POSTER: A Behavioral Biometric Authentication Framework on Smartphones

Ahmed Mahfouz[1]
e.ahmedmahfouz@mu.edu.eg

Tarek M. Mahmoud[1,2]
d.tarek@mu.edu.eg

Ahmed Sharaf Eldin[3,4]
profase2000@yahoo.com

[1] Computer Science Department, Minia University, EL-Minya, Egypt
[2] Canadian International College (CIC), Cairo, Egypt
[3] Information Systems Department, Helwan University, Egypt
[4] Faculty of Information Technology and Computer Science, Sinai University, Egypt

ABSTRACT

To protect smartphones from unauthorized access, the user has the option to activate authentication mechanisms : PIN, Password, or Pattern. Unfortunately, these mechanisms are vulnerable to shoulder-surfing, smudge and snooping attacks. Even the traditional biometric based systems such as fingerprint or face, also could be bypassed. In order to protect smartphones data against these sort of attacks, we propose a behavioral biometric authentication framework that leverages the user's behavioral patterns such as touchscreen actions, keystroke, application used and sensor data to authenticate smartphone users.

To evaluate the framework, we conducted a field study in which we instrumented the Android OS and collected data from 52 participants during 30-day period. We present the prototype of our framework and we are working on its components to select the best features set that can be used to build different modalities to authenticate users on different contexts. To this end, we developed only one modality, a gesture authentication modality, which authenticate smartphone users based on touch gesture. We evaluated this authentication modality on about 3 million gesture samples based on two schemes, classification scheme with EER 0.004, and anomaly detection scheme with EER 0.10.

Keywords

Smartphone; Authentication; Behavioral Biometrics

1. INTRODUCTION

Smartphones have become ubiquitous parts in our daily life. They combine the personal computing features in addition to the mobility features. Consequently, they contain a plethora of sensitive data and personal information. To protect these sensitive data, user has the option to enable an authentication mechanism. Unfortunately, 52% (out of 1,500) [1] and 34% (out of 500) [2] don't lock their smartphones. Inconvenience, lack of motivation and awareness were the most common reasons for not-locking smartphones. Also these mechanisms are vulnerable to different risks such as shoulder surfing and snooping attacks.

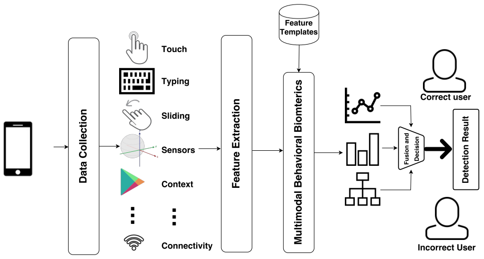


Figure 1: The Behavioral Biometric Framework architecture.

In this poster, we propose a behavioral biometric authentication framework that is going to (i) authenticate users implicitly (i.e., without interrupting their activities), and (ii) continually (i.e., authentication process is continuously repeated).

2. RELATED WORK

Several researchers conducted studies to understand the current unlocking mechanisms [4] and other researchers proposed new techniques to authenticate smartphone user, some of them based on biometric authentication [3], and others based on implicit authentication [6]. Our work is more related to implicit authentication.

In contrast with the previous work, where the majority of implicit authentication techniques were evaluated on datasets that collected in constrained settings. Moreover, some authentication methods have built based on very simple features, which expected to be statistically weak. In this work, we seek to collect realistic behavioral data in unconstrained environment and extract a discriminative set of features.

3. BEHAVIORAL BIOMETRIC FRAMEWORK

Figure 1 shows the behavioral biometric framework architecture. It has three main modules: data collection module, feature extraction module, fusion and decision module.

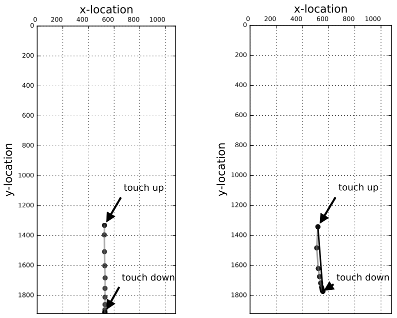
Our framework authenticate smartphone users based on authentication score, which is calculated from different authentication modalities. Each modality extracts a useful set

ASIA CCS '17 April 02-06, 2017, Abu Dhabi, United Arab Emirates

© 2017 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4944-4/17/04.

DOI: http://dx.doi.org/10.1145/3052973.3055160



(a) User u_i (b) User u_j

Figure 2: Stroke samples done by two different users from our dataset.

of features from user-level activities. Finally our framework leverages the decision fusion to take the final decision.

