{20}$ and $m = 2^{15}$. The values $m_{\text{pool}} = 2^{19}$ and $m = 2^{14}$ were also tested but we do not report them due to space constraints. The expected value of the embedding is approximately the same for both choices but the variance is lower when $m = 2^{15}$. The gap between the values of distances obtained with JPEG images and those obtained with RAW images is still present even though the variance in the distance measure introduced by the embedding used for compression causes the tails of the cumulative distributions to touch when a single test photo is used for verification. The minimum Hamming distances observed in the JPEG tests are: 0.469 (block, $m = 2^{15}$), 0.453 (full, $m = 2^{15}$), 0.464 (block, $m = 2^{14}$), 0.458 (full, $m = 2^{14}$). In light of the results, we can choose a value of $p_a = 0.45$ as a worst-case Hamming distance obtainable by an attacker using JPEG images and $p_l = 0.40$ as an upper bound on the distance typically obtained by legitimate users. We remark that the experiments show that a few legitimate test images exceed this value of p_l, especially when the number of test photos is low. However, this is due to some photos in our database having non-ideal exposure values. Indeed, a more careful exposure control yielding brighter images or an increase in the number of test/train photos could significantly lower the Hamming distance below the 0.40 threshold.

C. Polar Code With Synthetic Data

The purpose of this section is to choose a suitable set of parameters to design a polar code with security features for the wiretap channel as described in Sec. V. We use the p_l and p_a values derived from the experiments with compressed fingerprints to model the the worst-case channel seen by the legitimate user and the best-case channel seen by the attacker as a BSC-p_l and BSC-p_a, respectively. In the following experiment we tested the code by generating uniformly distributed binary vectors with length m bits that are encoded with the designed code, fed to independent BSC-p, and decoded using an SCD. Fig. 11 shows the block error probability as a function of the crossover probability p, averaged over 3 million independent experiments. Due to $p_a = 0.45$, the security parameter t was chosen to be $t = 256 > m(1-H_2(p_a)) = 236$ bits for $m = 2^{15}$ and $t = 128 > 118$ bits for $m = 2^{14}$. The secret length is

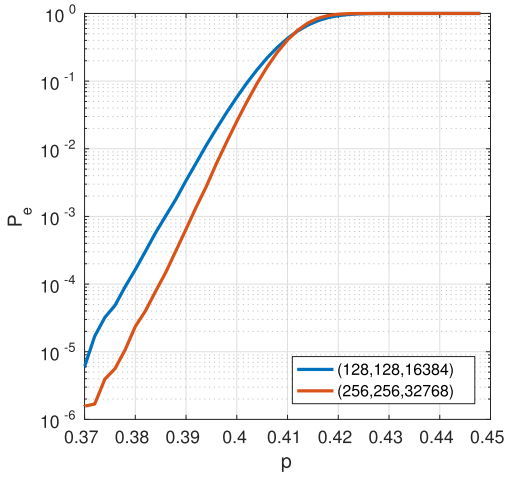


Fig. 11. Block error rates for the designed (k, t, m) polar code and successive cancellation decoding.

TABLE V
SUCCESS PROBABILITY OF AN ATTACK UNDER SCENARIO 3

k	t	m	μ	P_a
128	128	2^{14}	32	10^{-18}
256	256	2^{15}	32	10^{-37}

$k = 256$ bits and $k = 128$ bits respectively. The experiment confirms the expected behaviour of the code with a transition region between $p = 0.45$ and $p = 0.40$ and small block error rates for $p < 0.40$. Notice that the faster roll-off of the configuration with $m = 2^{15}$, $k = 256$, $t = 256$ allows to have a lower probability of rejection when a legitimate fingerprint has limited quality, i.e. when p is close to 0.4.

