Silent and Continuous Authentication in Mobile Environment

Gerardo Canfora[1], Paolo di Notte[3], Francesco Mercaldo[1,2] and Corrado Aaron Visaggio[1]
[1]*Department of Engineering, University of Sannio, Benevento, Italy*
[2]*Centro Regionale Information Communication Technology - CeRICT scrl, Benevento, Italy*
[3]*Reply, Torino, Italy*

Keywords: Continuous Authentication, Silent Authentication, Security, Behavioral Models, Android.

Abstract: Due to the increasing pervasiveness of mobile technologies, sensitive user information is often stored on mobile devices. Nowadays, mobile devices do not continuously verify the identity of the user while sensitive activities are performed. This enables attackers full access to sensitive data and applications on the device, if they obtain the password or grab the device after login. In order to mitigate this risk, we propose a continuous and silent monitoring process based on a set of features: orientation, touch and cell tower. The underlying assumption is that the features are representative of smartphone owner behaviour and this is the reason why the features can be useful to discriminate the owner by an impostor. Results show that our system, modeling the user behavior of 21 volunteer participants, obtains encouraging results, since we measured a precision in distinguishing an impostor from the owner between 99% and 100%.

1 INTRODUCTION

Smartphones have become ubiquitous computing platforms, allowing users to access the Internet and many online services anytime and anywhere. As a personal device, a smartphone contains important private information, such as text messages, always-logged-in emails, contact list etc. As a portable device, a smartphone is much easier to get lost or stolen than conventional computing platforms. In order to prevent the private information stored on smartphones from falling into the hands of adversaries, the authentication of mobile devices has become an important issue. Most of the methods for authenticating users on mobile devices define an entry point into the system. Login-time pins and textual and graphical passwords (Akula and Devisetty, 2004; Davis et al., 2004; Dhamija and Perrig, 2000) are the most popular mechanisms for authenticating smartphone users. With the growing popularity of touch interface based mobile devices, the touch-surface has become the dominant human-computer interface. This has led to the need for authentication techniques better suited to a touch interface, such as Sae-Bae and Memon (Sae-Bae and Memon, 2013). These mechanisms suffer from two drawbacks: (i) they are static, that means they authenticate the user only at the beginning of the session and does not offer any protection against illicit access post login, i.e. in the case of abandoned device or when a remotely controlled program runs on the device, (ii) passwords and pins require user's attention to their entry and therefore they are not suitable for continuous authentication, and finally (iii) passwords can be stolen or forgotten. In order to overcome this one-step authentication, the *continuous authentication*, also called *active authentication* was introduced, where the identity of the user is verified during all the usage of the device. Continuous authentication methods complement the entry point methods by monitoring the user after a successful login.

Research on continuous authentication started in 1995 when Shepherd (Shepherd, 1995) and Monrose et al. (Monrose and Rubin, 1997) showed some impressive results on continuous authentication using keystroke dynamics. Continuous authentication is prevalently realized by checking two subsets of biometric authentication, physiological and behavioral. These authentication methods identify the user through measurable physical or behavioral characteristics.

Physiological biometric authentication measures physical characteristics of users body that make them unique. Physiological methods include fingerprint scanning, facial recognition, hand geometry recognition or retinal scans (Bhattacharyya et al., 2009). One drawback of physical biometrics is that they need specific hardware to collect the biometric data. This hardware entails additional costs and a layer to the

Canfora, G., Notte, P., Mercaldo, F. and Visaggio, C.
Silent and Continuous Authentication in Mobile Environment.
DOI: 10.5220/0005965500970108
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT*, pages 97-108
ISBN: 978-989-758-196-0

login process. Another drawback is that all of the physical biometric methods still produce an error rate which is not acceptable for real applications (Bailey et al., 2014).

Behavioral biometric authentication makes use of behavioral profiles of a user resulting from both psychological and physiological differences from person to person. Behavioral methods include keystroke dynamics (Joyce and Gupta, 1990; Brown and Rogers, 1993), mouse dynamics (Ahmed and Traore, 2005; Shen et al., 2010), voice recognition (Bhattacharyya et al., 2009), signature verification (Bhattacharyya et al., 2009) and Graphical User Interface (GUI) usage analysis (Gamboa and Fred, 2004). Due to the variability of the human body and mind, the adoption of this type of biometrics has lagged behind physiological biometrics.

In this paper we propose a method to silently and continuously verify the identity of a mobile user. Our method defines the user profile by merging together information about:

- how the user handles the device;
- how the user touches the keyboards;
- daily habits of the user.

Using well-known machine learning algorithms we classify the features set obtained from real devices employed in real environment to test the effectiveness of the features extracted.

The experiment demonstrated that the fusion of these three classes of features is able to detect impostors with a precision of 0.995, a false acceptance rate of 0.7% and a false rejection rate of 0.3%, which are values largely better than the values gathered with the antagonist methods for the continuous authentication that can be found in literature.

The main advantages of our method are :

- these biometrics can be captured by using the device built-in sensors without additional hardware;
- the features can be gathered with a good degree of precision and are not influenced by external factors (noises, air impurity);
- they can be collected while the user is using the mobile phone: the user is not required to enter any image or voice (this is the reason why our method is called *silent*);
- the performances obtained are significantly better than those reported in literature.

The paper proceeds as follows: Section 2 describes and motivates our detection method; Section 3 illustrates the results of experiments; Section 4 discusses related work; finally, conclusions are drawn in Section 5.

2 THE METHOD

In this section we discuss the approach we propose. Basically, we extract a set of features, captured directly on the device, representing the user behaviour.

We consider the following elements of the human behaviour in order to characterize mobile users:

- *How users hold the devices*: each user is inherently characterized by the way she holds the device. The inclination is determined by the user's arm and eyes, in order to make the experience of device use comfortable. Indeed the way in which the device is held depends on both the anatomic aspects and the personal habits, including the confidence degree with the device;
- *How users write on the device keyboard*: in addition to the combination of eyes and arm position, an user is also characterized by the way she types. It is a consequence of the first aspect we expose, e.g. an user that keeps the device with one hand will have a different typing style from an user that keeps the device with two hands;
- *User daily habits*: this feature captures the frequency a user is located in a certain place. For instance, people usually work in the same place, and this is the reason why the device is connected for many hours a day with the same cell tower.

Starting from these considerations we define the information we need to collect to characterize mobile users: information about orientation (*How users hold the devices*), about touch (*How users write on the device keyboard*) and about cell (*User daily habits*).

