Please Hold On: Unobtrusive User Authentication using Smartphone's built-in Sensors

Attaullah Buriro*
University of Trento
Trento, Italy
attaullah.buriro@unitn.it

Bruno Crispo
University of Trento, & DistriNet, KULeuven
Trento, Italy & Leuven, Belgium
bruno.crispo@cs.kuleuven.be

Yury Zhauniarovich
Qatar Computing Research Institute(QCRI), HBKU
Doha, Qatar
yzhauniarovich@qf.org.qa

Abstract

Smartphones provide anytime-anywhere communications and are being increasingly used for a variety of purposes, e.g, sending email, performing online transactions, connecting with friends and acquaintances over social networks. As a result, a considerable amount of sensitive personal information is often generated and stored on smartphones. Thus, smartphone users may face financial as well as sentimental consequences if such information fall in the wrong hands. To address this problem all smartphones provide some form of user authentication, that is the process of verifying the user's identity. Existing authentication mechanisms, such as using 4-digit passcodes or graphical patterns, suffer from multiple limitations - they are neither highly secure nor easy to input. As a results, recent studies found that most smartphone's users do not use any authentication mechanism at all. In this paper, we present a fully unobtrusive user authentication scheme based on micro-movements of the user's hand(s) after the user unlocks her smartphone. The proposed scheme collects data from multiple 3-dimensional smartphone sensors in the background for a specific period of time and profiles a user based on the collected hand(s) movement patterns. Subsequently, the system matches the query pattern with the pre-stored patterns to authenticate the smartphone owner. Our system achieved a True Acceptance Rate (TAR) of 96% at an Equal Error Rate (EER) of 4%, on a dataset of 31 qualified volunteers (53, in total), using Random Forest (RF) classifier. Our scheme can be used as a primary authentication mechanism or can be used as a secondary authentication scheme in conjunction with any of the existing authentication schemes, e.g., passcodes, to improve their security.

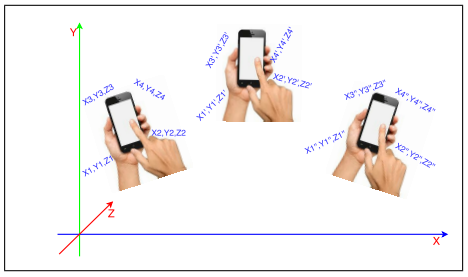


Figure 1: Different phone positions during the user interactions [4].

1. Introduction

Smartphones are the most widely used personal devices [3] and are forecast to replace laptops and desktops[1]. Besides their traditional use for voice communication, smartphones have been used for a variety of other purposes, such as performing online transactions, social networking, mobile commerce, etc. Consequently, an increasing amount of privacy sensitive personal information is generated and stored on smartphones. Hence, it is of paramount importance to restrict smartphone access only to the legitimate user.

Most commonly used knowledge-based authentication mechanisms (i.e., PIN, passwords, etc) are neither highly secure [20, 2] nor very usable [9]. It is estimated that an

*Attaullah Buriro did this work during his internship period at DistriNet, KULeuven, Belgium.

[1]https://www.wired.com/2015/02/smartphone-only-computer/

average user takes upto $4.7s$ to unlock her smartphone using the PIN. Repeating this operation more than an hundred of times a day can be quite annoying [1]. Biometrics-based smartphone unlock solutions, e.g., using face, iris, or fingerprint recognition have recently been introduced, however, they still face unsolved security[2] and usability issues [8]. Almost 47% of fingerprint and 36% face recognition users mentioned low usability as the primary reason for abandoning the use of these technologies [8].

More recently, behavioral-biometrics-based authentication schemes have attracted significant attention in the domain of smartphones. User authentication schemes based on call placing/answering [7, 5], gait [24], touch operations [26] and keystrokes [6] analysis have been proposed. Some advantages of these schemes are: minimal user interaction, unobtrusive data collection, no additional hardware required. An important disadvantage is the obtained accuracy. Usually these schemes do not achieve a TAR higher than 95% with EER from 5 to 10% in controlled and supervised testing environments.

