DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

Robust smartphone fingerprint by mixing device sensors features for mobile strong authentication

Irene Amerini - Media Integration and Communication Center (MICC), University of Florence, Florence, Italy
Paolo Bestagini - Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy
Luca Bondi - Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy
Roberto Caldelli - Media Integration and Communication Center (MICC), University of Florence, Florence, Italy and National Interuniversity Consortium for Telecommunications (CNIT), Parma, Italy
Matteo Casini - Media Integration and Communication Center (MICC), University of Florence, Florence, Italy
Stefano Tubaro - Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy.

Abstract

In the next coming years, many of our basic activities such as reading an e-mail, checking our bank account, buying on-line, etc., will be performed by using a smartphone in a mobile environment. It is quite obvious that the degree of security granted by a classic username-password access is not sufficient and that a stronger level of safeness is required. However, usually adopted additional instruments such as smart cards, USB sticks and OTP generators are not always available or usable in mobility. In this paper a possible solution which envisages the use of the user's own smartphone as a mean to grant a safer and easy mobile access is presented. The objective is to introduce a novel methodology to obtain a robust smartphone fingerprint by opportunely combining different intrinsic characteristics of each sensor. Modern mobile phones, in fact, have several kinds of sensors such as accelerometer, gyroscope, magnetometer, microphone and camera; such sensors can be used to uniquely identify each phone by measuring the specific anomalies left onto the signals they acquire. Satisfactory results have been obtained when the sensors are used in combination, especially accelerometer and digital camera, achieving a significant level of smartphone distinctiveness.

Introduction

Nowadays, and probably always more in the next coming years, many of our basic activities such as reading an e-mail, checking our bank account, buying on-line, etc., are performed by using a smartphone to access our personal accounts in a mobile environment. Generally, our actual degree of security is granted by the classic username and password access (something that the user knows). When a stronger level of safeness is required, additional instruments are usually adopted such as smart cards, USB sticks, OTP generators and so on (something that the user has got) in a two-factor authentication protocol [1]. Anyway, such means are not always available (must be carried around by the user at all times) or usable (they are not pluggable in a mobile device easily); so the need of a superior degree of security often conflicts with feasibility and usability. A possible solution could envisage the use of the user's own smartphone and its intrinsic characteristics as a mean to grant a safer mobile access by reducing the end-user involvement. The basic idea is to investigate and understand if it is possible to generate a specific fingerprint that allows to distinctively and reliably characterize each smartphone so to be used as a univocal security component when a strong authentication is needed. As a matter of fact, modern mobile phones are equipped with several kinds of sensors such as accelerometer, gyroscope, magnetometer, camera, etc. These sensors are characterized by peculiar anomalies left onto the acquired signals due to the imperfections generated during the manufacturing process [2]. Therefore, it is possible to measure these anomalies and exploit them as an asset for uniquely identifying each phone. The objective of this work is to present a methodology to obtain a robust smartphone fingerprint by opportunely combining different sensor fingerprints. The proposed methodology to create the smartphone fingerprint is firstly based on the individuation and definition of a set of distinctive features for each sensor; in our experiments we considered the accelerometer, the gyroscope and the camera. For the accelerometer and the gyroscope we considered two subsets of features both in the temporal and in the spectral domain, calculated onto the output data (x, y, z) acquired by each sensor [3, 4]. Concerning the camera, we computed spatial features derived from the 2D photo response non-uniformity (PRNU) noise [5, 6, 7], extracted from the R, G, B channels. All these features, organized in a vector, constitute the fingerprint of each device. According to these fingerprints, a classifier has been trained and some experimental tests to evaluate detection performances of the method have been carried out. Also different sub-combinations of the sensors have been considered in creating the fingerprint (e.g., only the accelerometer, accelerometer and the camera, accelerometer and gyroscope, etc.) to better understand which was the impact of each sensor on distinctiveness. Moreover, to decrease computational complexity, we investigated the possibility of reducing fingerprint size through hashing operations typically used for PRNU [8, 9, 10]. Furthermore, diverse operative conditions have to be analyzed: smartphone position (handheld or posed on a table of different materials), vibration on/off, with or without a cover and so on.

Sensors overview

Most smartphone devices, besides the photo-camera sensor, have built-in sensors that measure motion, orientation and various environmental conditions. These sensors are capable of providing raw data and are useful if you want to monitor three-dimensional device movement or positioning, or you want to monitor changes in the environment near a device. In general, both Android and iOS platforms, support three categories of sensors: motion sen-

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

sors, environmental sensors and position sensors. The first kind of sensors measures acceleration forces and rotational forces along three axes. This category includes accelerometer, gravity sensor, gyroscope and rotational vector sensor. The environmental sensors measure various parameters, such as ambient temperature and pressure, illumination and humidity. This category includes barometer, photometer and thermometer. The last kind of sensors measure the physical position of a device. This category includes orientation sensor, GPS and magnetometer. Some of these sensors are hardware-based and some are software-based. Hardware-based sensors are physical components built into a smartphone or a tablet device. They derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength or angular change. With regard to iOS platform many sensors are implemented such as TouchID, barometer, magnetometer, gyroscope, accelerometer, illumination and proximity. Regarding Android-powered devices, few of them have every type of sensor; for example, most handset devices and tablets have an accelerometer, a gyroscope and a magnetometer, but fewer devices have barometer or thermometer. On the other side, regarding operative system, all the sensors are supported by Android 4.0 and beyond (see Figure 1).