In order to obtain *Orientation information* we use the orientation sensor, which uses a device's geomagnetic field sensor in combination with a device's accelerometer. Using these two hardware sensors, an orientation sensor provides data for the following three dimensions (i.e., the orientation features):

- *Azimuth*, i.e. the degrees of rotation around the z axis. This is the angle between magnetic north and the device's y axis. For example, if the device's y axis is aligned with magnetic north this value is 0, and if the device's y axis is pointing south this value is 180. Likewise, when the y axis is pointing east this value is 90 and when it is pointing west this value is 270.
- *Pitch*, i.e. the degrees of rotation around the x axis. This value is positive when the positive z axis rotates toward the positive y axis, and it is negative when the positive z axis rotates toward the negative y axis. The range of values is 180 degrees to -180 degrees.

- *Roll*, i.e. the degrees of rotation around the y axis. This value is positive when the positive z axis rotates toward the positive x axis, and it is negative when the positive z axis rotates toward the negative x axis. The range of values is 90 degrees to -90 degrees.

The orientation sensor derives its data by processing the raw sensor data from the accelerometer and the geomagnetic field sensor. Because of the heavy processing that is involved, the accuracy and precision of the orientation sensor is diminished (specifically, this sensor is only reliable when the roll component is 0). As a result, the orientation sensor was deprecated in Android 2.2 (API level 8).

With *Orientation information*, we are referring to *Pitch*, *Roll* and *Azimuth* features.

The *Touch information* is retrieved using a 1 pixel per 1 pixel size window, placed at the top left corner of the touch-screen. When a touch event occurs the hardware layer sends a signal to a component called dispatcher, that is able to perform a check using the User Identifier. The check verifies who generated the information: if the check fails, the information is lost. The check was introduced in 2012 from API 17. We want to gather information everywhere in uncontrolled environment, and we used the WATCH_OUTSIDE_TOUCH parameter. Android documentation says "Note that you will not receive the full down/move/up gesture, only the location of the first down"[1]. We obtain the timestamp for each touch event and by a difference with the last touch event we retrieve the *Touch Gap*, i.e. the *Touch information*.

The *Cell information* is retrieved by using getCid() and getLac() methods provided by the GsmCellLocation class.

The getCid() method returns the GSM Cell ID (CID), an unique number used to identify each Base transceiver station (BST), while the getLac() method returns the location area code. A location area is a set of base stations that are grouped together to optimise signalling. To each location area, a unique number called location area code (LAC) is assigned. The LAC is broadcast by each base station, known as a BST in GSM, or a Node B in UMTS, at regular intervals.

Regarding the *Cell information* we consider as features the *CID* and the *LAC*.

In order to collect the features we implemented three components:

- *an Android application*: the application is able to retrieve the user-oriented features we previously described, the application works at user-level and it does not require root privileges. The application was developed to retrieve the full feature set: (i) *Orientation information*, (ii) *Touch information* and (iii) *Cell information*;
- *Drop Server*: we deployed a server to collect all the information retrieved by various devices with the Android application installed;
- *NoSQL database*: once the information is retrieved, it is stored in a non-relational database (namely, MongoDB) for facilitating the analysis. We chose this type of database because of the great amount of data accumulated from the devices, but also for its schema-less feature. We developed a script to read JSON files obtained from the devices and to insert them into the database. Using MongoDB we created one collection for each device and we stored every received features in an heterogeneous collection associated to it, it stored JSON files using an own format called BSON.

We use the Accessibility Service that must be enabled by users in order to collect the sensitive information we need.

3 THE EVALUATION

We designed an experiment in order to evaluate the effectiveness of the proposed technique.

More specifically, the experiment is aimed at verifying whether the features are able to classify a behaviour trace as performed by the owner or by an impostor. The classification is carried out by using a classifier built with the features discussed in the previous section. The evaluation consists of two stages: (i) a comparison of descriptive statistics of the populations of traces; and (ii) a classification analysis aimed at assessing whether the features are able to correctly classify the owner's and the impostor's behaviour traces.

We observed 21 users for 10 days: the evaluation time window began on *September 1, 2015* and finished on *September 11, 2015*. At the end of the observation window we gathered approximately 1 GB of raw data. Unfortunately two devices presented issues with the sensors and we were forced to conduct analysis on the remaining 18 users. Another user was not considered in the final results because the device suffered of incompatibility.

Table 1 shows the observed devices used to evaluate our method.

[1] https://android.googlesource.com/platform/frameworks/native/+/master/include/android/window.h

Table 1: Devices involved in the evaluation with owner characterization.

#	Device	OS	API	AGE	SEX	USED
1	*Samsung Galaxy S3*	KitKat	19	19	Female	Y
2	*LGE Nexus 5*	Marshmallow	23	22	Male	Y
3	*Samsung Galaxy S5*	Lollipop	21	26	Male	Y
4	*Samsung Galaxy S3*	KitKat	19	27	Male	Y
5	*Samsung Mini 2*	Gingerbread	10	33	Male	Y
6	*Samsung Note 3*	Lollipop	21	26	Male	Y
7	*Samsung S5 Dual*	Lollipop 5.1	22	28	Male	Y
8	*Samsung S4*	Jelly Bean	18	56	Male	Y
9	*Samsung Galaxy S4*	Jelly Bean	17	24	Male	Y
10	*LGE Nexus 5*	Lollipop	22	23	Female	Y
11	*Samsung Galaxy A5*	Jelly Bean	21	27	Male	N
12	*LGE G2*	Jelly Bean	17	29	Male	Y
13	*Samsung Galaxy S2*	Jelly Bean	16	30	Male	Y
14	*Samsung Corby*	Froyo	8	26	Male	N
15	*HUAWEI P6*	Jelly Bean	17	24	Male	Y
16	*HUAWEI P1*	Ice Cream	14	22	Female	Y
17	*OnePlus A0001*	Lollipop	22	27	Male	Y
18	*HUAWEI Honor 6*	KitKat	19	24	Female	Y
19	*HUAWEI P7*	KitKat	19	27	Male	Y
20	*HUAWEI P8 Lite*	KitKat	19	25	Female	Y
21	*HUAWEI Y530*	Jelly Bean	18	21	Male	N

3.1 Descriptive Statistics

The analysis of box plots related to the six features helps to identify whether the features are helpful to discriminate the behaviour of users.