This paper presents a novel and fully unobtrusive authentication scheme based on the profiling of the hand(s) micro-movements when a user unlocks her smartphone (see Figure 1). Existing user-profiling methods are not completely unobtrusive since user profiling is done by means of an additional activity (i.e. typing a sequence on the touchscreen) added for the sole purpose of user profiling. Our system, piggyback the profiling in the native unlocking mechanisms of the phone without requiring any additional action. Technically speaking, this is done by starting profiling the users after sensing their presence through Android *USER_PRESENT* broadcast receiver. The *USER_PRESENT* broadcast receiver is fired at the moment the user either enters her credentials or performs the slide-to-unlock gesture to unlock her smartphone. The proposed system uses the *unprivileged sensors*[3] on a smartphone for a short period of time after an unlock event occurs. Using state-of-the-art machine learning classifiers, our scheme decides if the smartphone is unlocked by the owner or by the impostor. In the case of an impostor, access is denied and the smartphone owner is notified. The proposed scheme neither requires any token nor needs the user to remember any specific gesture/action for unlocking. Thus, it is completely unobtrusive and usable both as a primary and secondary smartphone user authentication method. We validated our method on a realistic setting with unsupervised testers. The results confirm the effectiveness of the proposed approach. We report a TAR of 96% at EER of 4%.

The main contributions of the paper are listed below:

- A novel approach for fully unobtrusive user authentication using the existing smartphone's sensors and avoiding any additional hardware.
- Dataset of data collected from multiple sensors by 53 users, in total. Our dataset includes multiple smartphones and users tested our system in an uncontrolled fashion.
- Extensive experimental evaluation to analyse the accuracy of the proposed scheme and its feasibility.

[2] http://www.ibtimes.co.uk/iphone-6-touch-id-fingerprint-scanner-hacked-days-after-launch-1466843

[3] Unprivileged sensors can be used without any explicit user permission.

2. Related Work

Modern smartphones are equipped with highly sensitive built-in 3-dimensional sensors such as accelerometer, gyroscope, orientation, etc. Researchers have used these sensors to profile users for both static and continuous authentication. In static authentication (one-shot login), users are recognized based on pre-defined tasks, e.g., walking patterns [16], general phone-movement [13, 28, 21], special phone-movement (while entering PIN, password) [6], and lift-behavior (how they move their phone to place or answer a call [7, 5] and profiled gesture models [28]. The sensory data collected during these pre-defined tasks are then analyzed to verify the user's identity. Continuous authentication schemes typically collect the sensory data continuously to verify the user's identity throughout the whole session. In this section, we review sensory-data-based authentication schemes for smartphone user authentication proposed over the years.

Shi et al. [21] present a multi-sensor approach to passively identify a genuine user. Their system uses accelerometer, touch screen, voice and location data for user authentication. They report $\sim 97\%$ TAR, using the Naive Bayes classifier, on a dataset of 7 users (three females and four males). Li et al., [13] explored the utility of three different sensors: accelerometer, orientation, and compass in addition to the touch gestures for continuous user authentication. Their method profiles finger movements using classical touch-based features and interprets the sensed data as different gestures. An SVM classifier is than trained with gestures to perform authentication tasks. Accuracy of 95.78% is reported on a database of 75 users.

Zhu et al. [28] propose a mobile framework model *Sensec* based on accelerometer, orientation, gyroscope, and magnetometer to construct a user gesture profile. The model then continuously computes the sureness score to authenticate the user. By concatenating *X*, *Y*, *Z* values from the aforementioned sensors, a valid user is identified with 75% accuracy and an adversary with an accuracy of 71.3% (with 13.1% FAR) from a set of 20 users. However, the study requires a user to follow a fixed protocol and collects data for the entire user interaction session. The method proposed here is different since it does not require any specific protocol to be followed. Furthermore, data is collected only once

in the entire session (without requiring any explicit user interaction).

Conti et al. [7] exploit accelerometer and orientation sensor readings collected during call placing/answering, to profile the genuine user. Their study reports a FAR of 4.44% at a FRR of 9.33% on a dataset of 10 users with Dynamic Time Warping (DTW) classifier. The study by Buriro et al. [5] extends it to a tri-modal system which involves arm movement, finger swiping and voice recognition. 10.28% FAR at 3.93% FRR is reported on a dataset of 26 users. An important related work, i.e., HMOG by Sitova et al. [23] leverages *Hand Movement, Orientation, and Grasp* to continuously authenticate smartphone users. It transparently collects data from the accelerometer, gyroscope, and magnetometer when a user grasps, holds and taps on the smartphone screen. On a dataset of 100 test subjects (53 males and 47 females), HMOG achieves lowest EER of 6.92% in *walking* state with SVM classifier. Our method does not require any typing, keystrokes or grasp. Instead the data is collected transparently after an unlock event occurs (as a result of either slide-to-unlock, entering PIN or password, etc.).

