Authentication of Swipe Fingerprint Scanners

Vladimir I. Ivanov* and John S. Baras, *Life Fellow, IEEE*

***Abstract*—Swipe fingerprint scanners (sensors) can be distinguished based on their *scanner pattern*—a sufficiently unique, persistent, and unalterable intrinsic characteristic even to scanners of the same technology, manufacturer, and model. We propose a method to extract the scanner pattern from a single image acquired by a widely-used capacitive swipe fingerprint scanner and compare it with a similarly extracted pattern from another image acquired by the same or by another scanner. The method is extremely simple and computationally efficient as it based on moving-average filtering, yet it is very accurate and achieves an equal error rate below 0.1% for 27 swipe fingerprint scanners of exactly the same model. We also show the receiver operating characteristic for different decision thresholds of two modes of the method. The method can enhance the security of a biometric system by detecting an attack on the scanner in which an image containing the fingerprint pattern of the legitimate user and acquired by the authentic fingerprint scanner has been replaced by another image that may still contain the fingerprint pattern of the legitimate user but has been acquired by another, unauthentic fingerprint scanner, i.e., for *scanner authentication*.**

***Index Terms*—Authentication, biometrics, fingerprint, scanner, sensor, pattern, noise.**

EDICS: MMF-DEV, MMF-PROC-IM, BIO-MODA-PAD

I. Introduction

A FINGERPRINT scanner converts the surface or subsurface of the fingertip skin into a digital signal, typically an image [1]. This conversion process can never be made perfect in practice. The imperfections induced by the scanner we classify into: (a) persistent and largely time-invariant imperfections, which we call *scanner pattern*, and (b) imperfections that change rapidly over time, which we call *scanner noise*. Although studies that quantify these imperfections (as non-idealities) of the fingerprint scanners and their suitability for scanner identification are publicly unavailable, studies on the variability at semiconductor level and in digital cameras, FBI's Personal Identity Verification program tests that certify fingerprint scanners, and the interoperability problems among fingerprint scanners suggest that such scanner pattern exists.

We developed a simple, approximate, and tractable model of the relationship among the scanner pattern, the scanner noise, and the fingerprint pattern for one type of capacitive fingerprint scanners together with a method to extract the scanner pattern from a single image and compare it with the pattern similarly extracted from another image. A sufficient similarity between the two patterns indicates that the same scanner acquired both images. We present two modes of operation of the method that use a simple one-dimensional moving-average filter and a correlation coefficient as a similarity score. We applied them in an open-set scenario test on 5,400 images acquired by 27 swipe UPEK fingerprint scanners of exactly the same model. We present the receiver operating characteristics (ROCs) in function of the decision threshold, which is predetermined and independent from the particular scanner under test. The equal error rate (ERR) is $7.46 \cdot 10^{-4}$. The method is computationally efficient, robust, unconditionally stable, and does not require any hardware modifications, so it can be added as software to systems already manufactured and even put into service.

The method we propose can enhance a biometric system by incorporating an additional layer of security that verifies the authenticity also of the scanner connected to it, i.e., for *scanner authentication*, by detecting attacks on it, such as detecting if an image containing the fingerprint pattern of the legitimate user and acquired by the authentic fingerprint scanner has been replaced by another image that may still contain the fingerprint pattern of the legitimate user but has been acquired by another, unauthentic fingerprint scanner. Such verification is increasingly needed because when authenticating to mobile devices (e.g., smartphones and laptops) security problems may arise as this authentication usually takes place in unsupervised environments (e.g., at home). Since a mobile device can be easily stolen, an attacker with physical access to it can launch a powerful attack by manipulating the data acquired and transmitted by the biometric scanner. Furthermore, most biometric information has a low degree of secrecy as it can be captured by an unintended recipient and without user's consent. Since biometric characteristics are difficult to change and cannot be revoked, their compromise may lead to more serious consequences than, for example, compromise of a password. Finally, the widespread use of biometric technologies is set to make the biometric information essentially publicly available, with the face photos being public even today.

II. Related Work

Our observation is that the scanner pattern stems from the variability of element characteristics at the semiconductor level and is caused by imperfections of the conversion from the object applied to a scanner to the scanner output (a digital image). The most closely related research on using such device imperfections is forensics of digital cameras and scanners, which we review next. However, we do not discuss methods for cameras or scanners that classify different models, brands, or manufacturers because such methods look for similarities

Manuscript received on November 16, 2016, revised on March 10, 2017, and accepted on April 19, 2017. This research was supported in part by award MURI W911-NF-0710287 from the United States Army Research Office (ARO).

V. I. Ivanov was with the Institute for Systems Research at the University of Maryland, College Park, MD 20742, USA (e-mail: vladimir3456@gmail.com)

J. S. Baras is with the Department of Electrical and Computer Engineering, and the Institute for Systems Research at the University of Maryland, College Park, MD 20742, USA (e-mail: baras@umd.edu)

This paper includes a supplementary video demonstration of the authentication of swipe fingerprint scanners, available at http://ieeexplore.ieee.org. Please contact V. Ivanov at vladimir3456@gmail.com with questions about it.

* Corresponding author

in the acquisition and/or processing chain of devices of the same model, brand, or manufacturer, and therefore are unable to identify individual devices by problem objective. Not included are also methods which are able to identify cameras or scanners based on their unique characteristics, but which characteristics have not been created as result of manufacturing the device (but were introduced afterwards) or are relatively easy to alter, e.g., dust particles or scratches on the camera lenses or the scanner platen.

A. Digital Cameras and Flatbed Scanners

The established term for the variability of interest in digital cameras is "pattern noise" and is used to denote "any spatial pattern that does not change significantly from frame to frame" [2]. The pattern noise generally has two components: fixed-pattern noise (FPN) and photo-response non-uniformity (PRNU). The FPN, also called dark current noise, is the variation in pixel-to-pixel values when the sensor array is not illuminated; it is due to variations in the detector size and in the doping density, and to impurities. The FPN is additive and independent from the image content. The PRNU is the variation in the pixel responsivity when the sensor array is illuminated. The PRNU is caused by variations in the detector size, spectral response, and coatings' thickness; it is multiplicative and content dependent. Both the FPN and the PRNU are present in both CCD and CMOS image sensors. Besides the pattern noise, the photo image sensors have temporal noise that changes from frame to frame, including photodetector shot noise, pixel reset noise, readout circuit thermal and flicker noises, and quantization noise. An early work [3] estimated the pattern noise in two types of CMOS sensors.

Imperfections in the imaging sensor provide uniqueness that is relatively easy to extract. The pioneering studies [4], [5] used the coordinates of bright pixels caused by the FPN to identify 9 CCD video cameras of 4 models. Their method was applied in [6] to identify CMOS video cameras based on the pixel-to-pixel non-uniformity of both the dark current and the amplifier gain in each sensing element, and in [7] to identify 12 CCD digital still cameras of the same brand by detecting a larger set of sensor imperfections. Its extension [8] to CCD still cameras detected defect patterns due to dark currents and was tested on 3 cameras of the same model and 10 cameras of different models. A hybrid method [9] based on both FPN and PRNU that used a Gaussian smoothing filter was tested on 5 cameras of the same model and also identified 20 CCD modules of the same model. Although promising, these early studies did not provide quantitative assessment of the effectiveness of their methods. Furthermore, although prevalent among cheap cameras, not all cameras have such defective pixels. Many cameras also postprocess the images to remove such defects from them, and such defects may be masked by the image content. Finally, all these methods required the original camera used to acquire the query image or availability of sufficiently many images acquired by this camera in order to determine the unique pattern of the FPN or the defective pixels.

Lukas *et al.* [10], [11] introduced the currently prevalent approach for camera identification. It extracts the high-medium frequency part of the pattern noise, assumed to be a stationary additive white Gaussian signal, by subtracting a version of the image denoised by a wavelet-based algorithm [12] such that the residue is essentially the pixel non-uniformity (caused by the different light sensitivity of the sensor elements), the dominant component of the PRNU. The noise residues extracted from multiple (generally 300 and minimum 50) images were averaged to compute the camera reference pattern. The correlation coefficients between the noise residue of each image and the reference patterns of 9 cameras (2 of the same model) were computed, and the camera corresponding to the maximum coefficient was chosen in a closed-set scenario test. Despite its promising performance, the method is computationally intensive as it requires 2D-wavelet 4-level decomposition and reconstruction, and local variance estimation and Wiener filtering in the transform domain at all 4 levels of decomposition. In another test [13] on the same cameras, the decision thresholds, individual for every camera, determined by using the Neyman-Pearson method at false accept rate (FAR) of 10^{-3} yielded estimated false reject rates (FRR) from $1.14 \cdot 10^{-11}$ to $4.68 \cdot 10^{-3}$, depending on the camera. Using an approximate linearization of the sensor output, the continuation studies [14], [15] of this method estimated each reference pattern by a maximum-likelihood estimator (MLE) from 30 blue-sky or uniformly-lit images from 6 cameras (2 of the same model). A normalized generalized matched filter was applied to image blocks with parameters estimated by a correlation predictor based on a polynomial multivariate least-square estimator using heuristic features from the image, all of which incurred significant computational cost. A major drawback of all studies in this group is that their decision thresholds were camera and even image dependent, thus essentially not open-set scenario tests.

A continuation study [16] using peak-to-correlation energy (PCE) ratio for detection reported a large-scale open-set scenario test on over 1 million images acquired by about 6,900 cameras covering 150 models, with 60 to 200 images per camera. The FRR was $2.38 \cdot 10^{-2}$ and the FAR, estimated by chi-square fitting, $2.4 \cdot 10^{-5}$. The reference pattern was estimated from 50 images by using the MLE of [14], [15] and the same wavelet-based denoising filter. A variant [17] of this method verified if two images were acquired by the same camera in a set of 8 (2 of the same model), with 10 images per camera, by computing the primary-to-secondary peak ratio of their normalized cross correlations. The method of [18], similar to [10] and [14], had the reference pattern estimated from about 300 uniformly-lit images and the PRNU extracted by a high-pass filter equivalent based on a 2D Gaussian filter in spatial domain. With a single test image and 100 images per camera, the average closed-set scenario accuracy was 83.7%; in the open-set scenario, the FAR was 0.1%, the FRR 89%, and the EER 16%. In a different approach [19] for camera model identification, the 592 features of image measures and statistics were reduced to 192 and then classified by support vector machines (SVM). Its accuracy on 16 phone cameras (with 2 cameras of 2 models) and 100 training and 100 testing images per camera was 95.1%.