Figure 1 shows the box plots related to *Pitch* feature for each user involved in the evaluation, while figure 2 shows the box plots related to *Roll* feature and figure 3 shows the box plots related to *Azimuth* feature. All these box plots do not exhibit significant differences among the different users. A similar consideration can be done for the features related to the *touch information*, represented in figures 4, and 5. The things change when we consider the *LAC* box plots illustrated in figure 6. As a matter of fact, users exhibit an evident diversity among each other, which is represented by the different level of medians for each user and by the variability of the box plots' width. The analysis of descriptive statistics suggests that both *orientation* and *touch information* singularly taken could be insufficient to discriminate the owner from the impostor. The classification analysis will complete the picture, by indicating that the combination of all the measures can successfully help to identify correclty the impostors.

3.2 Classification Analysis

We classified the features extracted using Weka[2], an open source machine learning library, using two classification algorithms: J48 and RandomForest.

Five metrics were used to evaluate the classification results: precision, recall, ROC Area, FAR and FRR.

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$Precision = \frac{tp}{tp+fp}$$

where *tp* indicates the number of true positives and *fp* indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e. how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$Recall = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The Roc Area is defined as the probability that a positive instance randomly chosen is classified above a negative randomly chosen.

The last two metrics we consider are used in biometrics in order to verify the instance of a security system incorrectly identifying an unauthorized per-

[2]http://www.cs.waikato.ac.nz/ml/weka/

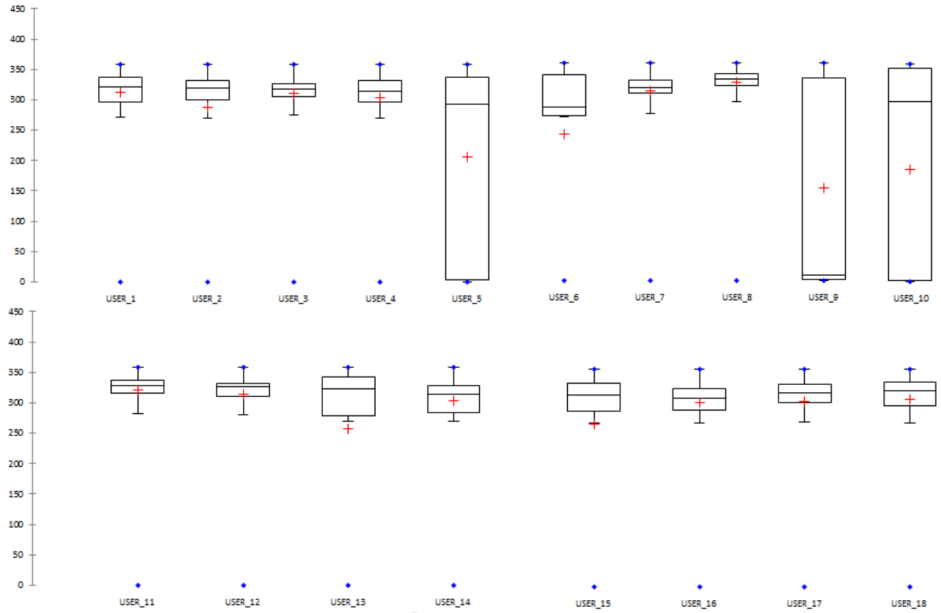

Figure 1: Box plots related to the pitch feature for each user involved in the experiment.

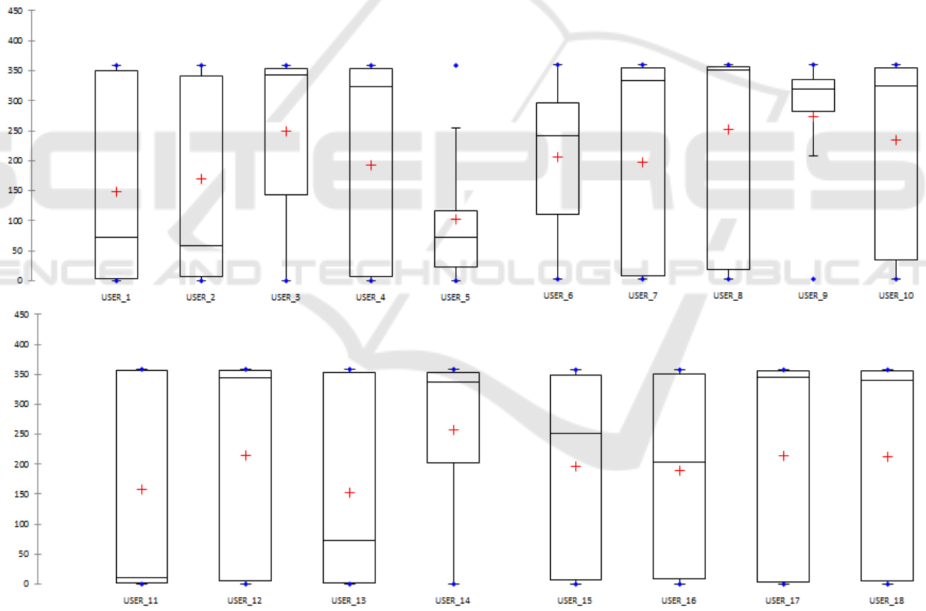

Figure 2: Box plots related to the roll feature for each user involved in the experiment.

son: *False Acceptance Rate* and *False Rejection Rate*.

The false acceptance rate (FAR) is the measure of the likelihood that the biometric security system will incorrectly accept an access attempt by an unauthorized user. A system's FAR typically is stated as the ratio of the number of false acceptances (fa) divided by the number of impostor attempts (ia):

$$\textit{False Acceptance Rate} = \frac{fa}{ia}$$

The FAR spans in the interval [0,1]: closer to 0 the FAR is the better is the capability to recognize correctly the impostor.

In biometrics, FRR, or false rejection rate is the instance of a security system failing to verify or identify an authorized person. Also referred to as a type I error, a false rejection does not necessarily indicate a flaw in the biometric system; for example, in a fingerprint-based system, an incorrectly aligned finger on the scanner or dirt on the scanner can result in