3.1 Data collection

To build a real-world unconstrained dataset (i.e., dataset that contains real user activities without intervention), we developed a monitoring framework and instrumented it in the Android OS (Lollipop version on Nexus 5 device). It recorded events related to device unlocking, touchscreen and sensors data in a real-life settings. By deploying our monitoring framework via Phonelab testbed, a programmable smartphone testbed, developed at the University at Buffalo and support to run experiments at the OS level on participants from University at Buffalo community [5].

The raw data corpus that we collected contains about 200 GB of smartphone user activities. The participants took part in our study during different time periods, all between July 06, 2016 and August 31, 2016, for at least 30 days each. The total number of participants who successfully accepted to install our monitoring framework were 133 but we have only included 52 participants who kept our monitoring framework for 30-day period or more.

3.2 Behavioral Biometric Modalities

Our goal is to develop more than one modality, and yet we have developed only one modality which is the gesture authentication modality. In this section, we provide a detailed description about this modality.

3.3 Gesture Modality

In this modality, we collected data from touch screen events, and then analyzed these data, extracted features vector, and then built the gesture authentication modality.

3.3.1 Gesture analysis

To interact with the touchscreen, user has to enter a **gesture**, a hand-drawn shape on a touch screen. This **gesture** can have one or more **strokes**, a sequence of consecutive timed points. Each point represented by an ordered pair of numerical coordinates (x, y), as illustrated in Figure 2.

For each touched point, we have collected the following raw data, **timestamp, coordinates, pressure, size and the action_code**, a code that specifies the state change such as touch_down, touch_move or touch_up.

3.3.2 Stroke detection

As illustrated in Figure 2, we detect the stroke based on the action_code. So, all consecutive points between touch_down and touch_up actions represent a stroke S. Each point in S represented by x_i and y_i, the coordinates of touched point, p_i, the pressure on the touched point, a_i, the size area of touched point, and t_i, timestamp of touch action, where $i = \{1, \ldots, n\}$, and n is the total number of points in the stroke.

3.3.3 Feature Extraction

For extracting useful features we analyzed the stroke from two directions, the geometry of the stroke, and its motion dynamics.

Geometric analysis, we extracted six features from the geometric analysis on the stroke, four of them represent touch down and touch up coordinates, $x_{down}, y_{down}, x_{up}, y_{up}$, and the other two features are stroke length S_{length} and stroke curvature $S_{curvature}$. Where S_{length} represents the length of specific path traveled from touch_down to touch_up points and is calculated based on the sum of line segment lengths as follows:

$$S_{length} = \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \qquad (1)$$

where n is the number of points in the stroke S. Stroke curvature $S_{curvature}$ represents the amount of deviation from being a straight line (see Figure 2b), and is calculated based on the following formula:

$$K = \frac{|X'Y'' - Y'X''|}{(X'^2 + Y'^2)^{\frac{3}{2}}} \qquad (2)$$

where X and Y are row vectors of points in the stroke. X', Y', X'', and Y'' represent the first and second derivatives, and K represents the row vector of curvatures for each point in the stroke. Then we calculate the mean of K to represent $S_{curvature}$ as follow:

$$S_{curvature} = \frac{1}{N} \sum_{i=1}^{N} K_i \qquad (3)$$

Dynamic analysis, we extracted four features from the dynamic analysis on the stroke. Given that, stroke contains a set of consecutive points. These points detected according to the motion of object (i.e., finger) on the touch screen. As a matter of fact, finger moves in curvilinear direction more often than in linear direction. Based on this fact, we calculated the displacement $S_{displacement}$, the length of the straight line between touch_down and touch_up, see Figure 2b for more clarification.

To understand how fast or slow the finger moves on the screen, we calculate the velocity at each point in the stroke as follows:

$$V = \sqrt{((X')^2 + (Y')^2)} \qquad (4)$$

where X' and Y' are the first derivative of row vectors of points in the stroke, and V is the row vector of velocities at each point. We extracted two features from this vector, mean velocity $S_{mean(V)}$ and maximum velocity $S_{max(V)}$. Also, we calculate the acceleration at each point in the stroke based on the following formula:

$$A = \frac{d^2s}{dt^2}T + k\left(\frac{ds}{dt}\right)^2 N \qquad (5)$$

where s is the travelled distance and T is the unit tangent vector and N is the unit normal vector. Then we extracted the mean acceleration for the vector A as a feature $S_{mean(acc)}$.

In addition to the extracted features from geometric and dynamic analysis, we extracted twelve features related to time, pressure and size which are described below.

Temporal Features: we extracted two temporal features $S_{duration}$, represents the total time taken to perform a stroke, and inter-stroke duration $S_{interduration}$, represents the time spent between the current and the previous stroke.