Google project - ABACUS, built a large dataset containing 27.62 TB of smartphone signals on Nexus 5 smartphones from 1500 users over a period of six months [17]. Data was obtained from multiple sensors, namely, camera, touchscreen, keyboard, accelerometer, magnetometer, gyroscope, light sensor, GPS, Bluetooth, Wi-Fi and application usage. Data was recorded for the entire user interaction session - from one smartphone unlock to the next time it is locked. Using optimized shift-invariant Dense Convolutional Mechanism (DCWRNN) an EER of 8.82% (per session) and 15.84% (per device) was reported. Here an EER of 8.82% means that 91.18% of the times, the correct user was holding and moving the phone, not necessarily interacting with it. In our case, we identify user after her interaction with the device. Upal et. al., [15], collected smartphone signals from 48 volunteers on Nexus 5 smartphone, over a period of two months. They collected data from the camera, touchscreen, gyroscope, accelerometer, magnetometer, light sensor, GPS, Bluetooth, WiFi, proximity sensor, temperature sensor and pressure sensor. Apart from face detection and recognition results, they reported swipe-based authentication results. Among multiple classifiers, the Random Forest classifier achieved lowest EER of 22.1%. However, both datasets have not been made available to the research community yet so it is difficult to compare to these solutions.

Most sensor-based authentication solutions listed above utilize the sensor(s) available in smartphones. They collect sensory data associated with either finger movements, user tappings or associated with the particular motion (e.g., call placing). Furthermore, most solutions are based on the data collected in laboratory settings. On the other hand, our method is different in the following ways:

- It is fully unobtrusive. It does not require any permission, participation, or cooperation from a user. Each authentication step is performed, transparently, in the background.
- Data was collected in totally uncontrolled manner.
- Our method utilizes all the 3-dimensional sensors available on the smartphones.
- Our scheme initiates all the sensors after receiving the user presence notification from the OS associated with the *USER_PRESENT* broadcast receiver. Therefore, it can complement the existing one-shot login methods and becomes more useful, especially, for those users (e.g., slide-unlock users) who do not want to invoke any explicit authentication mechanisms on their smartphones.

3. Proposed Method

This section presents the threat model, the intuition behind our solution and the overall approach.

3.1. Threat Model

We consider the situation where an attacker is already in possession of the smartphone. An attacker can be an unknown person, e.g., traveling with the real user in a bus or train and getting the smartphone access. Alternatively, an attacker could be the victim's friend, family member or a co-worker attempting to access the smartphone.

3.2. Intuition Assessment

It has been reported in previous studies [6, 4, 23, 7, 17] that each user holds, interacts and moves her phone in a unique way. This uniqueness of movement pattern increases the authentication accuracy on the one side and makes challenging for impostors to generate exactly the same movement patterns.

3.3. Our Approach

The proposed method is based on the idea of utilizing a user's hand micro-movements after she unlocks her phone using an authentication method, e.g., PIN, slide-to-unlock, etc. In either case, when the user unlocks her smartphone, the Android OS generates a specific broadcast event *USER_PRESENT*. The said event is generated only once per session (when the user unlocks her smartphone). Similar events[4] are generated also in other mobile operating systems, e.g., iOS. Thus, the proposed method can

[4]e.g., `PhoneApplicationFrame.Unobscured` event in Windows Phone OS, or `com.apple.springboard.lockstate` event in Apple iOS.