Sensor	Android 4.0	Android 2.3	Android 2.2	Android 1.5
ACCELEROMETER	✓	✓	✓	✓
AMBIENT_TEMPERATURE	✓	×	×	×
GRAVITY	✓	✓	×	×
GYROSCOPE	✓	✓	×	×
LIGHT	✓	✓	✓	✓
LINEAR_ACCELERATION	✓	✓	×	×
MAGNETIC_FIELD	✓	✓	✓	✓
ORIENTATION	✓	✓	✓	✓
PRESSURE	✓	✓	×	×
PROXIMITY	✓	✓	✓	✓
RELATIVE_HUMIDITY	✓	×	×	×
ROTATION_VECTOR	✓	✓	×	×
TEMPERATURE	✓	✓	✓	✓

Figure 1. *Sensors vs Android versions.*

Adopted sensors

Accelerometer

The accelerometer inside a smartphone is composed by a circuit having seismic mass (made up of silicon) that changes its position according to the orientation and it is attached to the circuit of the device. Actually an accelerometer is a circuit based on MEMS (Micro Electro Mechanical System), that measures the forces of acceleration that may be caused by gravity, by the movement or by tilting action. Such accelerations are measured in terms of g-force (m/s^2) on the three axes (x, y, z). MEMS-based accelerometers can consist of differential capacitors. Figure 2 shows the internal architecture of a MEMS-based accelerometer. As we can see there are several pairs of fixed electrodes and a movable seismic mass. Under no acceleration the distances d_1 and d_2 are equal and as a result the two capacitors are equal, but a change in the acceleration will cause the movable seismic mass to shift closer to one of the fixed electrodes causing a change in the generated capacitance. This difference in capacitance is detected and amplified to produce a voltage that is proportional to the acceleration. The minute imprecision in the electro-mechanical structure induce imperfections among the accelerometer chips.

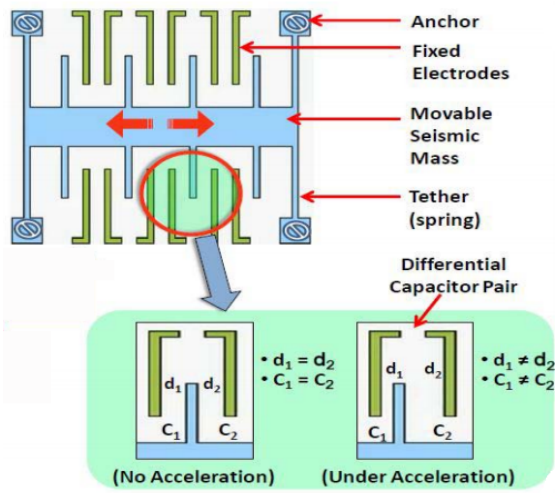

Figure 2. *MEMS accelerometer: how it works.*

Gyroscope

A gyroscope is a device for measuring or maintaining orientation, based on the principle of angular momentum. Mechanically, a gyroscope is a spinning wheel or disk in which the axle is free to assume any orientation (see Figure 3).

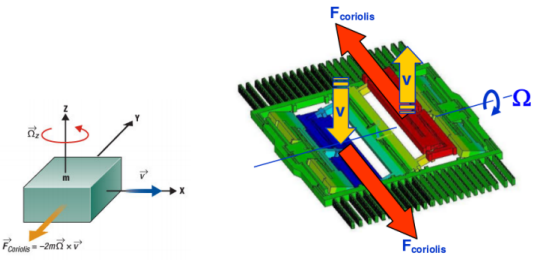

Figure 3. *Gyroscope: how it works.*

Same as accelerometer, gyroscope returns three-dimensional values along the three axes of the device and it measures the rate of rotation (in rad/s). MEMS-based gyroscopes use the Coriolis effect to measure the angular rate. Whenever an angular velocity Ω is exerted on a moving mass of weight m and velocity v, the object experiences a Coriolis force in a direction perpendicular to the rotation axis and to the velocity of the moving object. The Coriolis force is sensed by a capacitive sensing structure where a change in the vibration of the proof-mass causes a change in capacitance which is then converted into a voltage signal by the internal circuitry. The slightest imperfections in the electromechanical structure could introduce differences across chips.