D. Attacker Error Bounds Under Scenario 3

Under attack scenario 3 (Sec.VI-C) the security of the system is guaranteed by the wiretap channel coding method of Sec. V. The upper bound on the probability of success of the attacker presented in (10) may seem rather loose. In this section we use the polar code designed in the previous section and explicitly characterize the performance of its bitchannels in order to provide a tighter bound than (10) and therefore show that an attacker has negligible probability of success.

In particular, we use the technique described in [41] to derive an upgraded version of the bitchannels for the previously designed code and a BSC-p_a with $p_a = 0.45$. Since we can compute the error probability δ_i for each upgraded bitchannel and that upper bounds the error probability of the actual bitchannel, a $(k+t)$-bit message is successfully decoded if all the chosen $(k + t)$ bitchannels do not incur in an error, i.e., $P_a \leq \prod_{i \in \mathcal{A}_l \setminus \mathcal{A}_a} (1 - \delta_i)$. Table V reports the numerical value of such upper bound on the probability of success of the attacker. It can be noticed that the success probability is upper bounded by a negligible value. Parameter μ controls the accuracy and computational complexity of the estimation procedure that avoids the exponential complexity of exact estimation (refer to [41] for further details).

TABLE VI
ACCEPTANCE RATE

Method	Correct RAW 1	2	3	4	5	Wrong RAW	JPEG
Block, 2^{14}	98.85%	99.82%	100%	100%	100%	0%	0%
Full, 2^{14}	99.38%	99.73%	100%	100%	100%	0%	0%
Block, 2^{15}	99.61%	99.66%	100%	100%	100%	0%	0%
Full, 2^{15}	99.82%	99.82%	100%	100%	100%	0%	0%

E. Polar Code With Real Data

In this section we employ the designed polar code using the compressed fingerprints extracted from actual photos. Table VI reports the acceptance rate, i.e., the percentage of times a user authenticating with RAW or JPEG images passed the server-side verification implemented using the previously designed polar code. In particular, the *correct RAW* columns refer to using a fingerprint extracted from RAW images of the correct camera, while *wrong RAW* refers to using a fingerprint extracted from RAW images of a different camera (since all the acceptance rates for all values of N_{test} are always 0 we summarised them in a single column).

VIII. CONCLUSIONS

We proposed a user authentication scheme based on using the high-frequency components of the PRNU pattern of optical sensors as a weak PUF. This was shown experimentally to provide a fingerprint that cannot be reliably extracted if only JPEG compressed images are available. Moreover, we devised a practical scheme to transmit such fingerprint to a verification server. In the proposed approach, the compression step is intimately linked to the server-side verification functionality implemented via a fuzzy extractor without the need to directly store the fingerprint.

We showed that the system is provably secure under different attack scenarios. One of the assumptions made in this paper is that a user does not publicly disclose RAW images acquired by the device to be used for authentication purposes. This is a quite reasonable assumption since it is not common practice to do so, especially for smartphones. Nevertheless, the security analysis shows that other elements of the system such as the random projection matrix can guarantee security even if RAW images are leaked.

Future work may focus on improving the technique for fingerprint extraction from RAW images. The current technique mimics the processing chain of JPEG in order to achieve orthogonality with fingerprints extracted from compressed images. However, this may not be the optimal method.

REFERENCES

[1] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Commun. ACM*, vol. 58, no. 7, pp. 78–87, Jun. 2015.
[2] W. E. Burr *et al.*, "Electronic authentication guideline," NIST—Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST Special Pub. 800-63-2, 2013.
[3] N. Haller, *The S/KEY One-Time Password System*, document RFC 1760, Internet Requests for Comments, Feb. 1995.
[4] D. M. Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, *HOTP: An HMAC-Based One-Time Password Algorithm*, document RFC 4226, Internet Requests for Comments, Dec. 2005.

[5] D. M. Raihi, S. Machani, M. Pei, and J. Rydell, *TOTP: Time-Based One-Time Password Algorithm*, document RFC 6238, Internet Requests for Comments, May 2011.

[6] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Commun. ACM*, vol. 43, no. 2, pp. 90–98, Feb. 2000.