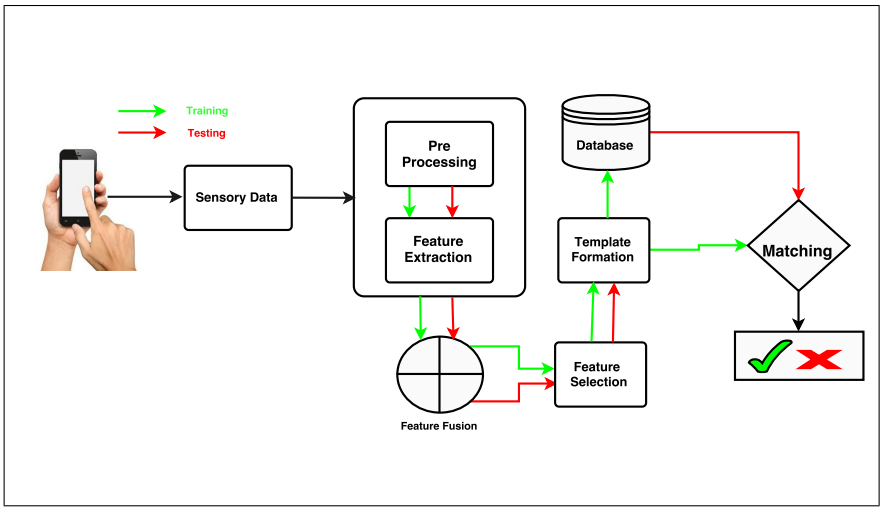


Figure 2: Flowchart of the proposed method.

be implemented also for other popular mobile operating systems.

Our idea is based on profiling the user's hand micro-movements for a short period of time (at most 10 sec). The rationale behind choosing this time duration is the following: (i) it was empirically determined that this time is sufficient enough for pattern discrimination, and (ii) this duration is too short for an adversary to debug the device [25]. The collected data is pre-processed and relevant features are extracted. A final template is constructed by concatenating all extracted features and is fed to the classifier for training or for testing (see Figure 2). If a user during this period is classified as a genuine user, the system will not interrupt the owner's interactions with the smartphone. On the other hand, if the user is classified as an impostor, the system alerts the owner of the phone (e.g., sending an email), and may stealthily isolate the impostor from accessing sensitive functionality [27, 22], or ask for explicit authentication [19, 10]. We restrict ourselves to collecting information from unprivileged sensors. This allows our system to be implemented as a separate authentication service or to be integrated within an implicit authentication framework as the one proposed in [12]. Figure 2 illustrates our proposed approach for user authentication on mobile devices. The sensory data is first pre-processed and the features are extracted. Extracted features are then concatenated together, to make a feature vector, and this feature vector is fed into the feature selection module to find the most productive feature subset for onward user profiling. The selected feature subset is stored in the database for matching afterwards with the query sample to accept or reject the user.

4. Methodology

In this section, we discuss all the steps taken to implement our solution.

4.1. Data Collection

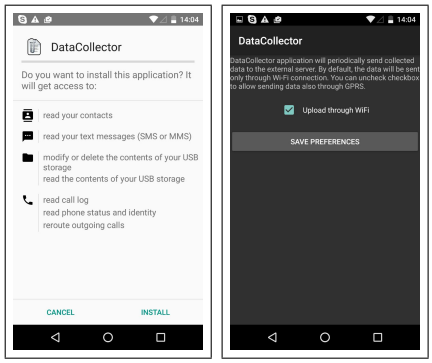
For the purpose of data collection, we developed an Android application called DataCollector which collects the data for the analysis. The application is designed to operate in the background (as a separate service), to emulate the behavior of an authentication application.

In order to collect data for our analysis (supervised learning task), it was necessary to collect user's data during their daily routine of using their smartphone. We set up a web page which explained the purpose, methodology, and other related details of the experiment and a download link where they could get the DataCollector. Moreover, the DataCollector app itself displayed to users all the above-mentioned details of the experiment. Users could install the application after agreeing to a consent form.

DataCollector collects data from multiple sensors, namely, accelerometer, gravity, gyroscope, magnetometer, and orientation. Additionally, we apply two filters to the data from accelerometer, i.e., High Pass Filter (HPF) and Low Pass Filters (LPF)[5] with the parameter $\alpha = 0.5$. Thus, we used 5 physical sensors and 2 extracted accelerometer readings (LPF and HPF). For each built-in sensor and sensory readings, we collect 3-dimensional values denoting the user's motion in a particular dimension, and additionally calculate their magnitude (norm).