Digital Camera

Digital camera acquisition pipeline is a complex chain of operations typically defined by the sketch in Figure 4. According to this diagram, light rays reflected by the scene are collected by a lens that focuses the rays on a CCD/CMOS sensor, after being mosaicked by a colour filter array (CFA). At this stage, each pixel of the generated image collects information about only one

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

colour (i.e., red, green, or blue). To obtain the final image I, some post-processing operations such as pixel interpolation and JPEG compression are applied.

Each of the aforementioned operations leaves some characteristic traces on the generated images. As a matter of fact, it is possible to detect on the final image I peculiar traces left by lens distortion [11], by the used de-mosaicking interpolation kernel [12], by quantization due to JPEG compression [13], and also by other phenomena [14, 15].

Even more interesting, each camera sensor leaves a peculiar noise artefact on the captured images known as photo response non-uniformity (PRNU) [2, 16], due to imperfections in the acquisition sensor manufacturing process. From a formal point of view, we can define each captured image as

$$I = I^{(0)} + I^{(0)}K + N,$$

where $I^{(0)}$ is a noiseless representation of the scene, N is an additive noise term, and K is the multiplicative PRNU noise trace [2]. This intrinsic noisy fingerprint, embedded in every image coming from the same camera, characterises not only the camera model, but also each camera instance (i.e., different devices of the same model). Many ways to estimate the term K have been proposed in the literature and they are typically based on denoising operations [17, 18].

Features vs Sensors

As mentioned in the previous section, sensors readings are affected by anomalies due to sensors imperfections. Our goal is to detect these anomalies and exploit them as an asset to understand which device generated them. To accomplish this goal, we make use of a set of features computed on signals acquired by the different sensors.

For both accelerometer and gyroscope it is possible to obtain raw values along three axes of the device at certain time. So, for a given time-stamp t we have two vectors of the following form: $a(t) = (a_x, a_y, a_z)$ and $\omega(t) = (\omega_x, \omega_y, \omega_z)$ for the accelerometer and gyroscope respectively.

As regarding accelerometer, 17 scalar features are extracted, on the basis of [3], in both time and frequency domains by using the MIRToolbox [19], a popular audio feature extraction library [20], starting from the two following signals:

$$T(k) = t(k+1) - t(k)$$

$$S(k) = \sqrt{a_x^2(k) + a_y^2(k) + a_z^2(k)}$$

The time domain features are calculated using T(k) and S(k) signals prior to interpolation while the frequency domain features are drawn from the interpolated versions. The signal T(k) has been considered as interesting being the time interval of acquisition slightly different for each device. In total a vector of 34 features f_a is obtained to describe the accelerometer sensor. In Table 1 and 2 all the features taken in consideration are outlined. For a complete description of each feature please refers to the MIRToolbox guide.

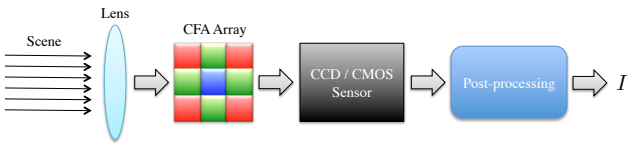

Figure 4. *Digital image acquisition pipeline. Ray-lights reflected from the scene are focused by the lens on a Color Filter Array superimposed to a CCD/CMOS sensor. The sensor output is processed and an RBG image is produced as output.*

Table 1: List of time domain features

Feature Name	Accelerometer	Gyroscope
Mean	x	
Std-Deviation	x	x
Average Deviation	x	x
Skewness	x	x
Kurtosis	x	x
RMS amplitude	x	x
Lowest value	x	x
Highest value	x	x
ZCR		x
Non-negative count		x

Table 2: List of frequency domain features

Feature Name	Accelerometer	Gyroscope
Spectral Std-Dev	x	x
Spectral Centroid	x	x
Spectral Skewness	x	x
Spectral Kurtosis	x	x
Spectral Crest	x	x
Irregularity-J	x	x
Smoothness	x	x
Flatness	x	x
Roll Off	x	x
Entropy		x
Brightness		x
Roughness		x

Regarding the gyroscope we consider data from each axis as a separate stream in the form of $\omega_x, \omega_y, \omega_z$. For all data streams, time and frequency domain characteristics are analyzed as for the accelerometer. To summarize the characteristics of each signal, 21 features are extracted, on the basis of [4], consisting of 10 temporal and 11 spectral features (listed in Table 1 and 2). In total a vector of 63 (21×3) features f_g is used to describe the gyroscope sensor for each device.