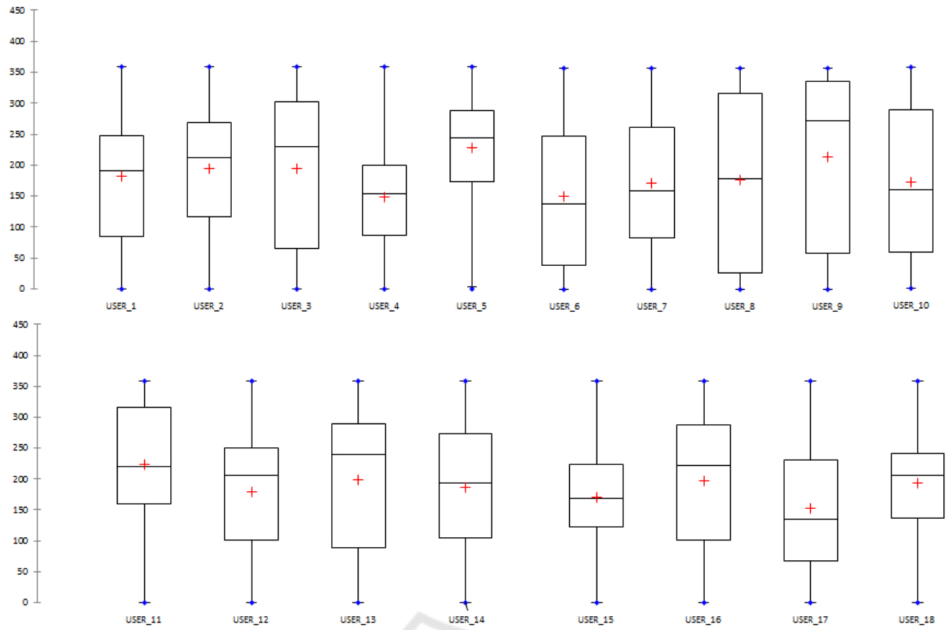

Figure 3: Box plots related to the azimuth feature for each user involved in the experiment.

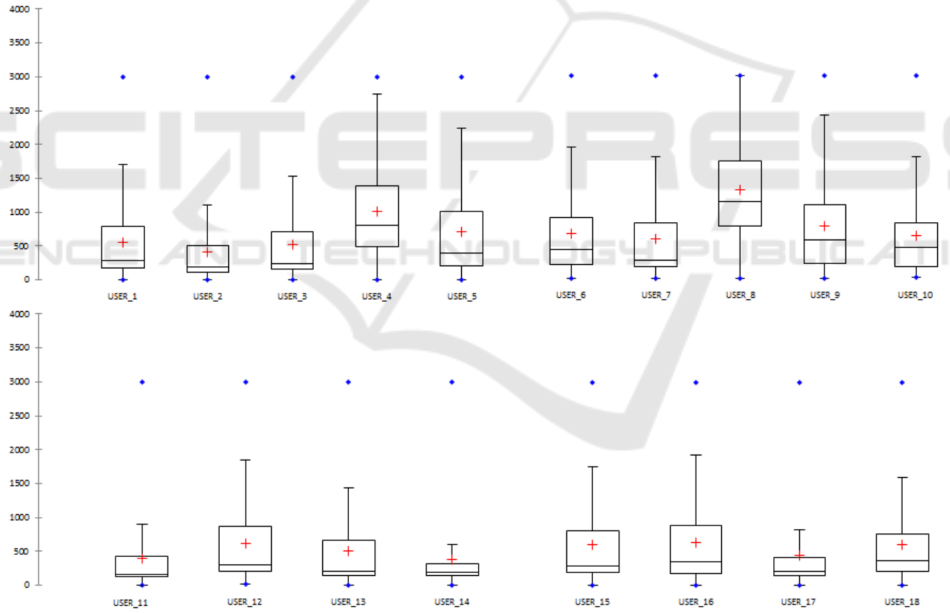

Figure 4: Box plots related to the touch gap feature for each user involved in the experiment.

the scanner misreading the fingerprint, causing a false rejection of the authorized user. The false rejection rate is the measure of the likelihood that the biometric security system will incorrectly reject an access attempt by an authorized user. A system's FRR typically is stated as the ratio of the number of false rejections (fr) divided by the number of owner attempts (oa).

The FRR is defined as:

$$\textit{False Rejection Rate} = \frac{fr}{oa}$$

The best FRR has the value of 0, while the worst FRR has the values of 1.

The classification analysis consisted of building classifiers in order to evaluate features accuracy.

For training the classifier, we defined T as a set of labelled behaviour traces (BT, l), where each BT is associated to a label $l \in \{impostor, owner\}$. For each BT we built a feature vector F $\in R_y$, where y

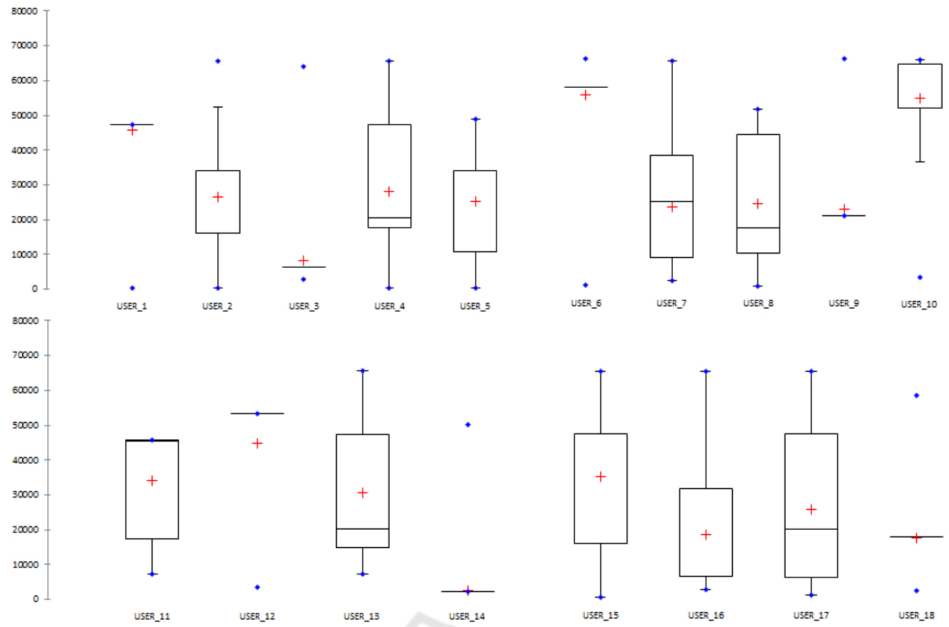

Figure 5: Box plots related to the CID feature for each user involved in the experiment.