Our app gathers information from the sensors with the `SENSOR_DELAY_NORMAL` delay. According to the An-

[5]http://developer.android.com/guide/topics/sensors/sensors_motion.html



(a) App Installer (b) WiFi Notifier

Figure 3: Screen shots of our DataCollector app: Figure 3a shows the application installer and the Figure 3b shows the connectivity manager.

droid documentation[6], for every sensor, data samples are generated at most every $200,000$ microseconds. Information about system events is recorded as soon as they occur. Every measurement is followed by a timestamp using the system call `System.currentTimeMillis()`. Thereafter, collected data are packed into the JavaScript Object Notation (JSON) message and stored as text entries into a file (one file for every sensor). Every two hours our application compresses the collected data to save storage space and sends the encrypted (to ensure data confidentiality) archives to our web server. After each successful transmission attempt, the compressed files on the device are deleted, otherwise, the app keeps retrying.

To ensure participant's privacy, we did not collect any information that can be used to identify a user (e.g., IMEI, IMSI, or phone number). To identify different app instances, DataCollector generates a random unique identifier during the installation. This identifier is later used to label different users on the server. Moreover, our application does not gather any sensitive information, e.g., location, user contacts, etc. To facilitate user participation, DataCollector was developed with the objective to limit the amount of interactions required to configure the app. User involvement is required only during the installation, initial configuration, and for the uninstallation of the app (see Figure 3). Initial configuration only required users to select if data must be transmitted only through WiFi or also using mobile broadband. A total of more than 90GB of raw data were collected.

4.2. Feature Extraction

We use statistical features calculated over the sensor measurements gathered within a specified time interval after the `USER_PRESENT` event. We experimented with time interval of $2, 4, 6, 8, 10$ seconds. From each sensor data, we extracted 7 statistical values namely, Mean, Mean absolute deviation (Mad), Median (Med), unbiased Standard Error of the Mean (Sem), Standard Deviation (std), unbiased Skewness (Skew) and kurtosis (Kurt). Thus, for every sensor there are 28 features listed in Table 1.

No.	Features
1	Mean
2	Mean Absolute Deviation (Mad)
3	Median
4	Unbiased Standard Error of Mean (Sem)
5	Standard Deviation (Std)
6	Unbiased skewness (skew)
7	Kurtosis (Kurt)

Table 1: List of extracted features from all four dimensions of each sensor (28 in total from each sensor).

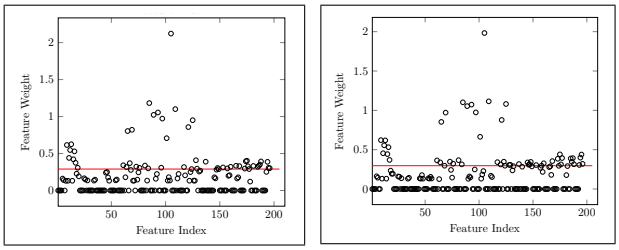
We chose time domain features because their calculation is computationally cheaper compared to the frequency domain ones (due to the expensive Fourier transformation). Moreover, the aim of this paper was to show the feasibility of the approach, so we started from the simplest possible approach to provide a baseline for future improvements of the system.

Normally, extracted features need to be scaled (or normalized depending on the context) before being processed by machine learning algorithms. However, in our case, we skipped this transformation for two reasons. Firstly, the Android system does not provide an Application Programming Interface (API) to find out the minimum and maximum boundaries of sensor measurements. Hence, the scaling operation will require the authentication application to analyze a large amount of historical data in order to detect the feature values boundaries. This demands additional storage space that is limited in mobile environments. Moreover, it is possible that after the training phase, some outliers may appear in our measurements. Scaled using the learned boundaries, these values will still hugely outperform them, thus, influencing a lot the final decision. Secondly, scaling operations require additional computational resources, which are limited in the case of mobile devices, so our system uses raw feature values.