Concerning digital cameras, we exploit a feature vector based on the PRNU. To this purpose, let I be a 2D digital image, and W be the noise residual extracted from I according to the denoising wavelet-based procedure described in [2, 5]. Given a set of images I_p, $p \in \{1, ..., P\}$ from the same camera, the maximum-likelihood estimation of the PRNU for that camera is

$$K = \frac{\sum_p W_p I_p}{\sum_p I_p^2}, \qquad (1)$$

where operations are applied pixel-wise. For particularly accurate PRNU estimation, the use of flatfield bright images is encouraged

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

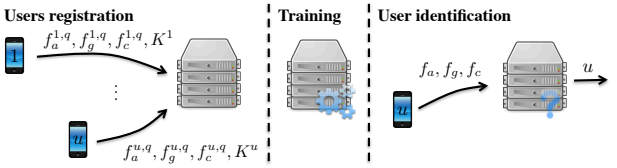

Figure 5. *Procedure pipeline.*

(e.g., shots of the sky). However, K can still be estimated (with less precision) using pictures of natural scenes. In this case, a sufficient greater number of images is needed (i.e., P must be increased with respect to the flatfield case) [2].

It is well known that the correlation between the noise residual W and the PRNU K assumes high values only if W is extracted from an image coming from the camera whose PRNU is K [2, 21, 18, 22]. Therefore, a feature characterizing a digital camera could in principle be the noise component W. However, for the sake of speeding up the device identification process, and motivated by state-of-the-art works on PRNU compression [8, 9], we perform an additional step. More specifically, we consider only the 512×512 pixels patch taken from the center of W and binarize it according to its sign. Formally, the feature vector used to describe a camera is $f_c = \text{sign}(W_{512\times512})$, where $W_{512\times512}$ is the 512×512 central portion of W. Notice that, thanks to binarization, f_c can be stored using only 512×512 bits (i.e., less than 33 kBytes), which is much less than a typical image size and allows very fast feature transmission also in low bandwidth conditions.

Procedure

In order to achieve device identification, we propose a methodology based on supervised classification and the features described in the previous section (i.e., f_a, f_g and f_c). In the considered scenario, there are two main entities: (i) the user owning a device that wants to be identified; (ii) a trained system that analyses the data provided by the user in order to identify (or not) its device. The overall identification procedure works in three steps, as shown in Figure 5: (i) each new user registers into the system; (ii) the system is trained based on the acquired registration data; (iii) device identification can be accomplished each time a user needs it by sending a new set of features (as fingerprint) to the system.

For the registration procedure, let us consider the u-th user out of all the possible U ones. First, he/she runs an application installed on its device to collect Q sensors readings from the accelerometer and the gyroscope, then finally shots $Q+P$ pictures of natural scenes (possibly neither saturated nor overly dark). From the first two sensors readings, Q different sets of feature vectors $f_a^{u,q}$ and $f_g^{u,q}$, $q \in \{1,...,Q\}$ are computed. From Q images, the device computes a set of feature vectors $f_c^{u,q}$, $q \in \{1,...,Q\}$. From the remaining P images, the PRNU K^u is estimated according to (1). The user finally sends to the server all the feature sets $f_a^{u,q}$, $f_g^{u,q}$ and $f_c^{u,q}$, $q \in \{1,...,Q\}$, and the PRNU estimate K^u. This procedure is followed by every user.

At this point, the system can be trained. To this purpose, from each 512×512 bits training feature $f_c^{u,q}$, a U-dimensional feature vector $\hat{f}_c^{u,q}$ is computed, defined as

$$\hat{f}_c^{u,q} = \left[\rho(f_c^{u,q}, K^1),\ \rho(f_c^{u,q}, K^2),\ \ldots,\ \rho(f_c^{u,q}, K^U)\right],$$

where ρ computes the cross-correlation. In other words, each component of $\hat{f}_c^{u,q}$ is the correlation value between $f_c^{u,q}$ and one of the possible PRNU templates K^u, $u \in \{1,...,U\}$. From an intuitive point-of-view, the vector $\hat{f}_c^{u,q}$ should point in the u-th direction, strongly indicating the correct camera. Features $f_a^{u,q}$, $f_g^{u,q}$ and $\hat{f}_c^{u,q}$ are concatenated and used to train a supervised classifier (in our experiments a Bagged Decision Tree [23]).

Once the system has been properly trained, each time a user needs to be identified, he/she can simply collect, by means of an ad-hoc application, a few seconds sensors reading and shoot a picture. The user's device automatically computes the tuple of features f_a, f_g and f_c and sends them to the server. As per training, feature f_c is converted into $\hat{f}_c$, and classification is performed using the concatenation of f_a, f_g and $\hat{f}_c$ as input for the classifier. This procedure increases the security level because does not require to store the "device fingerprint" within the smartphone in a safe folder but it is calculated each time on-the-fly. It could be conceived to integrate such information with other data (e.g. a time stamp, a gps position) to improve the authentication phase: future works will investigate these issues.