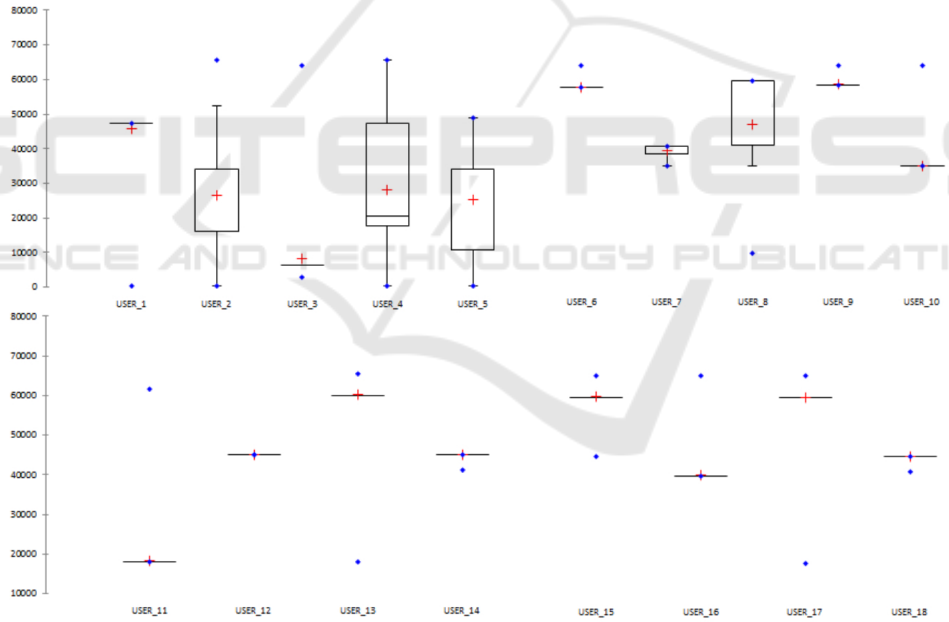

Figure 6: Box plots related to the LAC feature for each user involved in the experiment.

is the number of the features used in training phase ($1 \leq y \leq 6$).

For the learning phase, we use a k-fold cross-validation: the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset for testing the model, while the remaining k-1 subsets of the original dataset are used as training data. We repeated the process for k=10 times; each one of the k subsets has been used once as the validation dataset. To obtain a single estimate, we computed the average of the k results from the folds.

We evaluated the effectiveness of the classification method with the following procedure:

1. build a training set $T \subset D$;
2. build a testing set T' = D÷T;
3. run the training phase on T;
4. apply the learned classifier to each element of T'.

We performed a 10-fold cross validation: we re-

peated the four steps 10 times varying the composition of T (and hence of T').

We classify using three different sets of features:

- *S1*: orientation features;
- *S2*: orientation and touch features;
- *S3*: orientation, touch and cell-id features.

The aim of the classification with the *S1* feature set is to achieve a behavioral pattern in order to succeed in identifying the owner or an impostor considering the device orientation. The *S1* features set is composed by: *pitch*, *roll* and *azimuth* that represents an orientation event.

The second feature set, i.e. *S2*, adds to the *S1* feature set the touch feature, i.e. the touch-gap.

The third features set, i.e. *S3*, adds to *S2* features set the cell-event characterized by *Cell-Id* and *Location-Area* of a precise cell tower.

Each classification was performed using 20% of the dataset as training dataset and 80% as testing dataset.

We defined $C_{u,s}$ as the set of the classifications we performed, where u identifies the user ($1 \leq u \leq 18$) and s represents the features set used in the classification ($s = \{S1, S2, S3\}$).

For sake of clarity we explain with an example the method we adopted: when we perform $C_{2,1}$ classification, we label the traces related to the user #2 as owner traces, and the traces of the other user as impostor using the S1 features set (i.e., just the orientation features) for classification.

Table 2 shows the results obtained with this procedure using the *S1* features set.

The orientation features are too weak for identifying the owner. As a matter of fact the greatest precision obtained is 0.979, but most values are smaller than 0.96. Additionally, FAR and FRR show very high values, i.e. around 0.1. By considering orientation and touch features together, the performances do not improve significantly, as Table 3 shows.

Performances improve when we accomplish the classification with all the features grouped together, as shown in Table 4. For most users, precision and recall are over 99%.

Table 5 shows the average results obtained using the *S1, S2 and S3* features set, in order to facilitate the comparison among the different features.

We obtain the following average values when classifying the behavioral traces by using the RandomForest classification algorithm:

- a precision of 0.910 and a recall of 0.859 using the *S1* feature set;
- a precision of 0.896 and a recall of 0.895 using the *S2* feature set;

Table 2: Classification Results: Precision, Recall, ROC Area, FAR and FRR for classifying Owner and Impostor traces for each user involved in the experiment, computed for *S1* feature set with the J48 and RandomForest(RF) algorithms.

User	Alg.	Precision	Recall	ROC	FAR	FRR
1	J48	0.873	0.892	0.903	0.136	0.117
	RF	0.906	0.910	0.970	0.104	0.084
2	J48	0.776	0.740	0.824	0.230	0.2476
	RF	0.854	0.880	0.935	0.165	0.128
3	J48	0.835	0.820	0.886	0.194	0.139
	RF	0.901	0.910	0.964	0.120	0.079
4	J48	0.835	0.790	0.882	0.154	0.175
	RF	0.879	0.820	0.947	0.130	0.112
5	J48	0.871	0.840	0.908	0.100	0.162
	RF	0.908	0.920	0.964	0.097	0.086
6	J48	0.907	0.910	0.938	0.083	0.104
	RF	0.946	0.930	0.983	0.053	0.056
7	J48	0.774	0.780	0.837	0.284	0.175
	RF	0.872	0.770	0.953	0.154	0.104
8	J48	0.814	0.830	0.856	0.222	0.154
	RF	0.857	0.840	0.938	0.186	0.106
9	J48	0.885	0.880	0.921	0.132	0.097
	RF	0.932	0.950	0.972	0.080	0.056
10	J48	0.939	0.940	0.956	0.073	0.060
	RF	0.961	0.978	0.991	0.0041	0.036
11	J48	0.896	0.880	0.922	0.131	0.079
	RF	0.929	0.916	0.981	0.089	0.055
12	J48	0.823	0.840	0.868	0.224	0.136
	RF	0.979	0.972	0.954	0.152	0.091
13	J48	0.924	0.934	0.946	0.089	0.064
	RF	0.955	0.961	0.989	0.053	0.034
14	J48	0.882	0.800	0.917	0.121	0.115
	RF	0.930	0.918	0.978	0.076	0.0063
15	J48	0.885	0.840	0.926	0.122	0.109
	RF	0.936	0.922	0.982	0.073	0.056
16	J48	0.859	0.870	0.903	0.157	0.126
	RF	0.915	0.900	0.974	0.099	0.070
17	J48	0.862	0.853	0.896	0.161	0.117
	RF	0.915	0.899	0.974	0.099	0.071
18	J48	0.873	0.862	0.917	0.129	0.125
	RF	0.921	0.934	0.973	0.089	0.071

- a precision of 0.995 and a recall of 0.995 using the *S3* feature set.