4.3. Feature Subset Selection

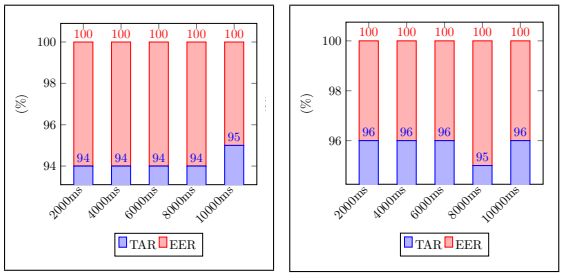
Feature, attribute or variable subset selection is the process of selecting the most productive feature subset (which gives maximum accuracy), from the original feature set. Feature selection is performed mainly for three reasons: firstly, to identify redundant and irrelevant features from the original feature vectors (features below the red line in the Figure 4), secondly, to decrease the computational cost, i.e., processing smaller feature vectors is computationally inexpensive as compared to processing original feature vectors.

[6]http://developer.android.com/guide/topics/sensors/sensors_overview.html



(a) $4000ms$ (b) $6000ms$

Figure 4: Feature Selection for different time periods, i.e., $4000ms$, and $6000ms$. Due to space limitations, we show these figures for 2 time durations.



(a) MLP Classifier (b) RF Classifier

Figure 5: Results of MLP and RF verifiers (averaged over all 31 qualified users) on selected features set.

	2000ms		**4000ms**		**6000ms**		**8000ms**		**10000ms**	
Classifiers	TAR	EER	TAR	EER	TAR	EER	TAR	EER	TAR	EER
BN	0.89	0.11	0.89	0.11	0.89	0.11	0.88	0.12	0.89	0.12
MLP	**0.93**	**0.07**	**0.93**	**0.07**	**0.94**	**0.06**	**0.94**	**0.06**	**0.94**	**0.06**
1NN	0.88	0.12	0.88	0.12	0.89	0.11	0.89	0.11	0.90	0.10
RF	**0.95**	**0.05**	**0.95**	**0.05**	**0.95**	**0.05**	**0.95**	**0.05**	**0.95**	**0.05**

Table 2: Results of different classifiers for different lengths of data acquisition (averaged over all 31 qualified users) with full features.

Finally, smaller feature vector reduces the complexity of the model and hence results in faster classifier learning.

To select the best subset, that is the subset which yields maximum accuracy, out of all 196 available features, we relied on InfoGainAttributeEval[7] - a WEKA implementation for Information Gain (IG) based feature selection. It evaluates the worth of a feature by computing the information gain of that feature with respect to the class. We straight away excluded all the non-contributing features, i.e., having *zero* value (see Figure 4). In addition, to avoid any chances of overfitting, we picked 50 top-gain features (marked above the red line), making them equivalent to the number of samples, for further analysis.

5. Validation

Our experimental validation involves the collection of labeled raw data from multiple 3-dimensional smartphone sensors and then transforming them into the patterns. A pattern here is the horizontal concatenation of all the features of all the sensors (196 before feature subset selection), as discussed in section 4.2. The resulting 50 patterns for each of the 31 users are 196 feature long. Note that we take into account only users with ≥ 50 patterns.

[7]http://weka.sourceforge.net/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html.

5.1. Classification Methods

The choice of the classifier depends on various parameters such as the size of the dataset, training time, simplicity, computational constraints, etc. It is usually impossible to know in advance which classifier fits a dataset better. We used four classification algorithms from the WEKA workbench [11] for user authentication: BayesNET (BN), K-Nearest Neighbor (KNN), Multilayer Perceptron (MLP) and Random Forest (RF). BN and KNN were chosen for their simplicity, fast learning phase and robustness. MLP classifier belongs to the neural network family and was found extremely accurate in related studies. The RF classifier is yet another classifier shown to be very accurate in the previous study [6]. Additionally, RF classifier does not overfit and is extremely quick even if it consumes more memory than a single decision tree.

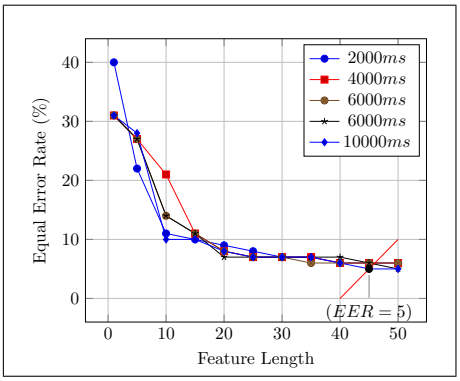
Since we have limited number of user patterns (50 only), our analysis is based on 10-fold cross-validation for all experiments with 10 runs. The setting looks justified because in this way, each available sample is tested and their average is reported.