In the method [20] for identifying flatbed scanners, the averaging of the 2D noise residues, each computed by an anisotropic local polynomial estimator based on directional multiscale optimization, produced the 2D reference pattern, the rows of which and the noise residue were then averaged to compute the linear reference and noise patterns, respectively. A correlation coefficient and SVM using two sets of 8 statistical features were used for classification. In their third set of closed-set scenario tests, the accuracy among 3 and all 4 scanners was 97.6% and 96%, respectively. In a similar study [21], averaging 100 2D noise residues, computed as in [10], produced the array reference pattern, which was averaged by rows to compute the line reference pattern. No error rates were reported, but the identification with the array reference pattern was better than with the line reference pattern. Another contemporary method [22] for flatbed scanners used 3 sets of noise features derived from the noise residue statistics of 5 denoising filters, the 2D wavelet decomposition coefficients, and the absolute neighborhood prediction error in smooth regions. A principal component analysis (PCA) reduced the 60 features to 25, which were classified by SVM, yielding 90.1% accuracy with 13 training and 13 testing images from 7 scanners. Its extension [23] reported 84% accuracy when classifying individual scanners in a set of 14, with 50 training and 50 testing images, although most errors were among scanners of the same model. In an extension [24] of [20] using an additional denoising median and two Wiener filters, the statistics of the linear column pattern and of the row and column correlation vectors produced 204 features, which were then reduced by linear discriminant analysis (LDA) to 10 and finally classified by SVM. The accuracy when tested on 7 flatbed scanners with 200 training and 200 testing subimages was 99.26%.

A further improvement [25] of [10] reduced the scene details in the noise residue by applying small weight factors to large components in 6 weight models after low-pass filtering in the wavelet domain. The reference pattern was estimated from 50 blue-sky images. In open-set tests on 6 cameras, with 200 images per camera, the best FRR was 2.05% at 0 FAR and 1.75% FRR at 0.03% FAR. In an improvement [26] of [15], only regions of the noise residues with pixels having the top 20% highest SNR (w.r.t. the PRNU) were used for detection. A comparative study [27] for the methods in [11], [15], and 2 models of [25] concluded that [11] combined with a mixed correlation coefficient performs best. Each reference pattern, estimated as in [11] from 100 blue-sky images, was correlated with the noise residues first whitened in the frequency domain and yielded best FRR of 0.36% at 0% FAR on 7 cameras with 200 images per camera. Its improvement [28] introduced a correlation over circular cross-correlation norm as a similarity score to suppress the contamination from periodic components and reported best FRR of 0.1% at 0% FAR on the same dataset, outperforming [13], [15]/[16], and [25]. The method in [29] identified regions with similar smoothness and brightness, assigned weights to the different regions, and computed the reference pattern as their weighted average. The tests on 16 cameras showed better ROCs over those in [13] and of model #3 in [25]. At FAR of 0.1%, the lowest FRR was 51.7% with 30 training images and 55.7% with 15 training images, with EER of about 8% in both cases.

One of the alternative methods [30], using a DCT, unsharp and high-pass filters to extract spectral noise, reported best accuracy of 99.06% in a closed-set tests on 8 scanners with 160 training and 160 testing images per scanner. A novel sparse 3D transform-domain collaborative filtering was used for denoising in [31]. Reference [32] proposed an edge adaptive pattern noise predictor based on context adaptive interpolation and adaptive Wiener filter in the spatial domain to remove scene details. A simpler, faster, and easier to implement method [33] using a median filter and anisotropic diffusion algorithm for image denoising achieved EER of 0.5% and FRR of 1.4% at FAR of 0.1% on 69 cameras of 7 models. Another simple method [34], using a spatial 2D adaptive Wiener and two median filters, estimated the reference pattern from over 100 blue-sky images and retained for matching 20% of the pixels with the largest magnitudes, significantly improving the ROC over the common wavelet-based method when tested on 18 cameras. In [35], homomorphic filtering converted the PRNU into additive, extracted it by SVD, and matched with PCE; it was tested on 5 phone camera models with 50 training and 50 testing images. In [36], pixels from positive and negative clusters (pixels with similar noise residue values) were combined into cluster-pairs. Too small a number of mismatched cluster-pairs indicated the image was acquired by the same camera. An SVM generalization [37] for two open-set classifications—camera attribution (when each image is attributed to a specific camera) and device linking (whether two images were acquired by the same camera)—used the correlations from 9 image regions and proposed boundary carving to adjust the SVM decision hyperplane to reduce the false matches from unknown classes; it achieved 97.18% accuracy for the camera attribution and 87.4% for the device linking on large datasets.

B. Fingerprint Scanners

The method for digital cameras of [10] and [13] was applied in [38] to 16 optical and 4 capacitive fingerprint scanners in 3 sets. In the first set, with 2 optical models having 3 scanners each, the accuracy was 99.65% with 1 training image and 100% with 4 images or more, but most errors were among scanners of the same (optical) model. In the third set, with 8 scanners (optical and capacitive), the accuracy was 98% with 64 training images but dropped to about 85% with 1 image. In the second, the most problematic set, with 3 models (2 optical and 1 capacitive) having 2 scanners each, there were many errors even for optical scanners of the same model. The highest accuracy of 95% was with 256 training images but dropped to 45% with 1 image. All this supports our finding that identifying fingerprint scanners, especially of the same model, requires another approach.

A critical drawback of the PRNU-based methods is their poor performance with a single or even a few training images, which, however, is the case in biometric authentications where typically 3 to 5 images become enrolled. Moreover, to achieve high accuracy, some of these methods require homogeneous training images, e.g., uniformly lit or of a blue sky, which is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2702592, IEEE Transactions on Information Forensics and Security

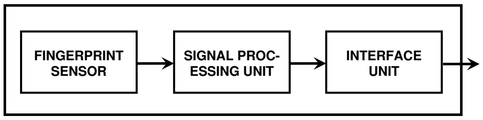


Fig. 1. A block diagram of a fingerprint scanner

inapplicable to our case because the enrolled image(s) contain fingerprint patterns. Furthermore, the mixture of the fingerprint pattern and the scanner pattern in capacitive fingerprint scanners is not simply multiplicative (as in case of PRNU) because the image acquisition in these scanners is very different from that in digital cameras. Finally, an image acquired by swipe scanners manifests a structure that if exploited eliminates the need for 2D signal processing, which is what the PRNU-based methods use.

Our method [39]–[41] to distinguish area fingerprint scanners using a single image, acquired by each scanner, used 2D wavelet decomposition and reconstruction by zeroing the LL-subband coefficients, and a correlation coefficient, computed using the pixels with magnitudes below a threshold. By fitting Gaussian PDFs on the distributions in the worst-case scenario of the open-set tests on 24 area capacitive fingerprint scanners of exactly the same model, we estimated an EER of $2.8 \cdot 10^{-10}$. Although simple, universal, and extremely accurate, the method is computationally intensive (due to the 2D wavelet transform) and uses many (all) pixels of the image.

III. Signal Model and Signal Characteristics

A. Signals and Signal Model

The imperfections induced by the fingerprint scanner we classify into two categories: (a) imperfections that are persistent and largely time-invariant, which we call *scanner pattern*, and (b) imperfections that change rapidly over time, which we call *scanner noise*. The scanner pattern can be a function of many and diverse factors in the hardware and software, e.g., the specific sensing method, the semiconductor technology, the chip layout, the circuit design, and the post-processing. Pinpointing the exact factors or much less quantifying them, however, is difficult because such information is proprietary. Nevertheless, our observation is that the scanner pattern is mainly caused by non-idealities and variability in the fingerprint sensor (see Fig. 1); however, the signal processing unit can also contribute to it. The intrinsic characteristics that determine the scanner pattern, as per our definition, remain relatively unchanged over time. Variations in these characteristics, however, may still exist and may be caused by environmental changes, e.g., changes in temperature, air pressure, air humidity, and sensor surface moisture; material aging; or scratches, liquid permeability, and ESD impact on the surface. The scanner noise is generally caused by non-idealities in the conversion that vary considerably within short periods of time, e.g., thermal noise (inherently present in any electronic circuit) and quantization noise (from the analog-to-digital conversion).

The actual function describing the relationship among the scanner pattern, the scanner noise, and the fingerprint pattern (when present) can be very complex. This function depends on the particular sensing technology, the scanner design, and its implementation, all of which are proprietary and usually publicly unavailable. Furthermore, even if the function can be determined exactly, using it to estimate the scanner pattern may prove analytically intractable or require computationally impractical signal processing. However, this function can be simplified by considering only the major contributing factors and by using approximations; this simple, approximate model we call *signal model*.

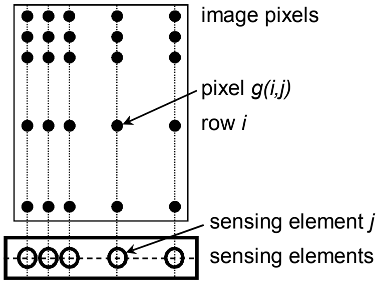


Fig. 2. Image construction and sensing elements of a swipe scanner

Generally, in swipe (a.k.a. slide or sweep) scanners, a line (e.g., a row or a column) of sensing elements performs an instant scan of a tiny strip of the fingertip skin and converts this scan into a line of pixels. As the fingertip is swiped over this line of sensing elements, a sequence of lines of pixels is produced and then assembled (and also possibly enhanced) to construct a two-dimensional fingerprint image (see Fig. 2). The method proposed here we developed for the capacitive swipe scanners of UPEK, Inc., that were formerly known as TouchStrip® scanners (sensors). UPEK was acquired by AuthenTec, Inc., in 2010, then AuthenTec was acquired in 2012 by Apple, Inc., which in 2013 sold the UPEK technology to DigitalPersona, Inc., which in turn merged with Crossmatch, Inc., in 2014. Nevertheless, we refer to them hereinafter as “swipe UPEK scanners” as they are still known in the industry or simply as “swipe scanners” unless noted otherwise. Each such scanner that we used is a TCESC4K module, contains a TCS4K swipe sensor, and is connected (via a development kit) to a computer via a USB. To the capacitive area UPEK scanners of [39], [42] hereinafter we refer as “area scanners.”

Directly applying the methods we developed for area scanners [39]–[43] to swipe scanners is problematic because: (a) constructing a 2D image from the instant scans in swipe scanners is very different from that in area scanners, (b) artifacts, including image enhancement, are possibly introduced in this construction process, and (c) swipe scanners have nearly 3 orders of magnitude fewer number of active sensing elements than area scanners have. However, in swipe scanners: (i) the image pixel values never saturate (i.e., never “clip”), unlike the pixels in area scanners, and hence all sensing elements can be used, and (ii) in a single image, each sensing element produces many (e.g., hundreds) pixel values and thus the scanner pattern of each sensing element gets “incorporated” into many image pixels, thus facilitating its estimation. Based on our analysis [40] of the image acquisition and the images

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2702592, IEEE Transactions on Information Forensics and Security

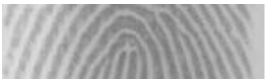


Fig. 3. A segment from a fingerprint image acquired by a swipe scanner

we acquired by both area and swipe UPEK scanners, we hypothesize that the capacitive sensing elements of these two types of scanners are very similar, which leads to the signal model we propose next, despite that we could not find publicly available information to corroborate this. We also observed deviations from what we expected, which suggests that the sensing elements of swipe scanners possibly differ to some extent or that these scanners employ additional processing. Nevertheless, the excellent performance we obtained in both modes of operation (direct and indirect, described later) serves as an indirect validation of the suitability of this signal model.