We notice that the precision diminishes when moving from *S1* to *S2*, although the latter includes more features than *S1*. The best feature set in discriminating between impostor and owner traces is the *S3* feature set.

In order to measure the performance of our method, we report in Table 6 the average time employed for the classification task. The machine used was an Intel Core i5 desktop with 4 gigabyte RAM, equipped with Linux Mint 15.

The J48 algorithm is faster in classification task than RandomForest, anyway all the classifications employ less than a second.

Table 3: Classification Results: Precision, Recall, ROC Area, FAR and FRR for classifying Owner and Impostor traces for each user involved in the experiment, computed for *S2* features set with the J48 and RandomForest(RF) algorithms.

User	Alg.	Precision	Recall	ROC	FAR	FRR
1	J48	0.876	0.869	0.899	0.134	0.114
	RF	0.902	0.909	0.968	0.111	0.086
2	J48	0.776	0.765	0.832	0.234	0.234
	RF	0.825	0.811	0.911	0.185	0.166
3	J48	0.832	0.839	0.877	0.191	0.146
	RF	0.880	0.842	0.950	0.143	0.098
4	J48	0.840	0.856	0.895	0.180	0.142
	RF	0.877	0.893	0.948	0.140	0.108
5	J48	0.874	0.872	0.911	0.120	0.133
	RF	0.908	0.910	0.970	0.099	0.084
6	J48	0.896	0.905	0.928	0.265	0.111
	RF	0.926	0.938	0.975	0.065	0.084
7	J48	0.785	0.800	0.837	0.265	0.169
	RF	0.850	0.877	0.933	0.195	0.110
8	J48	0.881	0.897	0.917	0.162	0.082
	RF	0.903	0.916	0.956	0.142	0.057
9	J48	0.882	0.891	0.918	0.139	0.098
	RF	0.927	0.919	0.970	0.087	0.060
10	J48	0.925	0.918	0.956	0.091	0.060
	RF	0.949	0.954	0.988	0.059	0.045
11	J48	0.862	0.879	0.896	0.191	0.092
	RF	0.900	0.904	0.964	0.137	0.071
12	J48	0.820	0.841	0.866	0.228	0.138
	RF	0.872	0.889	0.946	0.163	0.093
13	J48	0.917	0.908	0.939	0.106	0.061
	RF	0.942	0.938	0.985	0.070	0.047
14	J48	0.840	0.890	0.890	0.156	0.165
	RF	0.889	0.901	0.957	0.118	0.105
15	J48	0.880	0.897	0.918	0.129	0.112
	RF	0.919	0.924	0.975	0.096	0.067
16	J48	0.850	0.872	0.892	0.167	0.135
	RF	0.897	0.937	0.964	0.124	0.085
17	J48	0.811	0.824	0.873	0.208	0.170
	RF	0.862	0.877	0.941	0.173	0.106
18	J48	0.853	0.849	0.901	0.165	0.129
	RF	0.893	0.927	0.96	0.122	0.093

Table 4: Classification Results: Precision, Recall, ROC Area, FAR and FRR for classifying Owner and Impostor traces for each user involved in the experiment, computed for *S3* features set with the J48 and RandomForest(RF) algorithms.

User	Alg.	Precision	Recall	ROC	FAR	FRR
1	J48	0.992	0.997	0.994	0.011	0.006
	RF	0.994	0.999	0.999	0.007	0.006
2	J48	0.995	0.992	0.995	0.007	0.002
	RF	0.996	0.997	1.000	0.007	0.001
3	J48	0.999	0.999	0.999	0.001	0.001
	RF	0.999	0.999	0.999	0.001	0.001
4	J48	0.977	0.969	0.983	0.025	0.021
	RF	0.981	0.988	0.998	0.021	0.017
5	J48	0.993	0.996	0.996	0.013	0.001
	RF	0.992	0.995	0.995	0.017	0.001
6	J48	0.995	0.990	0.998	0.005	0.005
	RF	0.996	0.996	1.000	0.004	0.004
7	J48	0.996	0.995	0.997	0.005	0.004
	RF	0.996	0.996	1.000	0.006	0.002
8	J48	0.992	0.994	0.997	0.014	0.002
	RF	0.903	0.909	0.956	0.142	0.057
9	J48	0.997	0.998	0.997	0.005	0.001
	RF	0.999	0.999	1.000	0.082	0.000
10	J48	0.998	0.998	0.999	0.002	0.001
	RF	0.999	0.999	1.000	0.010	0.001
11	J48	0.999	0.999	0.999	0.002	0.001
	RF	0.999	0.999	0.999	0.002	0.001
12	J48	1.000	1.000	1.000	0.000	0.000
	RF	1.000	1.000	1.000	0.000	0.000
13	J48	0.985	0.989	0.990	0.021	0.008
	RF	0.989	0.987	0.999	0.017	0.004
14	J48	0.999	0.999	1.000	0.001	0.001
	RF	0.999	0.999	0.999	0.001	0.001
15	J48	0.987	0.988	0.992	0.016	0.010
	RF	0.992	0.994	1.000	0.013	0.004
16	J48	0.997	0.998	0.999	0.004	0.001
	RF	0.999	0.999	1.000	0.003	0.001
17	J48	0.991	0.990	0.995	0.013	0.06
	RF	0.994	0.993	1.000	0.007	0.004
18	J48	0.997	0.998	0.998	0.030	0.030
	RF	0.997	0.998	1.000	0.003	0.002

Table 5: Classification Results: average value for Precision, Recall, ROC Area, FAR and FRR for classifying Owner and Impostor traces, computed for *S1*, *S2* and *S3* features set with the J48 and RandomForest(RF) algorithms.

Feat.	Alg.	Precision	Recall	ROC	FAR	FRR
S1	J48	*0.861*	*0.859*	*0.900*	*0.152*	*0.128*
	RF	*0.910*	*0.859*	*0.900*	*0.098*	*0.072*
S2	J48	*0.855*	*0.854*	*0.897*	*0.165*	*0.127*
	RF	*0.896*	*0.895*	*0.959*	*0.124*	*0.087*
S3	J48	*0.994*	*0.994*	*0.996*	*0.008*	*0.004*
	RF	*0.995*	*0.995*	*0.999*	*0.007*	*0.003*

4 RELATED WORK

There are two categories of biometrics for user identification: physiological (such as fingerprints, facial features) and behavioral biometrics (such as speaking, typing, walking). Physiological biometrics usually requires special recognition devices. Some physiological biometrics, like face and voice can be detected by smartphones, but usually entail expensive computation and energy costs and have a high error rate. For example, in Koreman et al. (Koreman et al., 2006), the (equal error rates) EER for face recognition is around 28% and for voice is around 5%. Keystroke is a popular behavioral biometric: Joyce and Gupta (Joyce and Gupta, 1990) presented a survey on the large body of literature on authentication with keystroke dynamics.