Pressure and Size features, as we mentioned before, we recorded the pressure and the size at each touched point in the stroke. We extracted five features for the pressure, two of them for touch_down S_{pDown} and touch_up S_{pUp}, and the other three features are extracted from the descriptive statistics of the pressure which are average, maximum and minimum, $S_{pAverage}, S_{pMin}, S_{pMax}$. Similarly for the touch size where we extracted $S_{sDown}, S_{sUp}, S_{sAverage}, S_{sMin}, S_{sMax}$.

3.3.4 Modeling and evaluation

We used two models to authenticate the user, classification model, which are trained on data from both legitimate user and imposters, and anomaly detection model, which are trained on data from legitimate user only.

Classification model, for each user u_i, the classifier calculates an authentication score $p(u_i)$ that represents the probability of u_i being a legitimate user. We used k-nearest neighbors classifier based on one-vs-all scheme, where we used data from other users as imposters.

Anomaly detection model, we trained the anomaly detection learning algorithm (lsanomaly [7]) with legitimate user samples and then tested for new sample based on novelty detection scheme.

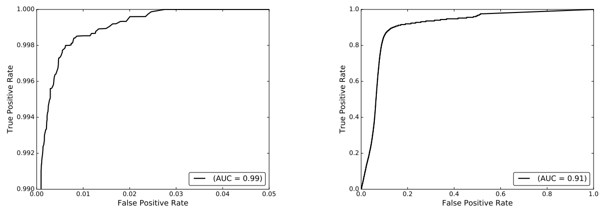
Validation method, To evaluate the accuracy of the classifiers, dataset is separated into training set and testing set. Then we performed 10-fold cross-validation.

Performance metric, in order to evaluate the performance of both models, we used ROC curve as an evaluation metric, which depicts the trade-off between TPR and FPR in a single curve at various threshold values. The top left corner of the plot represents the ideal point, where TPR equal one and FPR equal zero.

Results, *Knn* classifier achieved EER 0.004 with AUC 0.99 as shown in Figure 3a. On the other hand, the anomaly detection achieved EER 0.10 with AUC 0.91. As we can see, the classification results are better than anomaly detection because the classifier has enough training set to model both legitimate and imposters behavior but anomaly detection not. Although the classification is more powerful in terms of error cost (FP/FN) than anomaly detection. Using it could be impossible in practice, in case of the huge attack space or few training examples. So using anomaly detection is a good fit to work here.

3.4 Decision fusion

Even we have developed only one modality, but we would like to share our idea on how decision fusion will be. Our goal is to leverage different data sources to develop more than one authentication modality to authenticate the user in different contexts. Each modality is going to have a decision performance. One modality can have a strength in specific context where others not. We are going to apply some fusion scenarios on how to use theses different modalities based on different contexts and also use them as a complimentary to each other.



(a) ROC (classifier) (b) ROC (anomaly detection)

Figure 3: ROC analysis for classification model in (a) and anomaly detection model in (b) according to our dataset. AUC represents the Area under curve and summarize the performance of the models. The more the AUC is the better the system is.

4. CONCLUSIONS AND FUTURE WORK

We conducted a field study on Android phone users. We collected data related to user behavioral activities and we are developing a multimodal behavioral biometric authentication framework to authenticate smartphone users based on different contexts. Yet we are done with only one modality, gesture authentication modality (see section 3.3). Our future work is going to concentrate on developing other modalities based on keystroke and behavioral profiling biometric traits.

5. ACKNOWLEDGMENTS

We would like to thank our colleagues for their feedback on the earlier version of this Poster. The first author would like to thank Egyptian Mission sector for the doctoral scholarship.

6. REFERENCES

[1] E. Bursztein. Survey: Most people don't lock their android phones - but should. https://www.elie.net/blog/survey-most-people-dont-lock-their-android-phones-but-should. April 2015.

[2] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner. Are you ready to lock? In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 750–761, New York, NY, USA, 2014. ACM.

[3] A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80–105, 2016.

[4] A. Mahfouz, I. Muslukhov, and K. Beznosov. Android users in the wild: Their authentication and usage behavior. *Pervasive and Mobile Computing*, 32:50 – 61, 2016. Mobile Security, Privacy and Forensics.

[5] A. Nandugudi, A. Maiti, T. Ki, F. Bulut, M. Demirbas, T. Kosar, C. Qiao, S. Y. Ko, and G. Challen. Phonelab: A large programmable smartphone testbed. In *Proceedings of First International Workshop on Sensing and Big Data Mining*, SENSEMINE'13, pages 4:1–4:6, New York, NY, USA, 2013. ACM.

[6] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, July 2016.

[7] J. A. Quinn and M. Sugiyama. A least-squares approach to anomaly detection in static and sequential data. *Pattern Recogn. Lett.*, 40:36–40, Apr. 2014.