In swipe UPEK scanners, the line of sensing elements is perpendicular to finger's length and sequentially scans the fingertip while the latter is being swept over the scanner in direction of finger's length. In this way, the consecutive lines of pixels form rows in the image and therefore for any row index i, all pixels in any column j are produced by the same sensing element with index j (see Fig. 2). Based on our signal model we proposed in [40], [42], [43] for area UPEK scanners:

$$g(i,j) = \frac{s(i,j)}{1 + s(i,j)f(i,j)} + n(i,j,t),$$

for swipe UPEK scanners we propose the following model:

$$g(i,j) = \frac{s(j)}{1 + s(j)f(i,j)} + n(i,j,t) = r(i,j) + n(i,j,t) \quad (1)$$

where $g(i,j)$ is the pixel value at row i and column j, $f(i,j)$ is the fingerprint pattern, and $s(j)$ is the scanner pattern of the j-th sensing element in the row because $s(j) = s(i,j)$ for all i since the scanner pattern along columns is the same as it is produced by the same sensing element j. $n(i,j,t)$ is the scanner noise; it is temporal in the row index i because of the same reason. t is time and represents the temporal nature of the scanner noise across different image acquisitions; hence $n(i,j,t)$ becomes $n(i,j)$ when considering a single image. $r(i,j)$ is simply a short notation for the first additive term:

$$r(i,j) \triangleq \frac{s(j)}{1 + s(j)f(i,j)}. \quad (2)$$

B. Signal Characteristics

Each pixel value $g(i,j)$ has 8 bits (although some implementations produce fewer effective bits per pixel) and ranges from 0 to 255 grayscale levels. We assume that the values $g(i,j)$ as saved in a computer file are not further enhanced (or compressed) by image processing as to facilitate the fingerprint authentication or are enhanced (or compressed), but the scanner pattern information in them has not been destroyed or substantially altered. Fig. 3 shows a segment from an image with a fingerprint pattern acquired by such a swipe scanner.

1) Fingerprint pattern $f(i,j)$: A scanner scans the surface of the fingertip skin (a sequence of ridges and valleys) and represents it as a 2D signal. Along with the scanner imperfections, this process may also include nonlinear transformations, e.g., projection of the 3D fingertip onto the 2D scanner platen, and nonlinear sensing and conversion of the ridges and valleys into electrical signals, hence making the fingerprint pattern a nonlinear function of the actual skin surface. For our purposes, we can view $f(i,j)$ in either dimension roughly as one dominant single-frequency oscillation together with its harmonics. This frequency depends on the width of the ridges and valleys, which are specific for each individual. It also depends on the finger type: typically, thumbs have wider ridges and deeper valleys than little fingers, and index fingers have narrower ridges and valleys than thumbs but wider than little fingers. This frequency also depends on the gender (men typically have wider ridges and valleys than women) and on the age (children usually have narrower ridges/valleys). Finally, it may also vary even within the same fingertip. For our purposes, a precise estimate of it is unnecessary, and our study concluded that about 0.63 radians per pixel is sufficiently representative. Furthermore, the spatial scanning resolution of the scanners is sufficiently high as they sample the fingertip skin at the rate of about 10 times faster than a typical $f(i,j)$ along either dimension. Finally, $f(i,j)$ in our model (1) is normalized to (0, 1]. In the regions with absolutely no skin, i.e., away from the fingertip, $f(i,j) = 0$ and $g(i,j) \approx s(j)$.

2) Scanner noise $n(i,j,t)$: We denote with this the combined effect of all factors that result in short-term variations, i.e., from within several seconds to much faster, in the pixel values of consecutively acquired images under exactly the same acquisition conditions and under exactly the same environmental conditions (e.g., without changes in temperature, air humidity, and air pressure). Reproducing the exact same conditions when a fingertip is applied is certainly impossible, and therefore this is only an abstraction to define the scanner noise. Examples for factors in such short-term variations are thermal, shot, flicker, and so forth noises present in electronic circuits, and quantization noise. Other contributing factors may also exist, but identifying them without details about the proprietary scanner implementation is difficult and we have made no effort to do so because the statistical characteristics of the aggregation of all these short-term noises are important for us. However, unlike for area scanners [40], measuring the scanner noise of swipe scanners and quantifying its characteristics, in either space or time, proved to be difficult because swipe scanners acquire images only with fingerprints, not also with air as area scanners do. Moreover, precisely estimating these characteristics proved essentially unnecessary for the herein method development. Assuming that the sensing elements of area and swipe UPEK scanners are very similar to one another, we can assume that $n(i,j)$ is:

a) additive: A plausible and widely-used assumption is that the combined effect of the noises in an electronic circuit can be modeled as an additive noise, as in (1).

b) zero mean: Based on our analysis of the scanner noise in area UPEK scanners, we assume that $n(i,j)$ has zero mean when averaged along many (e.g., over 100) rows i.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2702592, IEEE Transactions on Information Forensics and Security

Hence, by averaging $g(i,j)$ for all rows i, the corresponding average noise $n_{avg}(j)$ at sensing element j is approximately 0, regardless of the distribution of $n(i,j)$ along index i.

c) much smaller than $r(i,j)$: We observed that for swipe UPEK scanners, the pixel values $g(i,j)$ never become 0 and rarely if ever fall below 100 and even 120. In [40], we estimated that the scanner noise of area UPEK scanners has a standard deviation of about 1.336 ($= \sqrt{1.785}$), on average. Assuming a similarly strong scanner noise in swipe UPEK scanners, even 3σ of 1.336 (≈ 4) is much smaller than 100 and therefore also much smaller than $r(i,j)$.

3) Scanner pattern $s(j)$: Unlike for area scanners and similarly to the scanner noise (above), quantifying the characteristics of the scanner pattern of swipe scanners proved difficult and essentially unnecessary. Hereby we summarize the important conclusions about the scanner pattern of area scanners from [40] that gave us important insight and aided the method development for swipe scanners. First, we assume that the scanner pattern $s(j)$ and the fingerprint pattern $f(i,j)$ are independent. Next, as defined in (1), $s(j)$ ranges from 0 to 255 and can be viewed as having two components:

$$s(j) = \mu_s(j) + s_v(j), \text{ where:} \tag{3}$$

- $\mu_s(j)$ is the mean of $s(j)$. It varies gradually in space but may (considerably) change over time in the long term and also under different environmental conditions (e.g., under changes in temperature or moisture) and other factors, which typically results in a relatively constant offset from $\mu_s(j)$ in normal conditions. Our objective is to remove $\mu_s(j)$ because it is not reproducible and cannot serve as a persistent characteristic of the individual scanners;
- $s_v(j)$ is the variable part of $s(j)$. It varies rapidly in space but is relatively invariant in time (both in short term and in long term) and under different environmental conditions. $s_v(j)$ is reproducible and can serve as a persistent characteristic of each scanner. Therefore, our objective is to estimate $s_v(j)$ and use it to authenticate the scanner.

Our analysis [40] of the probability distribution of $s_v(j)$ in area UPEK scanners showed that for practical purposes it can be assumed Gaussian. Deviations from normality, however, do exist, such as outliers in particular and possibly heavy tails. Furthermore, our study on the spatial dependence of the scanner pattern concluded that it exhibits some limited correlation but can be assumed largely uncorrelated, i.e., approximately white. Nevertheless, it is very important that the scanner pattern estimation be robust against significant deviations from this Gaussian assumption and also be able to tolerate a certain degree of correlation in the scanner pattern, as the method we propose here is.

We observed that we can obtain the variable part $s_v(j)$ by subtracting from $s(j)$ its low-pass filtered version $\mathcal{F}\{s(j)\}$:

$$s_v(j) = s(j) - \mathcal{F}\{s(j)\}. \tag{4}$$

$\mathcal{F}\{.\}$ is a (possibly noncausal) moving-average filter, which essentially computes the local (sample) mean, i.e., an estimate of $\mu_s(j)$. Using other filters $\mathcal{F}\{.\}$ or holistically equivalent high-pass filters is also possible, but we chose the moving-average filter due to its simplicity and sufficient filtering here.

IV. Our Method

A. Techniques

The method we propose here is based on two techniques:

1) Averaging along columns: By averaging the pixel values $g(i,j)$ for all rows i (i.e., along each column j) and by using assumptions (a) and (b) in III-B2 for $n(i,j)$, the average $n_{avg}(j)$ of $n(i,j)$ along columns becomes close to 0 and we obtain a close approximation for the average row $g_{avg}(j)$:

$$g_{avg}(j) = \frac{1}{I}\sum_{\forall i} g(i,j) = \frac{1}{I}\sum_{\forall i}\left[r(i,j) + n(i,j)\right] \approx \\ \approx \frac{1}{I}\sum_{\forall i} r(i,j), \text{ where } I \text{ is the number of rows.} \tag{5}$$

We call this case *direct* mode. Note that independence of the scanner noise $n(i,j)$ from $r(i,j)$, and thus from the scanner pattern $s(j)$ and the fingerprint pattern $f(i,j)$, is not required here as long as $n(i,j)$ is additive and zero mean (i.e., assumptions (a) and (b) in III-B2).

Another way of averaging, which we call *inverse* mode, is to first invert the pixel values. By using assumptions (a) and (c) in III-B2 for $n(i,j)$, we neglect the scanner noise $n(i,j)$ in (1), thus assuming that $g(i,j) \approx r(i,j)$, and define the pixel inverse:

$$h(i,j) \triangleq \begin{cases} 1, & \text{if } g(i,j) = 0 \\ \frac{1}{g(i,j)} \approx \frac{1}{s(j)} + f(i,j), & \text{otherwise.} \end{cases} \tag{6}$$

Here again, independence of the scanner noise $n(i,j)$ from $r(i,j)$, and thus from the scanner pattern $s(j)$ and the fingerprint pattern $f(i,j)$, is not required as long as $n(i,j) \ll r(i,j)$ (i.e., assumptions (a) and (c) in III-B2) and can be neglected.

Next, since $s(j)$ is the same along each column j, averaging along columns, i.e., for all rows i, will “amplify” it with respect to $f(i,j)$. Hence, for the average row $h_{avg}(j)$ we have:

$$h_{avg}(j) = \frac{1}{I}\sum_{\forall i} h(i,j) \approx \frac{1}{s(j)} + f_{avg}(j) \tag{7}$$

where $f_{avg}(j)$ is the average fingerprint pattern when averaged along columns. An example for $h_{avg}(j)$ is shown in the upper plot of Fig. 4. Notice the remarkably small local variations of $h_{avg}(j)$ along column indices j despite the fact that the pixel values $g(i,j)$ along nonadjacent columns (i.e., in function of index i) may substantially differ, as visible in the lower plot of Fig. 4. This can be explained by looking at the inverses of the adjacent columns in Fig. 5—even though (the inverses of) the pixel values considerably differ on a pixel-by-pixel basis, their averages along each column are very similar in value. Thus, $h_{avg}(j)$ changes slowly, not rapidly as the sequences of ridges and valleys $g(i,j)$ do in the lower plot of Fig. 4 and the inverses $h(i,j)$ in Fig. 5.