5.2. Results

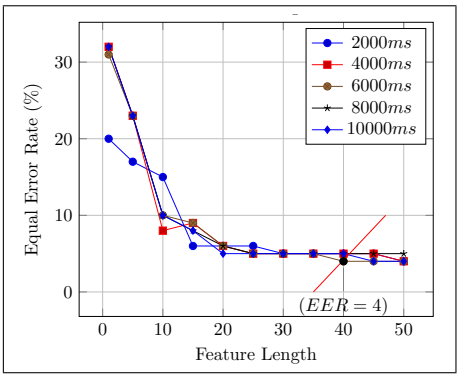
We present the results in terms of TAR and EER. TAR is the fraction of legitimate user attempts correctly classified. EER is the rate at which both false acceptances and false rejections becomes equal.

The results of all of our chosen classifier before the feature selection are shown in the Table 2. We can see that RF and MLP classifier performed best with default parameters (100 trees for RF and 1 hidden layer for MLP) yielding up to 95% and 94%, TAR, respectively. Thus, we take these two classifiers for further analysis. The Figure 5 shows the outcome of both MLP and RF classifiers on the subset of selected features. MLP classifier performed best on $10s$ data yielding 95%, however, RF classifier proved itself consistent on all the available lengths of the dataset.

We evaluated further the different feature lengths in or-



(a) MLP Classifier



(b) RF Classifier

Figure 6: Results in terms of EER for different feature lengths (from selected features).

der to (i) cross check our earlier obtained results, and (ii) observe if the same accuracy can be obtained with even less features (see Figure 6). The best EER of 5% is obtained for $8s$ and $10s$ durations with MLP classifier, however, RF classifier is found consistent with all the durations yielding 4% EER except the $8s$ time interval. It is also worth-mentioning that for $6s$ time duration, we obtained 4% EER with just 40 features. Of course, lower time intervals has to be preferred to long ones, if accuracy is the same because they are faster and reduce battery consumption. So the best option to choose is the interval of $6s$ with just 40 features.

6. Discussion and Future Work

Our participants reported a higher power consumption of about $5 - 12\%$ measured using the Android's internal power reporter, due to use of the DataCollector app. However, the end system will consume less power because it will collect the sensory readings for smaller time periods, i.e., 6 sec, while DataCollector gathers sensor readings all the time when the screen is on, which is on average equal to 70.3sec [9]. Moreover, we expect in the near future that all mobile platforms will be equipped with low-power continuous sensing modules [18], that will further reduce power consumption. The final implementation and its complete evaluation is a subject of future work.

We assume that during the experiment a smartphone was used solely by the owner. However, in general case this is not true, e.g., sometimes a smartphone may be used by a family member, a friend, etc. We did not apply any outlier detection approach to filter out and delete such outliers. Such filtering should in principle lead to better results.

Our model does not consider the impact of situations while authenticating. As some papers show [5], situations (i.e., walking, standing, running, etc.) may affect the behavioural pattern. If the phone is unlocked while walking the resulting pattern would be different if the same user unlocks the phone while lying on a bed. On a positive side, we tested the system in an uncontrolled fashion so the users were not constrained to a specific situation and data were gathered in a realistic fashion mixing different situations. Nevertheless might be interesting to check the impact of each situation on the aggregate results.

As future work, we will extend the DataCollector app to recognize situations (e.g., by using JigSaw engine [14], etc.) and select the most appropriate set of features for that situation.

We plan also to extend the experimental validation with a higher number of testers.

7. Conclusion

This paper presents a novel approach for unobtrusive user authentication on smartphone. Our method is based on profiling hand(s) micro-movements, after an unlock event occurs, using smartphone built-in unprivileged sensors. The design allow to implement our method as a device unlocking method and/or as a separate authentication service, which may be used by different applications (i.e., mobile banking, m-health app, etc.).

We have shown that by profiling the user based on simple time-domain features, extracted from sensory data, we can authenticate the smartphone users. To validate our approach, we launched an uncontrolled experiment with 31 qualified users (53, in total). We collected real-world readings from commonly available smartphone sensors (5 physical and 2 extracted sensory readings, i.e., LPF and HPF) and share this dataset with the research community. Using the obtained data, we inferred critical parameters for our system, e.g., the data collection time interval. We also used the dataset to assess our system. The experiments showed that our prototype achieves the TAR of 96% at an EER of 4% in the authentication task.