Experimental Results

Different experimental tests have been carried out to verify the effectiveness of the proposed methodology and some of them are presented hereafter in this section. In particular, in the first subsection the dataset of the considered smartphones is described (only Android platform has been analyzed) together with the way sensor acquisitions are performed, while the second subsection reports of the diverse test scenarios that have been investigated. Finally, the third subsection presents the achieved results in the various circumstances.

Test setup

Experimental tests have been carried out on 10 different smartphones that are listed hereafter in Table 3; to each device is assigned an index (third column) that will be used within experimental tests section. It is worthy to highlight that one half are

Table 3: List of smartphones

Device	Amount	Index
LG Nexus 5	5	*1,2,3,4,5*
Motorola Moto G 2015	1	*6*
Samsung Galaxy S3	2	*7,8*
Samsung Galaxy S4	1	*9*
Samsung Galaxy S2plus	1	*10*

of the same model and this has been chosen in order to investigate which was the actual capacity to distinguish also within devices of the brand and model. The acquisitions from the sensors have been done by means of a specific mobile application, named *SensorData* (see Appendix) which is able to interact with the smartphone sensors and get their output signal. Both for the accelerometer and for the gyroscope 20 acquisitions ($Q = 10$ for training plus 10 for testing), of 2 seconds each, have been taken. Because of the different characteristics of every smartphones, the number of samples within each acquisition is diverse, so each sequence has been resampled by using spline interpolation to compute spectral features. For what concerns digital images, they have been taken

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

at the default resolution and settings of the device, which is (in pixels): 2448×3264 (Nexus5, S2plus and S3), 1836×3264 (Motorola) and 2322×4128 (S4). For each camera, $P = 10$ images have been used for PRNU estimation, $Q = 10$ for training and other 10 for testing. Notice that only the central 512×512 pixels portion of each image has been used, thus no image resampling is needed even if different devices have different camera resolutions.

Test scenarios and evaluation metric

Different test scenarios have been envisaged in order to understand which could be the operative circumstances that could affect performances. There are, in fact, many aspects that can influence the acquisition phase both during training and testing steps: first of all, the smartphone's position (leaning on a table, hand-held by a still or slightly moving user, etc.), secondly, the usage conditions (characteristics of the table surface, presence or not of a telephone cover made of diverse materials, audio/vibration stimulation, running processes on the operating system, etc.) and so on. In the next subsection, some of the main achieved results are presented; in particular, two basic cases have been taken into account: when the acquisition takes place in a more controlled environment with the smartphone on a flat wooden surface and when it is hand-held (without any cover in any case). Tests have been carried out in order to understand how different conditions can impact on training with respect of testing and viceversa. The issue of using or not a vibrating impulse has been considered when acquiring only from the accelerometer, but no requirements have been imposed on the processes that are running on the smartphone. Classification has been done by resorting at a Bagged Decision Tree approach and experimental tests has been carried out both for each sensor separately and also in combination. The obtain results have been evaluated in terms of *F-score (F)* which is defined as in Equation (2):

$$F = (2 * Pr * Re)/(Pr + Re) \tag{2}$$

where $Pr = TP/(TP + FP)$ and $Re = TP/(TP + FN)$ stands for *Precision* and *Recall*. The overall *F-score* is the average of *F-score* computed on each class.

Results

This Section presents experimental results with reference to some test configurations exploited in Table 4. When the parameter "Vibration" is set up ON, it is intended that the acquisition for the accelerometer sensor has been done when there was a stimulation generated by the vibration motor.

Table 4: Test configurations

	Training set		Test set	
	Position	*Vibration*	*Position*	*Vibration*
Config1	Table	ON	Table	ON
Config2	Table	OFF	Table	OFF
Config3	Hand-held	ON	Hand-held	ON
Config4	Table	ON	Hand-held	ON
Config5	Hand-held	ON	Table	ON

It is interesting to underline that the two last test configurations conceive that training and testing conditions are not aligned. This circumstance has revealed as being more challenging, as expected, but it represents an actual operative scenario where there is no control over the end user activities. In Figure 6 the F-score values for the first three test configurations are depicted. It can be seen, as general, that the two sensors, accelerometer and gyroscope, are both able, by themselves, to provide reliable results in terms of device distinctiveness; however, when both are used together performances are improved with a F-score that tends to achieve values around 100%.

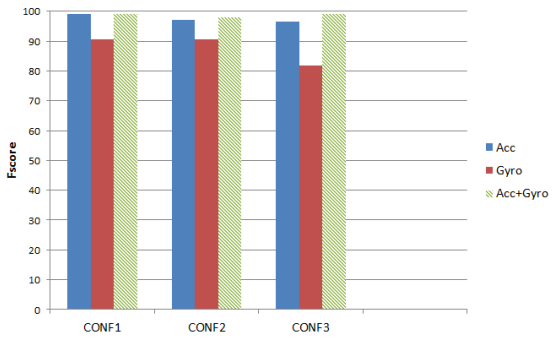

Figure 6. *F-score in percentage for the scenarios where training and testing are aligned.*

In particular, for the case of *Configuration3* (Hand-held) which is more challenging, it is interesting to notice that a high value of F-score is achieved similarly to what happened for the other two configurations in which the smartphone is leaning on the table.