Third, we observed that the mean $\mu_s(j)$ also changes slowly, and since the magnitude of the variable part $s_v(j)$ around this mean is small, the trend of $1/s(j)$ is determined by $1/\mu_s(j)$. Hence, $f_{avg}(j)$ in (7) must also change slowly,

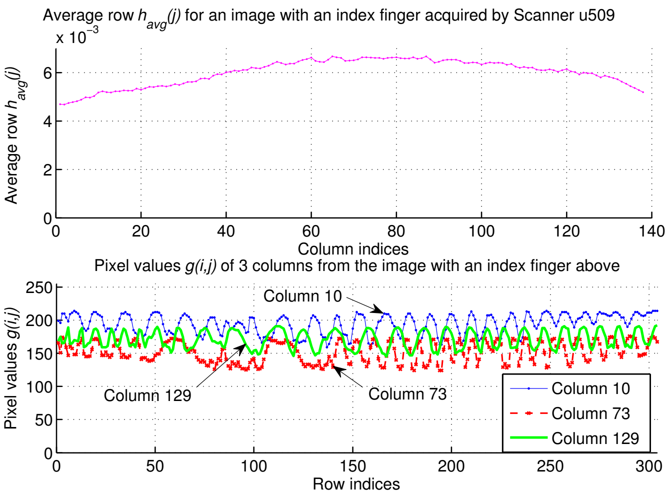


Fig. 4. An average row $h_{avg}(j)$ and 3 columns $g(i,j)$ from the same image

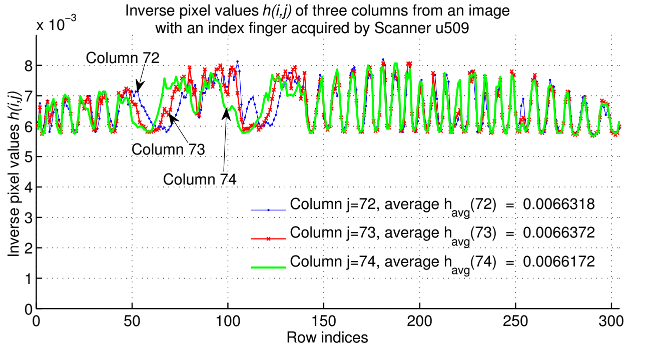


Fig. 5. Inverse pixel values $h(i,j)$ of 3 adjacent columns and their averages

being comparable to the rate of change of $h_{avg}(j)$, i.e., $f_{avg}(j)$ is a slowly varying function in the column index j. This can be explained with the high spatial resolution of the scanner—the scanned fingerprint pattern *on average* does not change significantly from one column to the next one. Finally, since $f_{avg}(j)$ and $s(j)$ (and thus also $1/s(j)$) are independent, the problem of separating them becomes reduced to separating the rapidly changing, noise-like signal $1/s(j)$ and the slowly varying signal $f_{avg}(j)$, independent from it. This can be done similarly to (4): $h_{avg}(j) - \mathcal{F}\{h_{avg}(j)\}$. Two exemplary $\mathcal{F}\{.\}$ are the moving-average filter we present here and the adaptive Wiener filter described in [40], [44].

Fig. 6 shows an example for the average row $g_{avg}(j)$ and its corresponding (i.e., computed from the same image) $h_{avg}(j)$.

2) Linear approximations: Our analysis discovered that $g_{avg}(j)$ approximately contains the variable part of $s(j)$ and also in a form that is relatively easy to extract. This can be seen from the series of approximations we present next. First, we noticed that for swipe UPEK scanners, $r(i,j)$ is approximately a linear function of the fingerprint pattern $f(i,j)$; this approximation and its accuracy we detail in Appendix A. In summary, in (18) we conclude that:

$$r(i,j) \approx b(j)\left\{1 - b(j)\left[f(i,j) - a\right]\right\}, \tag{8}$$

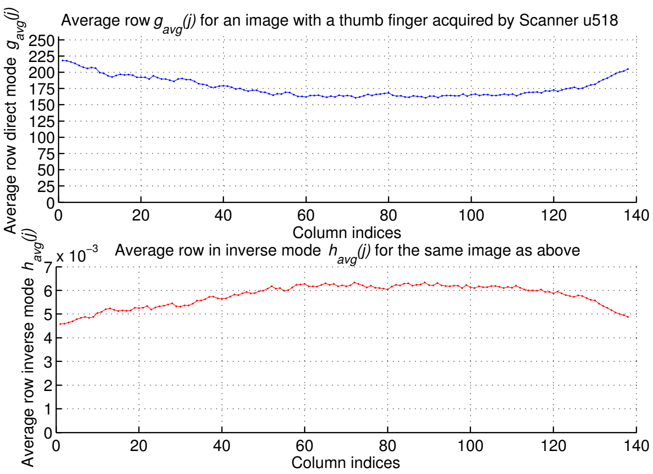


Fig. 6. Average rows in direct and inverse mode

$$\text{where } b(j) = \frac{s(j)}{1 + s(j)a} \text{ and } a = 0.0025. \tag{9}$$

With this a and the scanner pattern values $s(j)$ in the range from about 100 to 255, $b(j)$ is between about 80 and 156.

By substituting approximation (8) in (5) and averaging $r(i,j)$ along each column j, the average row $g_{avg}(j)$ becomes:

$$g_{avg}(j) \approx \frac{1}{I}\sum_{\forall i} r(i,j) = b(j)\left\{1 - b(j)\left[f_{avg}(j) - a\right]\right\} \tag{10}$$

because $b(j)$ does not depend on the row index i since it is a function only of the scanner pattern $s(j)$. As the number of rows is typically large (well over 100), $f_{avg}(j)$ is close to the population mean of $f(i,j)$ along i, observed to be about 0.0025, which is also the constant a in (9). All this makes $[f_{avg}(j) - a]$ approximately constant and in the order of 10^{-4} in magnitude. This, together with the approximate range of $b(j)$ (above), makes the second additive term $b(j)\left[f_{avg}(j) - a\right]$ in (10) in the order of 10^{-2} and thus dwarfed by 1, which is the first additive term in (10). Therefore, $g_{avg}(j)$ is approximately equal to $b(j)$.

Finally, our analysis in Appendix B shows that for swipe scanners with their signal ranges and with the filters we propose (described next), the approximation $b(j) \approx const{\cdot}s(j)$ with $const$ varying within $\pm 3\%$ is sufficiently accurate. This implies that $b(j)$, and therefore also $g_{avg}(j)$, is essentially the scaled scanner pattern $s(j)$. Hence, a filter can remove the mean of $s(j)$ and thus obtain only its variable part, as we wanted. That is, similarly to (4) and $h_{avg}(j)$ in inverse mode, here in direct mode we can also compute $g_{avg}(j) - \mathcal{F}\{g_{avg}(j)\}$ and achieve our goal. Again similarly, two exemplary $\mathcal{F}\{.\}$ are the moving-average filter we present here and the adaptive Wiener filter described in [40], [44].

B. Modules

Fig. 7 shows the conceptual diagram in which the acquired image $\mathbf{g}$ is processed to produce the scanner verification decision d, together with the main interface signals between the modules. A module can process its signals in several

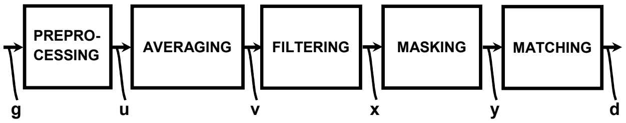


Fig. 7. Signal processing modules and conceptual diagram of operation

ways, which we call *modes of operation*. The modules in Fig. 7 and their modes are a subset of the those in [40], [44] and are presented here in order to illustrate the concept and demonstrate exemplary performance.

Some swipe scanners produce images containing also rows and/or columns of pixels with constant values (like padding) around the actually acquired image, which need to be cropped out from it. Let I and J be the number of rows and columns, respectively, of this cropped image $\mathbf{g}$ with pixels $g(i, j)$.

1) Preprocessing Module: It takes $\mathbf{g}$ as input and produces $\mathbf{u}$, a 2D signal with the same size as $\mathbf{g}$. It has two modes: direct $u(i, j) = g(i, j)$ and inverse $u(i, j) = h(i, j)$.

2) Averaging Module: It computes the average row $\mathbf{v}$ of the pixels along columns (for swipe UPEK scanners) of $\mathbf{u}$, i.e., the average of pixels produced by the same sensor element j:

$$v\left(j\right) = \frac{1}{I}\sum_{i=1}^{I} u\left(i, j\right), \quad \text{for } j = 1..J. \tag{11}$$

In case of computational or time constraints, only some of the I rows can be averaged, e.g., about 100 rows are sufficient.

Some swipe scanners may employ more than one line of sensing elements, in which case the construction of the 2D fingerprint image from the sequence of lines of pixels, acquired by the sensing elements, involves signal processing which is manufacturer proprietary and usually publicly unavailable.

3) Filtering Module: It filters the average row $\mathbf{v}$ to produce $\mathbf{x}$, which contains the scanner pattern:

$$\mathbf{x} = \mathbf{v} - \mathcal{F}\{\mathbf{v}\}. \tag{12}$$

This module essentially removes the fingerprint pattern and the (variable) mean $\mu_s(j)$ of the scanner pattern, yielding only its variable part $s_v(j)$. Depending on $\mathcal{F}\{.\}$, the Filtering Module can operate in several modes. Here we present only the moving-average filter mode; another mode that uses an adaptive Wiener filter is described in [40], [44].

Generally, for a pixel k sufficiently far from the ends of $\mathbf{v}$ so that $(k + j)$ does not address elements outside of it, the output (i.e., local mean) $\mathbf{v}^{(\mathbf{lm})}$ of a moving-average $\mathcal{F}\{.\}$ is:

$$v^{(lm)}\left(k\right) = \mathcal{F}\{v(k)\} = \frac{1}{M}\sum_{j=-\left\lfloor \frac{M}{2}\right\rfloor}^{\left\lfloor \frac{M-1}{2}\right\rfloor} v\left(k + j\right) \tag{13}$$

where M is the size of the moving-average window. It is preferable that M is odd so that the window is symmetric, but choosing M to be even is also possible. We obtained optimal results with $M = 3$, but good overall performance can also be achieved with M from 2 to about 7.

Because of the finite length of $\mathbf{v}$, processing the signal discontinuities at the beginning and the end may lead to undesired artifacts. To avoid this, we shorten the computation when the averaging window goes outside of $\mathbf{v}$, which method we recommend. It is also possible to pad with replicas of the pixels located near each end or with a suitably chosen constant. Incorporating such methods to mitigate edge effects may seem trivial, but actually it is quite important because the length of $\mathbf{v}$ is relatively small and these artifacts may impair the scanner pattern estimate for about 10 pixels, which is not negligible and may decrease the performance. Furthermore, since tightly applying a fingertip in the regions close to the two ends of a swipe scanner is difficult, the pixels in these regions typically contain little to no fingerprint pattern; hence, the estimate of the scanner pattern there can be made very precise if such unwanted artifacts are mitigated or altogether avoided.

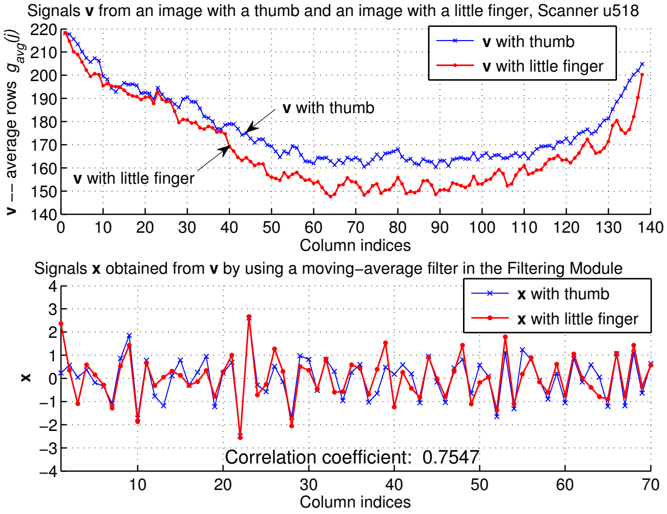


Fig. 8. Input & output signals of the moving-average filtering in direct mode

Another aspect of the processing in this module is applying a windowing function in (13). By weighing the pixel values $v(k + j)$ differently, the pixels away from the current index receive smaller weights and their effect on the filtering is diminished, which may increase the accuracy. For simplicity, here we do not use windowing.