Some researchers proposed authentication token based mechanisms to identify legal users,e.g., wire-

Table 6: The performance evaluation for classifying the *S1*, *S2* and *S3* feature set, with the J48 and RandomForest(RF) algorithms.

Feature Set	Algorithm	Time
S1	J48	0.47 s
	RF	0.56 s
S2	J48	0.51 s
	RF	0.72 s
S3	J48	0.68 s
	RF	0.84 s

less token (Nicholson et al., 2006). However, they require additional hardware and are not convenient for daily smartphone usage. On the smart phones with touch screens, PINs, pass-phrases, and secret drawn gestures are the commonly used authentication methods (Dunphy et al., 2010).

Recently, there is a growing body of work that uses the features of touch behavior to verify users. Several existing approaches have used the touch behavior biometrics for various security purposes. De Luca et al. (De Luca et al., 2012) propose a password application, by which the user draws a stroke on the touch screen as a input password. Pressure, coordinates, size, speed and time of the stroke are used to identify the valid user. Overall the accuracy of this method is 77% with a 19% FRR and 21% FAR. Zheng et al. (Zheng et al., 2014) use four features (acceleration, pressure, size, and time) to distinguish the true owner and impostor to enhance the security of passcode. Their identification system achieves 3.65% EER. A user enters a password by tapping several times on a touch surface with one or more fingers. PassChord failed to authenticate for 16.3% of the time. There are some other works addressing the user identification issue with touch features, e.g., Seo et al. (Seo et al., 2012). With pure touch data, there may be a high error rate.

Riva et al. (Riva et al., 2012), rather than exploring a new authentication scheme, address the problem of deciding when to surface authentication and for which applications. Their approach combines multiple signals (biometric, continuity, possession) to determine a level of confidence in a users authenticity. They built a prototype running on modern phones to demonstrate progressive authentication and used it in a lab study with nine users. Their system is able to reduce the number of required authentications by 42%.

Kwapisz et al. (Kwapisz et al., 2010) use accelerometer data to identify or authenticate cell phone users. They aggregate the raw time-series accelerometer data into examples, since most classification algorithms cannot operate directly on time series data. Each of these examples is associated with a specific cell phone user, thus forming labeled training data. For user authentication they build separate models for each user in order to determine whether an example came from that user or from someone else, obtaining an accuracy value equal to 87.6% using the J48 classification algorithm of identification of 10-seconds examples.

Frank et al. (Frank et al., 2013) propose a set of 30 behavioral touch features that can be extracted from raw touchscreen logs and demonstrate that different users populate distinct subspaces of this feature space. They collected touch data from users interacting with a smart phone using basic navigation maneuvers, i.e., up-down and left-right scrolling. They propose a classification framework that learns the touch behavior of a user during an enrollment phase and is able to accept or reject the current user by monitoring interaction with the touch screen. Their classifier achieves a median equal error rate of 0% for intra-session authentication, 2%-3% for inter-session authentication and below 4% when the authentication test was carried out one week after the enrollment phase.

Killourhy et al. (Killourhy and Maxion, 2009) collect a keystroke-dynamics data set, in order to measure the performance of a range of detectors so that the results can be compared soundly. They collected data from 51 subjects typing 400 passwords each, evaluating 14 detectors from the keystroke-dynamics and pattern-recognition literature. The three top-performing detectors achieve equal-error rates between 9.6% and 10.2%.

Bo et al. (Bo et al., 2014) build a touch-based biometrics model of the owner by extracting some principle features, and then verify whether the current user is the owner or guest/attacker. When using the smartphone, some unique operating dynamics of the user are detected and learnt by collecting the sensor data and touch events silently. When users are mobile, the micro-movement of mobile devices caused by touch is suppressed by that due to the large scale user-movement which will render the touch-based biometrics ineffective. To address this, we integrate a movement-based biometrics for each user with previous touch-based biometrics. We conduct extensive evaluations of our approaches on the Android smartphone, showing that the user identification accuracy is over 99%.

Murmuria et al. (Murmuria et al., 2015) propose the continuous user monitoring using a machine learning based approach comprising of an ensemble of three distinct modalities: power consumption, touch gestures, and physical movement. They evaluated the method using a dataset retrieved by monitoring 73 volunteer participants using the same device (Google Nexus 5) with Android 4.4.4 version on

board. In the evaluation, they obtain an equal error rate between 6.1% and 6.9% for 59 selected users.

Gascon et al. (Gascon et al., 2014) analyze typing motion behavior captured while the user is entering text. The authors developed a software keyboard application for the Android OS that stores all sensor data for further learning and evaluation. In a field study with more than 300 participants, they reached a false positive rate of 1% and a total positive rate of 92%.

Clarke et al. (Clarke and Mekala, 2006) discuss the application of biometrics to a mobile device in a transparent and continuous fashion and the subsequent advantages and disadvantages that are in contention with various biometric techniques. In order to facilitate the use of signature recognition transparently, their method must verify users based upon written words and not signatures. From the experiment conducted they were found that current signature recognition systems could indeed perform successful authentication on written words. Based upon 20 participants an average FAR and FRR of 0% and 1.2% respectively were experienced across 8 common words.

Brocardo and Traore (Brocardo and Traore, 2014) introduce a very novel approach for continuous authentication which is based on micro messages, by extracting lexical, syntatic, and application specific features. This technique should guarantee a shorter authentication delay, essential for reducing the vulnerability window of the device. This system obtained an EER of 9.18%, but the main limitation is that it can be applied only to messages, so it cannot be used if the user does not write messages.

Wu et al. (Wu et al., 2015) propose a method profiling behavioral biometrics from keystrokes and gestures, that also acquires the specific properties of a one-touch motion during the users interaction with the smartphone. The authors demonstrate that the manner by which a user uses the touchscreen that is, the specific location touched on the screen, the drift from when a finger moves up and down, the area touched, and the pressure used reflects unique physical and behavioral biometrics.

Piuri et al. (Piuri and Scotti, 2008) presented a set of techniques to extract the fingerprint ridge structure by image processing in images acquired by low-cost cameras and webcams. The approach allows the use of webcams and low-cost cameras as interoperable devices for fingerprint biometrics. Results show that even in normal illumination conditions and by using sensors of about 1Mpixel (or above), the ridge structure can be effectively extracted. The limited resolution of current CCDs can produces several artifacts in the extracted ridge structure, but experiments have shown that the presence of artifacts can be reduced by using higher image resolutions.