[7] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, *Secure Remote Authentication Using Biometric Data*. Berlin, Germany: Springer, 2005, pp. 147–163.

[8] K. Nandakumar and A. K. Jain, "Biometric template protection: Bridging the performance gap between theory and practice," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 88–100, Sep. 2015.

[9] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[10] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[11] J. Fridrich, "Digital image forensics," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 26–37, Mar. 2009.

[12] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2010, pp. 1–5.

[13] C.-T. Li, "Unsupervised classification of digital images using enhanced sensor pattern noise," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 3429–3432.

[14] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Compressed fingerprint matching and camera identification via random projections," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1472–1485, Jul. 2015.

[15] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1439–1449, Sep. 2015.

[16] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," *Proc. SPIE*, vol. 6072, pp. 60720Y-1–60720Y-11, Feb. 2006.

[17] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 6231–6235.

[18] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," *Proc. SPIE*, vol. 7254, pp. 72540I-1–72540I-12, Feb. 2009.

[19] J.-P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Proc. Int. Conf. Audio Video-Based Biometric Person Authentication*, 2003, pp. 393–402.

[20] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[21] E. Quiring and M. Kirchner, "Fragile sensor fingerprint camera identification," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[22] M. Goljan, J. Fridrich, and M. Chen, "Defending against fingerprint-copy attack in sensor-based camera identification," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 227–236, Mar. 2011.

[23] A. B. J. Teoh, A. Goh, and D. C. L. Ngo, "Random multispace quantization as an analytic mechanism for BioHashing of biometric and random identity inputs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1892–1901, Dec. 2006.

[24] A. Teoh and C. T. Yuang, "Cancelable biometrics realization with multispace random projections," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1096–1106, Oct. 2007.

[25] E. T. Anzaku, H. Sohn, and Y. M. Ro, "Multi-factor authentication using fingerprints and user-specific random projection," in *Proc. 12th Int. Asia–Pacific Web Conf. (APWEB)*, Apr. 2010, pp. 415–418.

[26] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1877–1893, Sep. 2011.

[27] S. Shariati, L. Jacques, F. X. Standaert, B. Macq, M. A. Salhi, and P. Antoine, "Randomly driven fuzzy key extraction of unclonable images," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 4329–4332.

[28] T. Holotyak, S. Voloshynovskiy, O. Koval, and F. Beekhof, "Fast physical object identification based on unclonable features and soft fingerprinting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 1713–1716.

[29] D. Valsesia and E. Magli, "Binary adaptive embeddings from order statistics of random projections," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 111–115, Jan. 2017.

[30] I. Amerini, P. Bestagini, L. Bondi, R. Caldelli, M. Casini, and S. Tubaro, "Robust smartphone fingerprint by mixing device sensors features for mobile strong authentication," *Electron. Imag.*, vol. 8, pp. 1–8, Feb. 2016.

[31] J. Yan, "Novel security and privacy perspectives of camera fingerprints," in *Proc. 24th Int. Workshop Secur. Protocols*, Apr. 2016, pp. 1–6.

[32] J. Lukáš, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," *Proc. SPIE*, vol. 5685, pp. 249–260, Apr. 2005.

[33] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[34] S. Bayram, H. Sencar, and N. Memon, "Efficient sensor fingerprint matching through fingerprint binarization," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1404–1413, Aug. 2012.

[35] I. K. Fodor, "A survey of dimension reduction techniques," *Center Appl. Sci. Comput., Lawrence Livermore Nat. Lab.*, vol. 9, pp. 1–18, Jun. 2002.

[36] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemp. Math.*, vol. 26, pp. 190–206, May 1984.

[37] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. 34th Annu. ACM Symp. Theory Comput.*, May 2002, pp. 380–388.

[38] A. Hinrichs and J. Vybíral, "Johnson-lindenstrauss lemma for circulant matrices," *Random Struct. Algorithms*, vol. 39, no. 3, pp. 391–398, Oct. 2011.