Finally, the output $\mathbf{x}$ of this module is:

$$x\left(k\right) = v\left(k\right) - v^{(lm)}\left(k\right), \quad \text{for } k = 1..J. \tag{14}$$

Fig. 8 shows the inputs $\mathbf{v}$ in direct mode and the outputs $\mathbf{x}$ of the Filtering Module using a moving-average filter for two images, one containing a thumb and another one a little finger, acquired by the same scanner, and the correlation coefficient between them. The lower plot shows only the first half of the columns for better visibility. For comparison, Fig. 9 shows the signals $\mathbf{v}$ and $\mathbf{x}$ for the same images as in Fig. 8 and using the same moving-average filter but processed in inverse mode, and the correlation coefficient between them. In contrast, Fig. 10 shows the inputs $\mathbf{v}$ in direct mode and the outputs $\mathbf{x}$ for two images containing the same thumb finger (as in Fig. 8) but acquired by two different scanners, and their correlation coefficient. Again, the lower plot shows only the first half of the columns for better visibility.

4) Masking Module: This module is optional, but using it can considerably improve the robustness and the performance

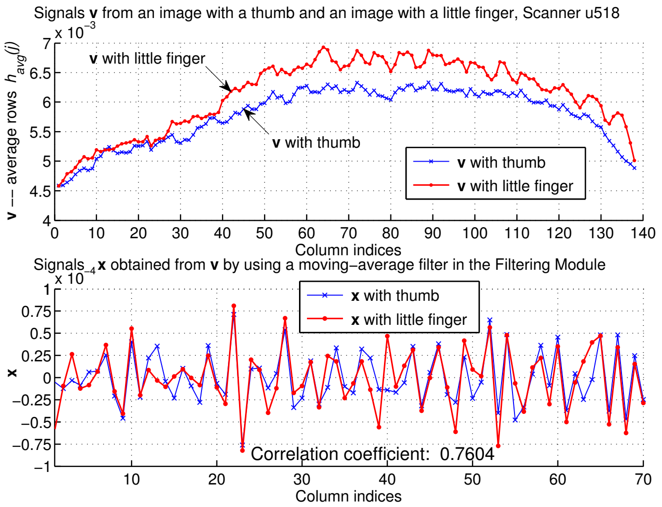


Fig. 9. Input & output signals of the moving-average filtering in inverse mode

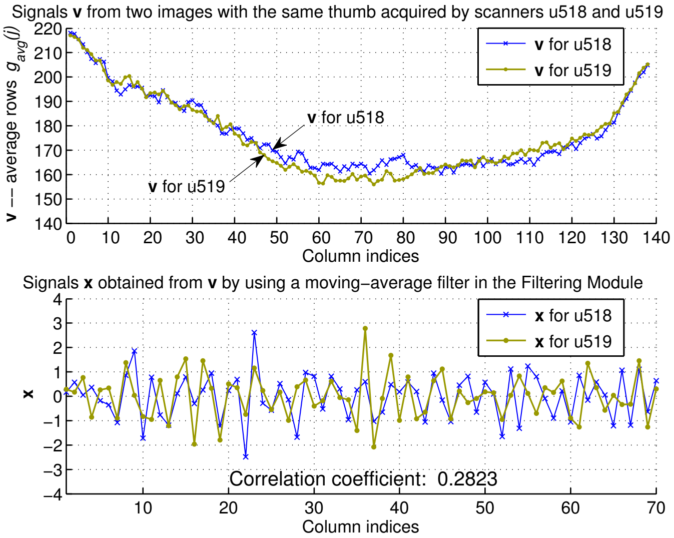


Fig. 10. Input & output signals of the moving-average filtering in direct mode

in boundary cases. Its output $\mathbf{y}$ is a mask indicating whether or not each pixel in $\mathbf{x}$ will be further used and processed:

$$y(j) = \begin{cases} 1, & \text{if } |x(j)| \leq \theta \text{ — pixel } j \text{ to be used} \\ 0, & \text{otherwise — pixel } j \text{ not to be used.} \end{cases} \qquad (15)$$

The threshold θ is chosen as to exclude from further processing two groups of pixels. The first group consists of the pixels close to the beginning and to the end of $\mathbf{x}$ as they may be unacceptably distorted as result of processing the discontinuities of the finite-length signal (described earlier). Although the techniques we proposed significantly mitigate such artifacts, sometimes this is insufficient and leads to a very inaccurate matching score; therefore, such pixels must be excluded. Twice the length of the filter used in the Filtering Module may serve as a loose upper bound for the total number of pixels that this group may contain. The second group of pixels includes pixels with excessively large magnitudes because we observed that they provide very imprecise estimates of the scanner pattern (due to various and unexplainable reasons). The number of pixels in this group is typically small (about several). Hence, if the combined number of pixels of the two groups becomes too large (i.e., in the order of tens), we recommend that the case be carefully analyzed and possibly the threshold θ increased.

The optimal value of θ should be determined by experimentation and tests. When this is infeasible, as a very approximate guideline, θ in direct mode can be chosen from about 2.5 to about 5 (we obtained very good results with $\theta = 3.5$), and in inverse mode from about $1 \cdot 10^{-4}$ to about $3 \cdot 10^{-4}$ (we obtained very good results with $\theta = 2.5 \cdot 10^{-4}$).

5) Matching Module: It computes a similarity score between the scanner patterns extracted from two images and produces a decision as to whether or not they are sufficiently similar. Let $\mathbf{x_e}$ and $\mathbf{x_q}$ be the outputs of the Filtering Module when the input is, respectively, an image $\mathbf{g_e}$ acquired for the scanner enrolment and an image $\mathbf{g_q}$ acquired for the scanner verification. The corresponding outputs of the Masking Module are $\mathbf{y_e}$ and $\mathbf{y_q}$. The Matching Module (a) selects the common pixels marked as to be used in $\mathbf{y_e}$ and $\mathbf{y_q}$, (b) quantifies in a score the similarity between $\mathbf{x_e}$ and $\mathbf{x_q}$ for these common pixels, and (c) compares this score with a threshold and produces a binary decision d as to whether or not $\mathbf{g_e}$ and $\mathbf{g_q}$ have been acquired by the same scanner.

Quantifying the similarity between $\mathbf{x_e}$ and $\mathbf{x_q}$ can be done in several ways (modes) (see [40], [44]): using a correlation coefficient (preferred), a normalized correlation, or a relative mean-square error. Correlation, i.e., matched filtering, is a natural choice as it is the conventional method for detecting digital watermarks and has already been used for identifying digital cameras [13]. Being a robust and simple method, correlation also requires little computational power, which is very important for the intended applications of this authentication.

Let D be the set of all indices k for which both $y_e(k) = 1$ and $y_q(k) = 1$, i.e., the pixels marked as to be used in both $\mathbf{x_e}$ and $\mathbf{x_q}$, and N_D be the number of elements in D. In case of a correlation coefficient, the similarity score z is:

$$z = \frac{(\mathbf{x'_e} - \overline{\mathbf{x'_e}}) \cdot (\mathbf{x'_q} - \overline{\mathbf{x'_q}})}{\|\mathbf{x'_e} - \overline{\mathbf{x'_e}}\| \, \|\mathbf{x'_q} - \overline{\mathbf{x'_q}}\|} \qquad (16)$$

where $\mathbf{x'_e}$ and $\mathbf{x'_q}$ are vectors containing only those elements of $\mathbf{x_e}$ and $\mathbf{x_q}$, respectively, whose indices are in D. $\overline{\mathbf{x'_e}}$ and $\overline{\mathbf{x'_q}}$ are the means of the elements of $\mathbf{x'_e}$ and $\mathbf{x'_q}$, respectively. If any of the denominator terms is 0, then $z = 0$. The decision d is 1, i.e., scanner match, if $z \geq \tau$, where τ is a fixed decision threshold predetermined as result of optimization and from the required trade-off between the FAR and the FRR. Otherwise, the decision d is 0, i.e., scanner nonmatch. It is important to note that our method is a solution to the harder, open-set problem because the decision threshold τ, once optimized, is fixed and is used to binary classify the query image as being acquired or not by the authentic scanner without comparing it with any other scanner image. In contrast, solutions to closed-set problems only point to the scanner that most likely acquired the query image in a fixed group of scanners.

If N_D is less than a predetermined number, the pixels in common are too few to produce a reliable similarity score and thus to make a decision; therefore, a new image needs to be acquired. This predetermined number generally depends on the number of sensing elements and is established experimentally. For swipe UPEK scanners, which have about 140 sensing elements that effectively can be used for our purposes, we recommend this number to be about 50.

To improve the score robustness and the overall accuracy, we recommend that several scores between $\mathbf{x_e}$ and $\mathbf{x_q}$ be computed and then combined into a single combined score that is compared with τ. For example, $\mathbf{x_e}$ and $\mathbf{x_q}$ (and $\mathbf{y_e}$ and $\mathbf{y_q}$, respectively) can be split into two to give two corresponding scores: one score z' computed between the first halves of $\mathbf{x_e}$ and $\mathbf{x_q}$, and another score z'' between their second halves. Combining the two scores can be done by using several types of means. Using a quadratic mean (i.e., a root mean square) provides very good results; the combined score in this case is:

$$z_{combined} = \sqrt{\frac{(z')^2 + (z'')^2}{2}}. \tag{17}$$

We observed that for swipe UPEK scanners, the scores z' and z'' computed for two images acquired by one and the same scanner in some cases and for some scanners can substantially differ: one of the scores can be much larger than 0.5, whereas the other one can be much smaller and even close to 0. The cause for this is unclear. Combining them with the quadratic mean, however, ensures that the combined score is sufficiently large as to produce a scanner match decision, thus reducing the FRR. Moreover, we observed that the quadratic mean also reduces the FAR. Finally, it is also possible to use uneven halves or to split the signals into more than two parts, although using many parts is discouraged because the individual scores may become unreliable as the number of pixels for their computation becomes too small.

To improve the fingerprint authentication, several images are typically used to enroll a fingerprint pattern. Similarly, (these) several images can also be used (as in [40], [44]) to increase the accuracy of the scanner authentication. In one method of using multiple images $\mathbf{x_e}$ acquired during the scanner enrolment, the Matching Module performs (a) and (b) in Section IV-B5 for each pair of one enrolled image and the query image, and then averages all scores before performing (c). Similarly, this can also be applied when several query images $\mathbf{x_q}$ are acquired. In another method, an "average" enrolled scanner pattern $\mathbf{x_{e,a}}$ is computed by pixel-wise averaging all enrolled $\mathbf{x_e}$, after which the average $\mathbf{x_{e,a}}$ is used as a single enrolled $\mathbf{x_e}$ for matching. This method, however, may be suboptimal in certain cases because thus-computed $\mathbf{x_{e,a}}$ may be considerably distorted for certain pixels and hence cause decision errors.