Kotroupolos and Samaras (Kotropoulos and Samaras, 2014) obtain the user's profile from recorded speech signals. They obtained as top accuracy 97.6% which is smaller than the one obtained with our experiment.

The main novelty of our method with respect to the existing literature stands in the set of features selected and in the significant performance obtained.

5 CONCLUSION AND FUTURE WORK

Current authentication mechanisms in mobile environment use as mechanisms to identify the device owner pins at login time and textual and graphical passwords. These mechanisms do not offer any protection against access post login. In this paper we propose a method to continuously and silently authenticate the user by extracting a set of features able to characterize the user behavior. The method is silent because it builds the profile of the user's behavior while the user is using the mobile phone, without requiring any further entry by the user, like voice or facial image. Results are significantly better than those reported in literature: we obtain a precision and a recall greater than 0.99 collecting data from 18 volunteer participants in a 10-day time window. Additionally we measured a FAR of 0.7 % and a FRR of 0.3%.

We plan to improve our experiment in several ways. With respect to data collection, we intend to increase the number of experimental subjects involved, and to collect further features per user, as well as the frequent pattern of applications used by the user.

REFERENCES

Ahmed, A. A. E. and Traore, I. (2005). Anomaly intrusion detection based on biometrics. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 452–453. IEEE.

Akula, S. and Devisetty, V. (2004). Image based registration and authentication system. In *Proceedings of Midwest Instruction and Computing Symposium*, volume 4.

Bailey, K. O., Okolica, J. S., and Peterson, G. L. (2014). User identification and authentication using multimodal behavioral biometrics. *Computers & Security*, 43:77–89.

Bhattacharyya, D., Ranjan, R., Farkhod Alisherov, A., and Choi, M. (2009). Biometric authentication: A review. *International Journal of u-and e-Service, Science and Technology*, 2(3):13–28.

Bo, C., Zhang, L., Jung, T., Han, J., Li, X.-Y., and Wang, Y. (2014). Continuous user identification via touch and movement behavioral biometrics. In *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*, pages 1–8. IEEE.

Brocardo, M. L. and Traore, I. (2014). Continuous authentication using micro-messages. In *Privacy, Security and Trust (PST), 2014*, pages 179–188. IEEE.

Brown, M. and Rogers, S. J. (1993). User identification via keystroke characteristics of typed names using neural networks. *International Journal of Man-Machine Studies*, 39(6):999–1014.

Clarke, N. and Mekala, A. (2006). Transparent handwriting verification for mobile devices. In *Proceedings of the Sixth International Network Conference (INC 2006), Plymouth, UK*, pages 11–14. Citeseer.

Davis, D., Monrose, F., and Reiter, M. K. (2004). On user choice in graphical password schemes. In *USENIX Security Symposium*, volume 13, pages 11–11.

De Luca, A., Hang, A., Brudy, F., Lindner, C., and Hussmann, H. (2012). Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI*, pages 987–996. ACM.

Dhamija, R. and Perrig, A. (2000). D ej a vu: A user study using images for authentication.

Dunphy, P., Heiner, A. P., and Asokan, N. (2010). A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 3. ACM.

Frank, M., Biedert, R., Ma, E.-D., Martinovic, I., and Song, D. (2013). Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *Information Forensics and Security, IEEE Transactions on*, 8(1):136–148.

Gamboa, H. and Fred, A. (2004). A behavioral biometric system based on human-computer interaction. In *Defense and Security*, pages 381–392. International Society for Optics and Photonics.

Gascon, H., Uellenbeck, S., Wolf, C., and Rieck, K. (2014). Continuous authentication on mobile devices by analysis of typing motion behavior. In *Sicherheit*, pages 1–12.

Joyce, R. and Gupta, G. (1990). Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176.

Killourhy, K. S. and Maxion, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134. IEEE.

Koreman, J., Morris, A., Wu, D., Jassim, S., Sellahewa, H., Ehlers, J., Chollet, G., Aversano, G., Bredin, H., Garcia-Salicetti, S., et al. (2006). Multi-modal biometric authentication on the securephone pda. In *Proceedings of the MMUA workshop on Multimodal User Authentication*.

Kotropoulos, C. and Samaras, S. (2014). Mobile phone identification using recorded speech signals. In *Digital Signal Processing (DSP), 2014 19th International Conference on*, pages 586–591. IEEE.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2010). Cell phone-based biometric identification. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7. IEEE.

Monrose, F. and Rubin, A. (1997). Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56. ACM.

Murmuria, R., Stavrou, A., Barbará, D., and Fleck, D. (2015). Continuous authentication on mobile devices using power consumption, touch gestures and physical movement of users. In *Research in Attacks, Intrusions, and Defenses*, pages 405–424. Springer.

Nicholson, A. J., Corner, M. D., and Noble, B. D. (2006). Mobile device security using transient authentication. *Mobile Computing, IEEE Transactions on*, 5(11):1489–1502.

Piuri, V. and Scotti, F. (2008). Fingerprint biometrics via low-cost sensors and webcams. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6. IEEE.

Riva, O., Qin, C., Strauss, K., and Lymberopoulos, D. (2012). Progressive authentication: deciding when to authenticate on mobile phones. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 301–316.

Sae-Bae, N. and Memon, N. (2013). A simple and effective method for online signature verification. In *BIOSIG*, pages 1–12. IEEE.

Seo, H., Kim, E., and Kim, H. K. (2012). A novel biometric identification based on a users input pattern analysis for intelligent mobile devices. *International Journal of Advanced Robotic Systems*, 9:1–10.

Shen, C., Cai, Z., Guan, X., and Cai, J. (2010). A hypo-optimum feature selection strategy for mouse dynamics in continuous identity authentication and monitoring. In *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on*, pages 349–353. IEEE.

Shepherd, S. (1995). Continuous authentication by analysis of keyboard typing characteristics. In *Security and Detection, 1995., European Convention on*, pages 111–114. IET.

Wu, J.-S., Lin, W.-C., Lin, C.-T., and Wei, T.-E. (2015). Smartphone continuous authentication based on keystroke and gesture profiling. In *Security Technology (ICCST), 2015 International Carnahan Conference on*, pages 191–197. IEEE.

Zheng, N., Bai, K., Huang, H., and Wang, H. (2014). You are how you touch: User verification on smartphones via tapping behaviors. In *ICNP*, pages 221–232. IEEE.