[39] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[40] R. Pedarsani, S. H. Hassani, I. Tal, and E. Telatar, "On the construction of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2011, pp. 11–15.

[41] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

[42] M. Goljan and J. Fridrich, "Sensor-fingerprint based identification of images corrected for lens distortion," *Proc. SPIE*, vol. 8303, pp. 83030H-1–83030H-13, Feb. 2012.

[43] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in PRNU-based camera identification: A 'Dresden image database' case-study," in *Proc. Multimedia Secur.*, Sep. 2012, pp. 109–114.

[44] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM Conf. Comput. Commun. Secur.*, Nov. 1999, pp. 28–36.

[45] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6428–6443, Oct. 2011.

[46] T. Bianchi, V. Bioglio, and E. Magli, "Analysis of one-time random projections for privacy preserving compressed sensing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 313–327, Feb. 2016.

[47] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, 1975.

[48] *Android. Hardware. Camera2 Reference*. accessed on Apr. 28, 2017. [Online]. Available: https://developer.android.com/reference/android/hardware/camera2/package-summary.html

Diego Valsesia (S'13–M'17) received the Ph.D. degree in electronic and communication engineering from the Politecnico di Torino, Turin, Italy, in 2016, and the M.Sc. degree in telecommunications engineering from Politecnico di Torino and the M.Sc. degree in electrical and computer engineering from the University of Illinois at Chicago, Chicago, IL, in 2012 and 2013, respectively. He is currently a Post-Doctoral Associate with the Department of Electronics and Telecommunications, Politecnico di Torino. His main research interests include the compression of remote sensing images, compressed sensing, and deep learning.

Giulio Coluccia (S'07–M'12) received the B.Sc. and M.Sc. degrees in telecommunications engineering from the Politecnico di Torino, Torino, Italy, in 2003 and 2005, respectively. He received the Ph.D. degree in electronic and communications engineering from the Electronics Department of the Politecnico di Torino, Torino, Italy, in 2009, under the supervision of Prof. G. Taricco.

He is currently a Post-Doctoral Researcher with the Image Processing Laboratory, Politecnico di Torino, leaded by Prof. E. Magli. His research is focused on compressed sensing, with particular interest in its application to image processing and forensics, multidimensional signals, and to distributed source coding and wireless sensor networks. He is involved in the five-year ERC Project "CRISP—Towards Compressive Information Processing Systems" and in the 18-month ERC Proof-of-Concept Project "ToothPic, a largescale camera identification system based on compressed fingerprints," both funded by the European Research Council.

Tiziano Bianchi (S'03–M'05) received the M.Sc. degree (Laurea) in electronic engineering and the Ph.D. degree in information and telecommunication engineering from the University of Florence, Italy, in 2001 and 2005, respectively. Since 2012, he has been with the Department of Electronics and Telecommunications, Politecnico di Torino, as an Assistant Professor. From 2005 to 2012, he was with the Department of Electronics and Telecommunications, University of Florence, as a Research Assistant. He has authored over 100 papers on international journals and conference proceedings. His research interests have involved signal processing in communications and processing of SAR images, multimedia security technologies, signal processing in the encrypted domain, and security aspects of compressed sensing.

Enrico Magli (S'97–M'01–SM'07–F'17) received the M.Sc. and Ph.D. degrees from the Politecnico di Torino, Torino, Italy, in 1997 and 2001, respectively. He is currently an Associate Professor with Politecnico di Torino, Torino, Italy. His research interests include compressive sensing, image and video coding, and vision. He is an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and the *EURASIP Journal on Image and Video Processing*, and a former Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He was an IEEE Distinguished Lecturer from 2015 to 2016. He was a recipient of the IEEE Geoscience and Remote Sensing Society 2011 Transactions Prize Paper Award, the IEEE ICIP 2015 Best Student Paper Award (as senior author), and the 2010 and 2014 Best Associate Editor Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.