C. Performance

Besides the modes we proposed above, the modules can also operate in other modes, as described in [40], [44], delivering a varying overall performance in a trade-off with the required computational power. One well-performing combination of modes is when the Preprocessing Module operates in direct mode, the Filtering Module uses a moving-average filter with $M = 3$ and shortens the local mean computation at the edges, the Masking Module uses threshold $\theta = 3.5$, and the Matching Module computes the correlation coefficients between the two even halves of $\mathbf{x_e}$ and $\mathbf{x_q}$, and combines them in a quadratic mean to be the final similarity score. Another well-performing combination is when the Preprocessing Module operates in inverse mode, the threshold θ is $2.5 \cdot 10^{-4}$, and the other modules operate as they do in direct mode.

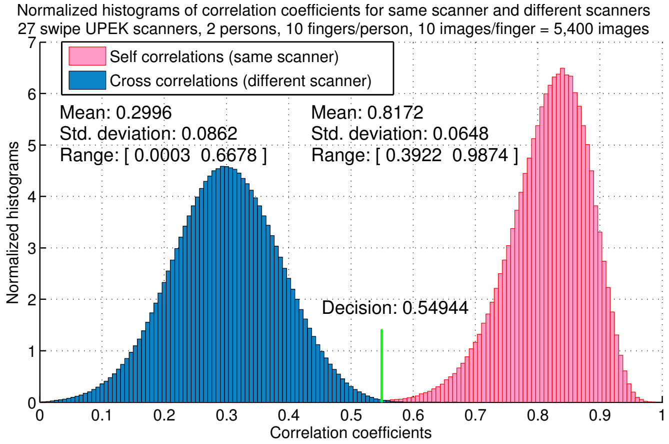


Fig. 11. Histograms of self correlations and cross correlations in direct mode

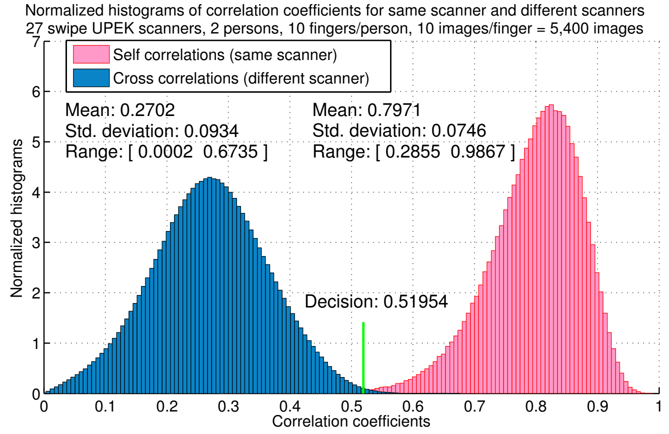


Fig. 12. Histograms of self correlations and cross correlations in inverse mode

The normalized histograms (integrating to 1) of the correlation coefficients in direct mode are shown in Fig. 11 and in inverse mode in Fig. 12. We tested both modes in an open-set scenario on 5,400 images acquired in the course of about 1.5 years at room temperature by 27 swipe UPEK scanners for all 10 fingers of 2 persons (a male and a female) with 10 images per finger. In either mode, only a single image $\mathbf{g_e}$ was used for the scanner enrolment and only a single image $\mathbf{g_q}$ for the scanner verification. Also, in either mode, every image was matched against all other images, with the total number of matchings being about 15 million. Only one of the matchings, AB or BA, was computed and recorded since all processing is completely symmetric for $\mathbf{g_e}$ and $\mathbf{g_q}$. There were no failures to enroll or to verify in either mode (direct or inverse).

The corresponding empirical ROCs when varying the decision thresholds in Fig. 11 and Fig. 12 are shown in Fig. 13. On the y-axis is shown the true accept rate (TAR), i.e., when an authentic image is correctly identified as authentic, which is equal to (1 − FRR). Both axes are shown on a

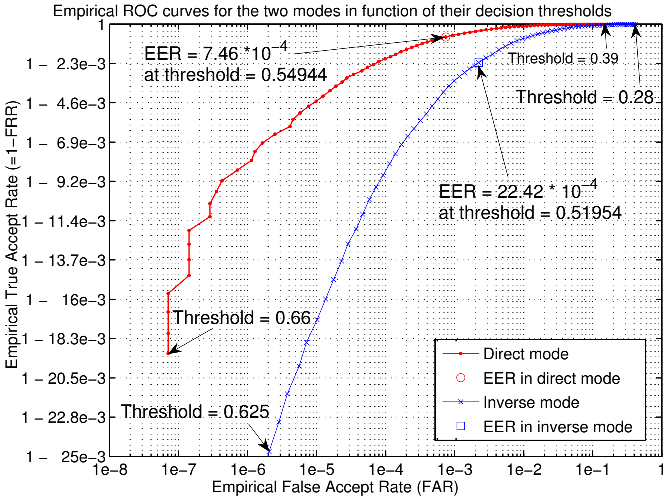


Fig. 13. Receiver operating characteristics in direct and inverse mode

logarithmic scale, but because the logarithmic function when being very close to 1 is almost linear, the y-axis marks, evenly spaced with the exponential step of -0.001, appear to be linearly spaced. The y-axis labels as shown correspond to $10^{-0.011}, 10^{-0.010}, 10^{-0.009}$, and so forth through 1, but by writing them as subtractions, the FRR becomes clearly visible since it is the complement of TAR to 1. In direct mode, the empirical EER (i.e., when the empirical FAR and FRR are equal) is $7.46 \cdot 10^{-4}$ and is achieved at decision threshold 0.54944. In inverse mode, the empirical EER is $22.42 \cdot 10^{-4}$, which is 3 times larger than the EER in direct mode, and is achieved at decision threshold 0.51954. By increasing the decision threshold, the FAR decreases, whereas the FRR increases. As visible in Fig. 13, for the same FAR, the FRR in inverse mode is considerably worse than the FRR in direct mode—the FRR in inverse mode is about 3.5 times larger than the FRR in direct mode at FAR of 10^{-6}, roughly 4 times larger at FAR of 10^{-5} and of 10^{-4}, and about 5.5 times larger at FAR of 10^{-3}. Although the parameters of either mode have not been fully optimized, we recommend using the direct mode when using a moving-average filter because it is simpler and clearly provides much better performance than the inverse mode. We attribute this to the inversion, which is nonlinear and leads to distortion of the scanner pattern since the scanner noise is not averaged out directly as in direct mode but comes in the denominator after the inversion (compare (5) with (6) and (7)). This widens both the cross-correlation distribution and especially the self-correlation distribution.

D. Features

Accuracy: To our best knowledge, our method is the first and the only one providing an EER below 10^{-3} on a large number of fingerprint scanners of the same acquisition technology, manufacturer, and exact same model. Furthermore, it requires only a single image for scanner enrolment and a single image for scanner verification. The performance of the combination of modules and modes presented here is just an example for its potential, not its upper bound. A larger set of modes and tools to optimize the performance and implement the modules on a particular target is provided in [40], [44].

Computational efficiency and speed: Since the main application of the method is envisioned to be in mobile devices, which are constrained in both computational power and energy, the computational efficiency is very important. Since the scanner authentication is only part of the user authentication (the fingerprint authentication is another one), it should add very little extra time. High speed of verification is extremely important if the method is also to be used for scanner identification. Our executable floating-point implementation of the method in direct mode on a low-performance Intel T2300 processor (32 bit, 1.66 GHz, dual core) takes less time than 10 ms per scanner verification.

Implementation and scalability: Extreme simplicity was also our central objective, which we achieved by using only one-dimensional signal processing and avoiding any transforms. This also results in linear dependence between the computations needed and the number of pixels used. Furthermore, the method can be implemented entirely in software, entirely in hardware, or partially in either one. For example, the moving-average filter simply scales by $1/M$ the adjacent pixels in a window; in a fixed-point implementation, this can be further speed-optimized by choosing M to be a power of 2 so that the division becomes "shift right" as a microprocessor instruction or a hardware operation. An FFT accelerator can compute the correlation. Finally, the different modes allow granularity with varying degrees of complexity depending on the constraints, and the accuracy increases when more pixels are used.

Robustness, stability, and fixed-point implementation: In addition to working with images with patterns of two different fingers, both as patterns and as finger types, the method works properly even when not a fingertip but another body part is applied to the scanner (e.g., a palm) because it inherently does not rely on specific characteristics of the fingerprint pattern. Furthermore, the module parameters can vary in wide ranges. All modules and modes are also unconditionally stable as there are no feedback loops in any form and at any level. Since floating-point coprocessors are typically absent in mobile phones, being implementable in fixed-point arithmetic is another important advantage because it tolerates round-off errors due to finite-length effects in the parameter, coefficient, and signal quantization. The image depth is limited to 8 bits/pixel, which simplifies the scaling at the interfaces between the consecutive processing stages both in a microprocessor and in dedicated computational hardware. The processing revolves around computing moving-average sums, which cannot create overshoots in the intermediate signals, and around scaling by bounded numbers. Computing the correlation involves multiplication and accumulation of two signals and can operate with the running indices so that the running sum never exceeds the dynamic range of the finite-precision arithmetic unit. The rest are indexing operations and comparisons.

Our method also does not employ transforms across domains (e.g., a wavelet transform), which are typically susceptible to numerical problems due to the finite word length. The inverse mode, however, may require care when implemented in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2702592, IEEE Transactions on Information Forensics and Security

fixed point because of potential round-off errors. Finally, our method has an edge even over methods that require floating point computations and use software libraries for this because they will probably be more time and/or energy consuming.

Cost and deployment in existing systems: The method was designed for and tested on a mainstream, general-purpose, commercial off-the-shelf, low-cost fingerprint scanners as mobile devices are most likely to be equipped only with such. Since it does not require any changes in the scanner, it can be added to systems already manufactured and put into service by upgrading their software or programmable hardware.

Other: Besides providing higher authentication accuracy than the inverse mode, the direct mode both simplifies the computations and decreases their number. It also makes possible implementing the method in limited-precision fixed-point arithmetic because it greatly reduces the dynamic range of the processed signals.

Since only a single image is needed for scanner enrolment, the method can perform automatic scanner re-enrolment: after a successful scanner verification, the query scanner pattern can be stored as a newly enrolled pattern and thus adapt the system to long-term variations of the scanner pattern.

Finally, we have also implemented the method in software that processes live images, acquired by the scanners, and successfully demonstrated it [45] at the Biometric Consortium Conference and Technology Expo (now Federal Identity Forum & Homeland Security) in Tampa, Florida, in September 2011 and 2012.