On the contrary, in Figure 7 the results obtained for *Configuration4* and *Configuration5* (e.g. training and testing misaligned) are presented. It is immediate to comprehend that performances are worsened and that the approach to join both sensors provides some benefits granting higher F-score values: 70% is achieved for *Configuration5* at most.

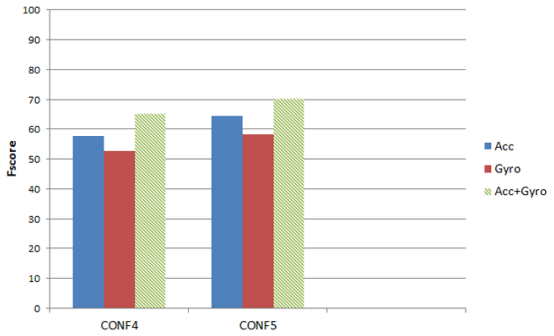

Figure 7. *F-score in percentage for the scenarios where training and testing are not aligned.*

In Figure 8, the results obtained when also the camera sensor is taken into account are shown. It can be evidenced that the first and the third configurations, that already presented very good performances, tends to reach 100% while the second one remains almost unaltered, even a bit lower, and this could be due to the fact that not using the vibration for the accelerometer impacts on performances as a whole.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

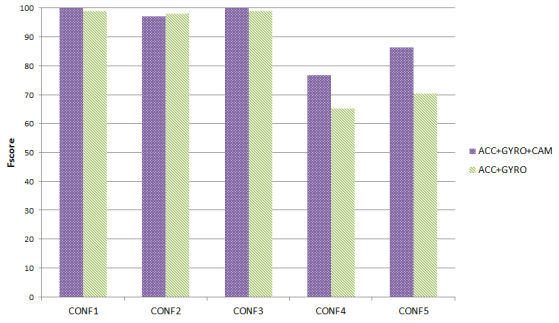

Figure 8. *F-score in percentage for all the scenarios when the camera sensor is added.*

It is appreciable that for *Configuration4* and *Configuration5* results are strongly improved and, for instance, a F-score of about 85% is reached in the last case.

In the light of these results, it is worth making a specific comment on the used camera fingerprint. As a matter of fact, PRNU is known to be very robust and reliable, ensuring very high accuracy in camera identification (often higher than 90%), especially when used for high-end devices. However, in our scenario, we must consider a series of constraints: (i) the user cannot be asked to shoot too many pictures for system training; (ii) we typically have no control on the kind of pictures the user sends; (iii) smartphones camera have hardware limitations with respect to high-end devices; (iv) the generated feature vector must be small enough to enable transmission also in low-bandwidth cases. Therefore, we work in a very disadvantaged scenario: (i) PRNU is estimated using only a few (i.e., $P = 10$) images; (ii) these images represent natural scenes and are not flatfield; (iii) the correlation procedure is performed using a strongly quantized (i.e., binarization according to the sign) image noise W; (iv) smartphones cameras are strongly affected also by other noise components with respect to high-end devices. This is the main reason we cannot expect to reach an even higher accuracy using camera fingerprint in this scenario.

In Figure 9, the test *Configuration5* when also the sensor camera is taken into account is analyzed in detail. The confusion matrix structured onto the ten target/output classes is presented (indexes assigned to each class refer to the smartphone indicated in Table 3); over the diagonal there are the correct classification (green blocks) while all the other are wrong (red blocks). It is interesting to highlight that most of the performance decrement is given by the classes numbered 1 and 2 (top-left of the matrix) otherwise performances would be averagely around a F-score of 95%. Going into details, it has been observed that these two smartphones are both LG Nexus5 and it happens that they are erroneously exchanged with other devices of the same model.

Conclusions

The novelty of the paper is to propose a method to combine different smartphone sensors to obtain a reliable, distinctive and easy-to-use fingerprint able to characterize each single device univocally. The challenge is to succeed in defining effective features and integrating them, though extracted from different sensors, to achieve a robust distinctiveness among diverse smartphones to be used in strong authentication procedures. The results obtained so