Acknowledgment

The authors would like to thank all the volunteers for their participation to the experiment and anonymous reviewers for their reviews and comments.

This work was partially supported by the EIT Digital SecurePhone project and European Training Network for CyberSecurity (NeCS) grant number 675320.

References

[1] Mobile users can't leave their phone alone for six minutes and check it up to 150 times a day. http://www.dailymail.co.uk/news/article-2276752/Mobile-users-leave-phone-minutes-check-150-times-day.html.

[2] A. J. Aviv et al. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, pages 1–7, 2010.

[3] M. Böhmer et al. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*, pages 47–56. ACM, 2011.

[4] A. Buriro et al. Hold & sign: A novel behavioral biometrics for smartphone user authentication.

[5] A. Buriro et al. Itsme: Multi-modal and unobtrusive behavioural user authentication for smartphones. In *International Conference on Passwords*, pages 45–61. Springer, 2015.

[6] A. Buriro et al. Touchstroke: Smartphone user authentication based on touch-typing biometrics. In *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*, pages 27–34. Springer, 2015.

[7] M. Conti et al. Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 249–259. ACM, 2011.

[8] A. De Luca et al. I Feel Like I'm Taking Selfies All Day!: Towards Understanding Biometric Authentication on Smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1411–1414, 2015.

[9] M. Harbach et al. It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *Proceedings of the Symposium On Usable Privacy and Security*, pages 213–230, 2014.

[10] E. Hayashi et al. CASA: Context-aware Scalable Authentication. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, pages 3:1–3:10, 2013.

[11] G. Holmes et al. Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE, 1994.

[12] H. Khan et al. Itus: An Implicit Authentication Framework for Android. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, pages 507–518, 2014.

[13] L. Li et al. Unobservable re-authentication for smartphones. In *NDSS*, 2013.

[14] H. Lu et al. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 71–84. ACM, 2010.

[15] U. Mahbub et al. Active user authentication for smartphones: A challenge data set and benchmark results.

[16] J. Mantyjarvi et al. Identifying users of portable devices from gait pattern with accelerometers. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii–973. IEEE, 2005.

[17] N. Neverova et al. Learning human identity from motion patterns. *IEEE Access*, 4:1810–1820, 2016.

[18] B. Priyantha, D. Lymberopoulos, and J. Liu. Littlerock: Enabling energy-efficient continuous sensing on mobile phones. *IEEE Pervasive Computing*, 10(2):12–15, 2011.

[19] O. Riva et al. Progressive Authentication: Deciding When to Authenticate on Mobile Phones. In *Proceedings of the 21st USENIX Conference on Security Symposium*, pages 301–316, 2012.

[20] F. Schaub et al. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, pages 13:1–13:10, 2012.

[21] W. Shi et al. Senguard: Passive user identification on smartphones using multiple sensors. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pages 141–148. IEEE, 2011.

[22] W. Shi et al. SenGuard: Passive User Identification on Smartphones Using Multiple Sensors. In *Proceedings of the IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 141–148, 2011.

[23] Z. Sitova et al. Hmog: A new biometric modality for continuous authentication of smartphone users. *arXiv preprint arXiv:1501.01199*, 2015.

[24] T. N. Thanh et al. The Largest Inertial Sensor-based Gait Database and Performance Evaluation of Gait-based Personal Authentication. *Pattern Recognition*, 47(1):228–237, 2014.

[25] T. Vidas et al. All Your Droid Are Belong to Us: A Survey of Current Android Attacks. In *Proceedings of the 5th USENIX Conference on Offensive Technologies*, pages 81–90, 2011.

[26] H. Xu et al. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium On Usable Privacy and Security (SOUPS 2014)*. USENIX Association, 2014.

[27] Y. Zhauniarovich et al. MOSES: Supporting and Enforcing Security Profiles on Smartphones. *IEEE Transactions on Dependable and Secure Computing*, 11(3):211–223, May 2014.

[28] J. Zhu et al. Sensec: Mobile security through passive sensing. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 1128–1133. IEEE, 2013.