V. Applications

A scanner authentication consists of a scanner enrolment and a scanner verification. In [39], [42], we introduced the term *bipartite authentication* to denote the two-part authentication: a biometric authentication with a scanner authentication, a combination that verifies the authenticity of both the user and the fingerprint scanner. A bipartite authentication consists of a *bipartite enrolment*, which enrolls both the biometric information of the legitimate user and the scanner pattern of the legitimate scanner from the same image(s), and a *bipartite verification* (see Fig. 14), which verifies both the biometric information and the scanner pattern contained in the query image, and allows access only if both verifications succeed. Thus, the scanner authentication implements an additional layer of security that verifies the authenticity of the acquisition device of the fingerprint image. As the bipartite authentication verifies both who the user is (their fingerprint) and what the user has (their scanner), it binds the user and the device.

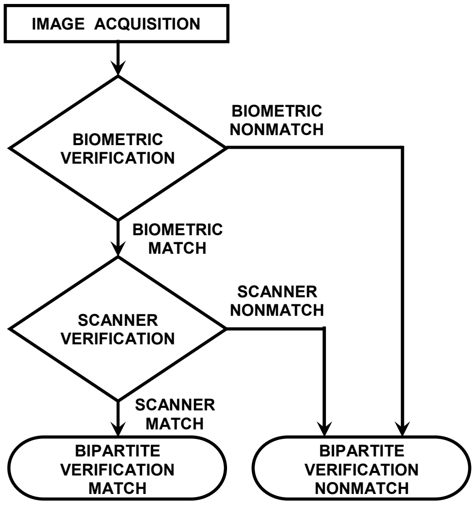


Fig. 14. Bipartite verification

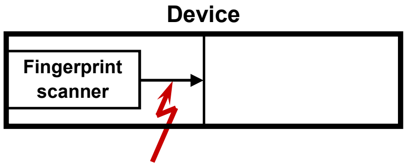


Fig. 15. An attack in a device

Therefore, the proposed method can be used to authenticate a scanner and thus detect attacks on it [1], e.g., to detect a malicious replacement of the authentic scanner or a replay of a stolen image of the authentic fingerprint at the input of the fingerprint authentication subsystem (which can be hardware and/or software), see Fig. 15. This type of attack is becoming increasingly feasible in mobile devices (e.g., smartphones equipped with fingerprint scanners) because they can be easily stolen, giving to an attacker physical access to them and thus the ability to launch so powerful an attack. This is a growing security threat as the biometric information has a low degree of secrecy and the widespread use of biometric technologies makes it essentially publicly available [46]. In particular, an attacker may possess stolen partial or even complete fingerprint information of the legitimate user, including digital images acquired by an unauthentic fingerprint scanner (i.e., another scanner) or information about user's fingertip obtained from a latent (i.e., left on a surface) fingerprint. The attacker may also create a digital forgery of the authentic fingerprint or digitally synthesize an image from the fingerprint features (e.g., minutiae) of the legitimate user. Furthermore, the attacker may also acquire an image by applying to the scanner a physical forgery of a real fingertip, e.g., a gummy finger made of Play-Doh or a foil with printed fingerprint pattern on it, as described in [46]. In all these cases, and as also demonstrated in [47], [48], such attacks on a system employing merely a fingerprint authentication may succeed. Another attack will arise if (or better say, when) online services start performing fingerprint (and generally, biometric) authentication on the cloud because this will open plenty of vulnerabilities both in the transit and in the storage of the images. When added, however, the scanner authentication can detect all of these attacks provided that the scanner which acquired the fake image is different from the authentic scanner; digitally synthesized images are even easier to detect as they don't contain a scanner pattern at all. Finally, phones with fingerprint scanners have become readily available: from Fujistu with its REGZA and ARROWS, Apple since iPhone 5s, and Samsung since Galaxy S5, to name a few, and it was shown that phone's fingerprint authentication can be easily spoofed [47], [48].

Another application example is a contextual authentication or enforcing different user rights depending on the access device. For example, a bank service that over the Internet receives fingerprint images from user's home computer and user's phone for increased security may limit the permitted operations only to low-privilege ones (like checking account balances) when the user authenticates from her phone because it is easier to steal or compromise; the differentiation is made based on the different scanner patterns of the fingerprint scanners present in her computer and in her phone. Although not a malicious or physical replacement, this is a type of scanner replacement. Other possible areas of application include mobile wallets, access to health care and medical records, and asset management. The scanner pattern can also provide source of randomness (e.g., for cryptographic purposes) and be used for device identification.

APPENDIX A
LINEAR APPROXIMATION FOR $r(i,j)$

Hereby we derive a linear approximation for $r(i,j)$ in function of $f(i,j)$. To simplify the notation, we omit the indices and work with a single pixel with indices (i,j). The standard tangent-line approximation for the function $l(f)$ at a given point a is:

$$l(f) \overset{\triangle}{=} r = \frac{s}{1+s.f} \approx k(f-a)+b, \quad \text{where}$$

$$k = l'(a) = \left(\frac{s}{1+s.f}\right)'\Bigg|_{f=a} = -\frac{s^2}{(1+s.a)^2} = -b^2$$

$$\text{because } b = \frac{s}{1+s.a}. \text{ Therefore, } l(f) = -b^2(f-a)+b.$$

The accuracy of this approximation largely depends on the selection of a. With $g(i,j)$ being over 100 and $r(i,j)$ about the same, the range of the fingerprint pattern f is very small. A possible explanation for this is that since the fingertip is swiped, pressing it hard enough to produce a sufficiently large f, much less to saturate the sensing elements, very seldom occurs. By using the typical scanner pattern s of 200, we have:

$$f = \frac{1}{r} - \frac{1}{s} \approx \frac{1}{100} - \frac{1}{200} = 0.005.$$

Hence, we conclude that f varies from 0 (no fingerprint) to about 0.005. The best overall approximation is when a is in the middle, i.e., $a = 0.0025$. Fig. 16 shows $l(f)$ and its linear approximation with $a = 0.0025$ for 3 values of s and the corresponding relative errors. Even in the worst case, i.e., when $s = 255$, the relative error is at most 15.2% and this occurs only near the ends of the range of f. Finally, with the indices:

$$r(i,j) \approx b(j)\left\{1 - b(j)\left[f(i,j) - a\right]\right\}, \tag{18}$$

$$\text{where } b(j) = \frac{s(j)}{1+s(j)a} \text{ and } a = 0.0025. \tag{19}$$

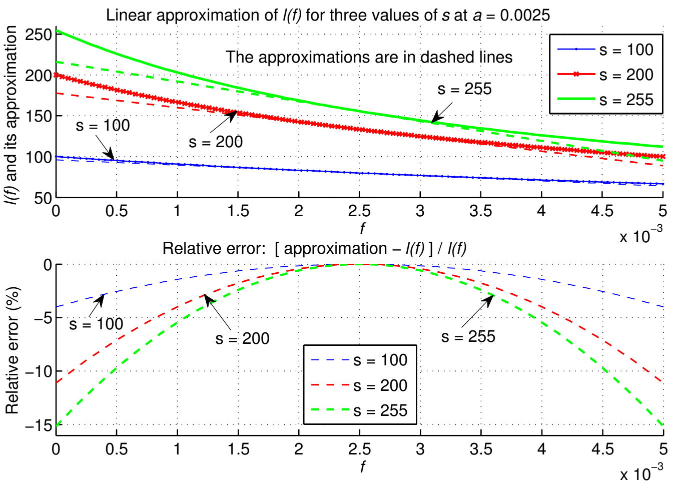


Fig. 16. A linear approximation for $r(i,j)$ and its accuracy in function of s

TABLE I
LINEAR APPROXIMATION FOR $b(j)$ AND IT ACCURACY FOR DIFFERENT μ_s

Factors and parameters	Min	Typical	Max
μ_s $\sigma_s = 5, \quad a = 0.0025 = 1/400$	**100**	**200**	**255**
$1+\mu_s a$	1.2500	1.5000	1.6375
$1+(\mu_s - 3\sigma_s)a$	1.2125	1.4625	1.6000
$1+(\mu_s + 3\sigma_s)a$	1.2875	1.5375	1.6750
$1/\left(1+\mu_s a\right)$	0.8000	0.6667	0.6107
$1/\left[1+(\mu_s - 3\sigma_s)a\right]$	0.8247	0.6838	0.6250
% of difference w.r.t. $1/\left(1+\mu_s a\right)$	**+3.1%**	**+2.6%**	**+2.3%**
$1/\left[1+(\mu_s + 3\sigma_s)a\right]$	0.7767	0.6504	0.5970
% of difference w.r.t. $1/\left(1+\mu_s a\right)$	**-2.9%**	**-2.4%**	**-2.2%**

APPENDIX B
LINEAR APPROXIMATION FOR $b(j)$

We claim that for the characteristics of the signals in our case and for our method, with *const* varying within ±3.1%:

$$b(j) = \frac{s(j)}{1+s(j)a} \approx const \cdot s(j) \tag{20}$$

which implies that $b(j)$ is essentially the scanner pattern $s(j)$.

First, in Table I we analyze the approximation:

$$\frac{1}{1+s(j)a} \approx \frac{1}{1+\mu_s a} \tag{21}$$

for the parameter ranges we have. The value of a is taken from Appendix A (the approximation for $r(i,j)$). Our analysis [40] showed that the scanner pattern standard deviation σ_s is typically smaller than 5, so we assume this value because it is the worst case for the approximation accuracy. Thus, for these values and parameter ranges, within ±3σ around the mean μ_s, we can assume that $\frac{1}{1+s(j)a} \approx \frac{1}{1+\mu_s a}$ within ±3.1%.

Next, substituting (3) and (21) in the equality of (20) gives:

$$b(j) = \frac{s(j)}{1+s(j)a} \approx \frac{1}{1+\mu_s(j)a}\left[\mu_s(j) + s_v(j)\right] = \tag{22}$$

$$= \frac{\mu_s(j)}{1+\mu_s(j)a} + \frac{1}{1+\mu_s(j)a} s_v(j). \tag{23}$$

Since the mean $\mu_s(j)$ is slowly varying, a low-pass filter $\mathcal{F}\{.\}$ on $b(j)$ as in (23) will remove the first term $\mu_s(j)/[1+\mu_s(j)a]$ and produce as output the processed and scaled variable part $s_v(j)$. Its scaling coefficient, however, still depends on the index j via the mean $\mu_s(j)$. Nevertheless, this is not a problem because the filter $\mathcal{F}\{.\}$ that we use has a short span (several pixels). Within this span, $\mu_s(j)$ is essentially the same, and thus the scaling factor $1/[1+\mu_s(j)a]$ is constant for the indices over which the filter $\mathcal{F}\{.\}$ operates and for the index j for which the Filtering Module produces the variable part $s_v(j)$. We note, however, that the approximation in (20) is appropriate only for our specific processing of $b(j)$ in order to extract the variable part $s_v(j)$, not in general.

Acknowledgment and Disclaimer

This research was supported by award MURI W911-NF-0710287 from the United States Army Research Office (ARO). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Army Research Office.

References

[1] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. London: Springer, Jun. 2009.

[2] G. Host, *CCD Arrays, Cameras, and Displays*. SPIE Optical Engineering Press, 1996.

[3] A. El Gamal, B. A. Fowler, H. Min, and X. Liu, "Modeling and estimation of FPN components in CMOS image sensors," in *Proc. SPIE 3301, Solid State Sensor Arrays: Development and Applications II, 168*, vol. 3301, Apr. 1998, pp. 168–177.