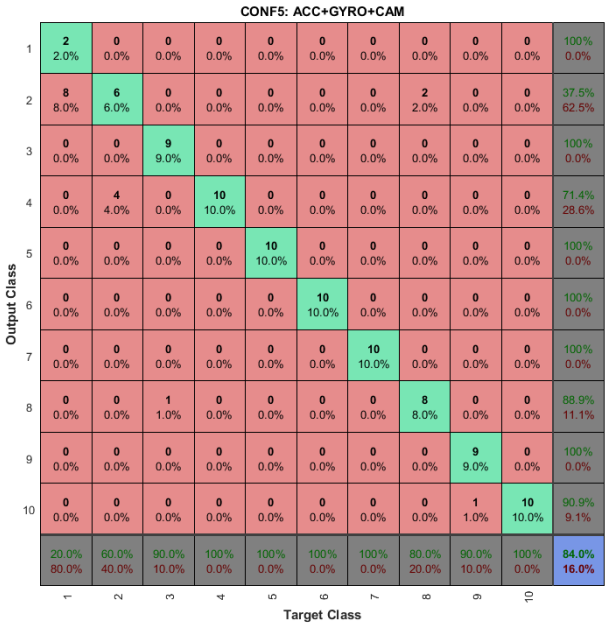

Figure 9. *Case Configuration5 (considering also sensor camera): confusion matrix. In the right end column* Precision *and* False Discovery Rate (1-Precision) *are reported respectively while in the bottom end row* True Positive Rate (TPR) *and* False Negative Rate (1-TPR) *are indicated respectively.*

far are encouraging, since mixing features from different sensors outperforms the classification obtained using the features from each sensor separately. Next steps will be devoted to the study of open-set scenarios, update the training procedure iteratively for new users to be registered into the system and to the exploitation of even more sensors when possible, particularly, to improve distinctiveness within devices of the same model.

Acknowledgments

This work was partially supported by the SMARTVINO Project funded by the PRAF 2012-2015-1.2.e programme of the Tuscany Region (Italy).

Appendix: the application SensorData

SensorData is an Android mobile application developed for the data acquisition from built-in device sensors by using the *Android Sensor Framework*. The Sensor Framework provides several classes and interfaces to perform a wide variety of sensor-related tasks. For example, the Sensor Framework can be used to:

- determine which sensors are available on a device;
- determine an individual sensor's capabilities, such as its maximum range, manufacturer, power requirements, and resolution;
- acquire raw sensor data and define the minimum rate at which you acquire sensor data;
- register and unregister sensor event listeners that monitor sensor changes.

SensorData application measures acceleration forces and rotational forces along the three axes (x, y, z) (see Figure 10).

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

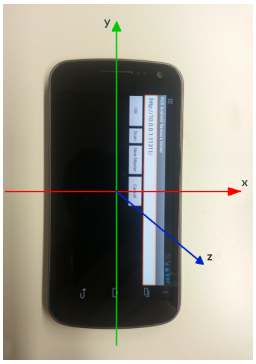

Figure 10. *Smartphone reference axes.*

In the top of user interface is reported the data delay (or sampling rate) that controls the interval at which sensor events are sent to the application via the callback method. The default data delay is set to *SENSOR_DELAY_FASTEST* (0 microsecond delay), but it could be changed. *SensorData* application implements three different sensor listeners (see Figure 11): accelerometer (m/s^2), gyroscope (rad/sec) and gravity sensor (μT), individually selectable from the user interface. Data acquisitions can be made for preset time values, variable from 1 to 60 seconds. The user can start the test by initially choosing the sensor's type from the radio buttons and then make the time period selection. With START button the user begins the data acquisition. Data acquisition can be stopped at anytime through the STOP button.

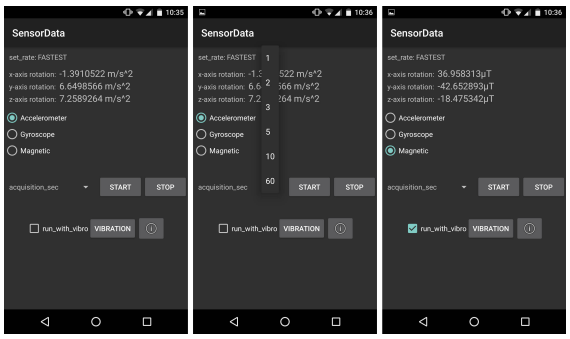

Figure 11. *Screenshots of the application "Sensordata".*

If necessary, the vibration can be activated before the acquisition starting. For each acquisition, the data are written on a specified text file in the device memory storage.

References

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall Press, 2010.

[2] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *Electronic Imaging*, 2005.

[3] S. Dey, N. Roy, W. Xu, R.R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometer make smartphones trackable," in *NDSS Symposium*, 2014.

[4] A. Das, N. Borisov, and M. Caesar, "Exploring ways to mitigate sensor-based smartphone fingerprinting," arXiv:1503.01874.

[5] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 1, pp. 205–214, 2006.

[6] I. Amerini, R. Becarelli, B. Bertini, and R. Caldelli, "Acquisition source identification through a blind image classification," in *IET Image Processing*, 2014.

[7] I. Amerini, R. Caldelli, P. Crescenzi, A. Del Mastio, and A. Marino, "Blind image clustering based on the normalized cuts criterion for camera identification," *Sig. Proc.: Image Comm.*, vol. 29, no. 8, pp. 831–843, 2014.