[4] K. Kurosawa, K. Kuroki, and N. Saitoh, "CCD fingerprint method-identification of a video camera from videotaped images," in *Proc. 1999 IEEE International Conference on Image Processing*, vol. 3, Oct. 1999, pp. 537–540.

[5] ——, "An approach to individual video camera identification," *Journal of Forensic Sciences*, vol. 47, no. 1, pp. 97–102, Jan. 2002.

[6] K. Kurosawa and N. Saitoh, "Fundamental study on identification of CMOS cameras," in *Proc. SPIE 5108, Visual Information Processing XII, 202*, vol. 5108, Aug. 2003, pp. 202–209.

[7] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for identification of images acquired with digital cameras," in *Proc. SPIE 4232, Enabling Technologies for Law Enforcement and Security, 505*, vol. 4232, Feb. 2001, pp. 505–512.

[8] N. Saitoh, K. Kurosawa, K. Kuroki, N. Akiba, Z. J. Geradts, and J. Bijhold, "CCD fingerprint method for digital still cameras," in *Proc. SPIE 4709, Investigative Image Processing II, 37*, vol. 4709, Jul. 2002, pp. 37–48.

[9] K. Kurosawa, K. Kuroki, K. Tsuchiya, N. Igarashi, and N. Akiba, "Case studies and further improvements on source camera identification," in *Proc. SPIE 8665, Media Watermarking, Security, and Forensics 2013*, vol. 8665, Mar. 2013, pp. 86 650C–86 650C–14.

[10] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *Proc. SPIE 5685, Image and Video Communications and Processing 2005*, vol. 5685, Apr. 2005, pp. 249–260.

[11] ——, "Digital 'bullet scratches' for images," in *Proc. 2005 IEEE International Conference on Image Processing*, vol. 3, Sep. 2005, pp. 65–68.

[12] M. K. Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, Mar. 1999, pp. 3253–3256.

[13] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[14] M. Chen, J. Fridrich, and M. Goljan, "Digital imaging sensor identification (further study)," in *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, Feb. 2007, pp. 65 050P–65 050P–13.

[15] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[16] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Proc. SPIE 7254, Media Forensics and Security*, vol. 7254, Feb. 2009, pp. 72 540I–72 540I–12.

[17] M. Goljan, M. Chen, and J. Fridrich, "Identifying common source digital camera from image pairs," in *Proc. 2007 IEEE International Conference on Image Processing*, vol. 6, Sep. 2007, pp. VI–125–VI–128.

[18] E. J. Alles, Z. J. Geradts, and C. J. Veenman, "Source camera identification for low resolution heavily compressed images," in *2008 International Conference on Computational Sciences and Its Applications*, Jun. 2008, pp. 557–567.

[19] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind identification of source cell-phone model," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, Sep. 2008.

[20] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification using sensor pattern noise," in *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX*, Jan. 2007, pp. 65 051K–65 051K–11.

[21] T. Gloe, E. Franz, and A. Winkler, "Forensics for flatbed scanners," in *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX*, Feb. 2007, pp. 65 051I–65 051I–12.

[22] H. Gou, A. Swaminathan, and M. Wu, "Robust scanner identification based on noise features," in *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX*, Jan. 2007, pp. 65 050S–65 050S–11.

[23] ——, "Intrinsic sensor noise features for forensic analysis on scanners and scanned Images," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 476–491, Sep. 2009.

[24] N. Khanna, A. K. Mikkilineni, and E. J. Delp, "Scanner identification using feature-based processing and analysis," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 123–139, Mar. 2009.

[25] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, Jun. 2010.

[26] B. B. Liu, Y. Hu, and H. K. Lee, "Source camera identification from significant noise residual regions," in *2010 IEEE International Conference on Image Processing*, Sept 2010, pp. 1749–1752.

[27] X. Kang, Y. Li, Z. Qu, and J. Huang, "Enhancing ROC performance of trustworthy camera source identification," in *Proc. SPIE 7880, Media Watermarking, Security, and Forensics III*, vol. 7880, 2011, pp. 788 009–788 009–11.

[28] ——, "Enhancing source camera identification performance with a camera reference phase sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 393–402, Apr. 2012.

[29] R. Li, C.-T. Li, and Y. Guan, "A reference estimator based on composite sensor pattern noise for source device identification," in *Proc. SPIE 9028, Media Watermarking, Security, and Forensics 2014*, vol. 9028, Feb. 2014, pp. 90 280O–90 280O–7.

[30] C.-H. Choi, M.-J. Lee, and H.-K. Lee, "Scanner identification using spectral noise in the frequency domain." in *2010 IEEE International Conference on Image Processing*, Sep. 2010, pp. 2121–2124.

[31] A. Cortiana, V. Conotter, G. Boato, and F. G. B. De Natale, "Performance comparison of denoising filters for source camera identification," in *Proc. SPIE 7880, Media Watermarking, Security, and Forensics III*, vol. 7880, Feb. 2011, pp. 788 007–788 007–6.

[32] G. Wu, X. Kang, and K. Liu, "A context adaptive predictor of sensor pattern noise for camera source identification," in *2012 IEEE International Conference on Image Processing*, Sep. 2012, pp. 237–240.

[33] W. van Houten and Z. Geradts, "Using anisotropic diffusion for efficient extraction of sensor noise in camera identification," *Journal of Forensic Sciences*, vol. 57, no. 2, pp. 521–527, 2012.

[34] A. J. Cooper, "Improved photo response non-uniformity (PRNU) based source camera identification," *Forensic Science International*, vol. 226, no. 1–3, pp. 132–141, 2013.

[35] A. R. Soobhany, K. Lam, P. Fletcher, and D. Collins, "Source identification of camera phones using SVD," in *2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 4497–4501.

[36] Y. Tomioka, Y. Ito, and H. Kitazawa, "Robust digital camera identification based on pairwise magnitude relations of clustered sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1986–1995, Dec. 2013.

[37] F. d. O. Costa, E. Silva, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution and device linking." *Pattern Recognition Letters*, vol. 39, pp. 92–101, 2014.

[38] N. Bartlow, N. Kalka, B. Cukic, and A. Ross, "Identifying sensors from fingerprint images," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop*, Miami, Fla., Jun. 2009, pp. 78–84.

[39] V. I. Ivanov and J. S. Baras, "Authentication of fingerprint scanners," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2011, pp. 1912–1915.

[40] V. I. Ivanov, "Authentication of fingerprint scanners," Ph.D. dissertation, University of Maryland, College Park, MD, USA, Apr. 2012.

[41] V. I. Ivanov and J. S. Baras, "Method and apparatus for authenticating area biometric scanners," U.S. patent 8,942,430, Jan. 27, 2015.

[42] ——, "Method and apparatus for authenticating biometric scanners," U.S. patent 8,577,091, Nov. 5, 2013.

[43] ——, "Method and apparatus for authenticating biometric scanners," U.S. patent 8,953,848, Feb. 10, 2015.

[44] ——, "Method and apparatus for authenticating swipe biometric scanners," U.S. patent 8,942,438, Jan. 27, 2015.

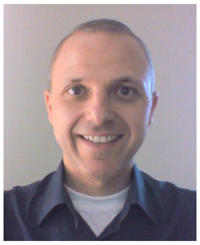
[45] ——. (2016, Nov.) Demonstration of the authentication of swipe fingerprint scanners. University of Maryland. College Park, MD. [Online]. Available: https://youtu.be/C_mlFxJaFsM

[46] T. Kleinz. (2008, Mar. 31) CCC publishes fingerprints of German Home Secretary. The H. [Online]. Available: http://www.h-online.com/security/news/item/CCC-publishes-fingerprints-of-German-Home-Secretary-734713.html

[47] E. Bott. (2013, Sep. 23) Apple's advanced fingerprint technology is hacked; should you worry? ZDNet. [Online]. Available: http://www.zdnet.com/apples-advanced-fingerprint-technology-is-hacked-should-you-worry-7000020908

[48] Z. Epstein. (2014, Apr. 15) Galaxy S5s fingerprint scanner has already been hacked, PayPal accounts at risk. BGR. [Online]. Available: http://bgr.com/2014/04/15/galaxy-s5s-fingerprint-scanner-hacked

John S. Baras (LF'13) received the Diploma in Electrical and Mechanical Engineering with highest distinction from the National Technical University of Athens, Greece, in 1970. He received the M.S. and Ph.D. degrees in Applied Mathematics from Harvard University, Cambridge, MA, in 1971 and 1973 respectively. Since 1973 he has been with the Department of Electrical and Computer Engineering, University of Maryland at College Park, where he is currently Professor, member of the Applied Mathematics, Statistics and Scientific Computation Program Faculty, and Affiliate Professor in the Fischell Department of Bioengineering and the Department of Mechanical Engineering. From 1985 to 1991 he was the Founding Director of the Institute for Systems Research (ISR) (one of the first six NSF Engineering Research Centers). In February 1990 he was appointed to the Lockheed Martin Chair in Systems Engineering. Since 1991 Dr. Baras has been the Director of the Maryland Center for Hybrid Networks (HYNET), which he co-founded. Dr. Baras has held visiting research scholar positions with Stanford, MIT, Harvard, the Institute National de Reserche en Informatique et en Automatique (INRIA), the University of California at Berkeley, Linkoping University and the Royal Institute of Technology (KTH) in Sweden. Among his awards are: the 1980 George S. Axelby Award of the IEEE Control Systems Society; the 1978, 1983 and 1993 Alan Berman Research Publication Awards from the NRL; the 1991, 1994 and 2008 Outstanding Invention of the Year Awards from the University of Maryland; the 1998 Mancur Olson Research Achievement Award, from the Univ. of Maryland College Park; the 2002 and 2008 Best Paper Awards at the 23rd and 26th Army Science Conferences; the 2004 Best Paper Award at the Wireless Security Conference WISE04; the 2007 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communication Systems; the 2008 IEEE Globecom Best Paper Award for wireless networks; the 2009 Maryland Innovator of the Year Award. In November 2012 he was honored by the Awards for both the Principal Investigator with Greatest Impact and for the Largest Selling Product with Hughes Network Systems for HughesNet, over the last 25 years of operation of the Maryland Industrial Partnerships Program. These awards recognized Dr. Baras pioneering invention, prototyping, demonstration and help with commercialization of Internet protocols and services over satellites in 1994, which created a new industry serving tens of millions worldwide. In 2014 he was awarded the 2014 Tage Erlander Guest Professorship by the Swedish Research Council, and a three year (2014-2017) Hans Fischer Senior Fellowship by the Institute for Advanced Study of the Technical University of Munich. Dr. Baras has been the initial architect and continuing innovator of the pioneering MS on Systems Engineering program of the ISR. Dr. Baras research interests include control, communication and computing systems. He holds eight patents and has four more pending. He is a Life Fellow of IEEE, a Fellow of SIAM and a Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA).



Vladimir I. Ivanov received a Ph.D. in Electrical Engineering from the University of Maryland, College Park, USA, and was also a research associate there. Before that, he was a system engineer in the ADSL modem team of Alcatel (now Nokia), Belgium, a signal processing engineer at Sparnex Instruments, Bulgaria, and a hardware engineer at Smartcom Bulgaria. He also received a Master's degree in Communication Technics and Technologies from the Technical University–Sofia, Bulgaria.