[8] M. Goljan, J. Fridrich, and T. Filler, "Managing a large database of camera fingerprints," in *SPIE Conference on Media Forensics and Security II*, 2010.

[9] M. Goljan and J. Fridrich, "Sensor fingerprint digests for fast camera identification from geometrically distorted images," in *SPIE Conference on Media Watermarking, Security, and Forensics*, 2013.

[10] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Compressed fingerprint matching and camera identification via random projections," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 10, pp. 1472–1485, 2015.

[11] M. Goljan and J. Fridrich, "Identifying images corrected for lens distortion using sensor fingerprints," in *SPIE Conference on Media Watermarking, Security, and Forensics*, 2012.

[12] S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro, "Demosaicing strategy identification via eigenalgorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[13] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compressions using first digit features," *APSIPA Transactions on Signal and Information Processing*, vol. 3, pp. 1–10, 2014.

[14] S.-H. Chen and C.-T. Hsu, "Source camera identification based on camera gain histogram," in *IEEE International Conference on Image Processing (ICIP)*, 2007.

[15] A.E. Dirik, H.T. Sencar, and N. Memon, "Digital single lens reflex camera identification from traces of sensor dust," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 3, pp. 539–552, 2008.

[16] K. Rosenfeld and H.T. Sencar, "A study of the robustness of PRNU-based camera identification," in *SPIE Conference on Media Forensics and Security*, 2009.

[17] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva, "Analysis of denoising filters for photo response non uniformity noise extraction in source camera identification," in *International Conference on Digital Signal Processing (ICDSP)*, 2009.

[18] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in PRNU based forgery detection," in *ACM Workshop on Multimedia in Forensics, Security and Intelligence (MiFor)*, 2010.

[19] O. Lartillot and P. Toiviainen, "MIR in Matlab (ii): A toolbox for musical feature extraction from audio," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2007, [https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox].

[20] P. Bestagini, M. Zanoni, L. Albonico, A. Paganini, A. Sarti, and S. Tubaro, "Feature-based classification for audio bootlegs detection," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2013.

[21] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva, "Estimate of PRNU noise based on different noise models for source camera identification," *International Journal of Digital Crime and Forensics*, vol. 2, pp. 21–33, 2010.

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-088

[22] L. Gaborini, P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Multi-clue image tampering localization," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014.

[23] T.K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 20, pp. 832–844, 1998.

Author Biography

Irene Amerini received the Laurea degree in computer engineering (2006) and the Ph.D. degree in computer engineering, multimedia, and telecommunication (2010) from the University of Florence, Florence, Italy. She is currently a Post-Doctoral Researcher with the Image and Communication Laboratory, Media Integration and Communication Center, University of Florence. She was a Visiting Scholar with Binghamton University, (NY, USA) in 2010. Her main research interests focus on multimedia security technologies, secure media and multimedia forensics.

Paolo Bestagini received the M.Sc. in Telecommunications Engineering and the Ph.D in Information Technology at the Politecnico di Milano, Italy, in 2010 and 2014, respectively. In 2012 he has been visiting student at Imperial College London, UK. He is currently a Post-Doctoral Researcher at the Image and Sound Processing Group (ISPG), Politecnico di Milano. His research activity is focused on multimedia forensics and acoustic signal processing for microphone arrays.

Luca Bondi received the MS degree in Computer Engineering in December 2014 from Politecnico di Milano. He is currently pursuing a PhD degree at the Department of Electronics, Informatics and Bioengineering, Politecnico di Milano. His research activities are focused on deep neural networks for efficient visual features extraction and compression.

Roberto Caldelli received the Laurea degree in electronic engineering (1997) and the Ph.D. degree (2001) in computer science and telecommunication from the University of Florence (Italy). From 2005 till 2013, he has been an Assistant Professor with the Media Integration and Communication Center, University of Florence. From 2014 he has joined the National Inter-University Consortium for Telecommunications (CNIT) as permanent researcher. His main research activities include digital image processing, interactive television, image/video digital watermarking and multimedia forensics.

Matteo Casini graduated in Telecommunications Engineering at the University of Florence (Italy) with a thesis concerning the study and development of a MHP application for on-line credit card payment through digital terrestrial television. He is currently working at Media Integration and Communication Center, University of Florence and is concerned of study and development of media security tools for mobile applications and of application tools for distribution of data in interactive environments.

Stefano Tubaro completed his studies in Electronic Engineering at the Politecnico di Milano, Italy, in 1982. He joined the Politecnico di Milano, first as a researcher and in 1991 as an Associate Professor. Since 2004 he has been appointed as Full Professor of Telecommunications. His current research interests are on advanced algorithms for video and sound processing and for image and video tampering detection. He authored over 180 scientific publications and more than 15 patents.